Leaky Integrate-and-Fire (LIF) Neuron Model:

From Intuition to Simulation:

Computational Neuroscience course at BioRTC, Yobe State University, Damaturu, Yobe State
(7th to 18th July, 2025)

Prof. Akinola S. Olayinka
Professor, Edo State University Iyamho, Edo State, Nigeria
PhET Fellow, University of Colorado, USA
LabXchange Ambassador, Harvard University
solayinkaa@gmail.com | akinola.olayinka@edouniversity.edu.ng | akinola.olayinka@colorado.edu |akinola.olayinka@c2snet.org
+2348062447411

Icebreaker: Quick Poll – –"Neurons vs. Computers"

If your brain were a computer, how many processing units (neurons) do you think it has?

https://pollev.com/akinolasamsonolayinka426

https://bit.ly/44Jj5Os

Icebreaker: Quick Poll - –"Neurons vs. Computers"

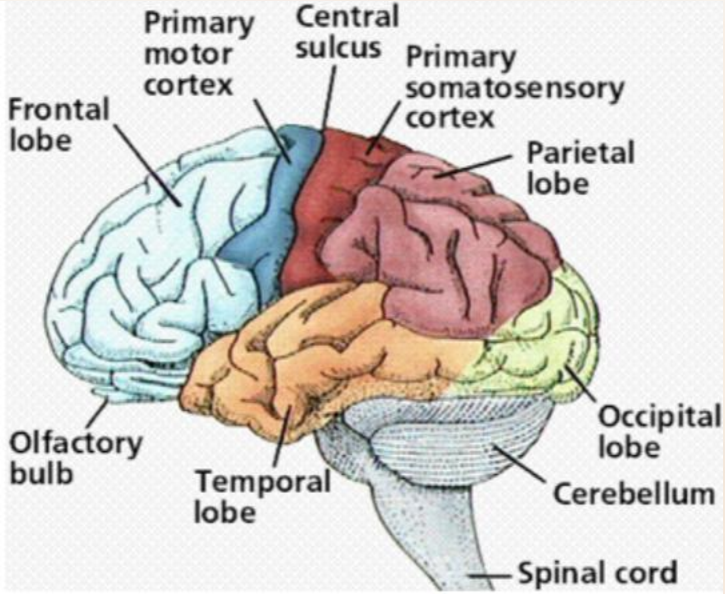
Answer: C – ~86 billion neurons!

Comparable to the stars in the Milky Way

The fascinating part is that we don't need to simulate every biological detail to understand the brain's logic.

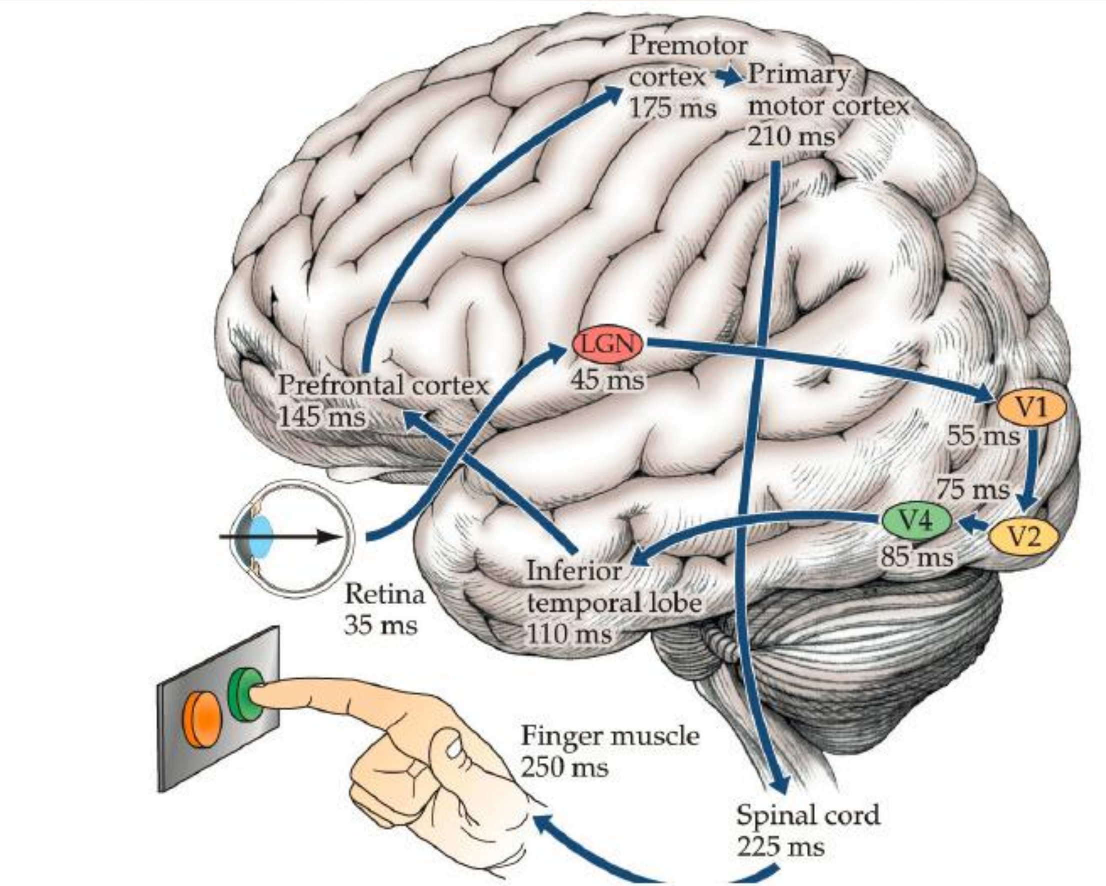
*We can use powerful, simplified models like the **Leaky Integrate-and-Fire** to discover how these 86 billion neurons fundamentally work.*

The Human Brain



The Human Brain

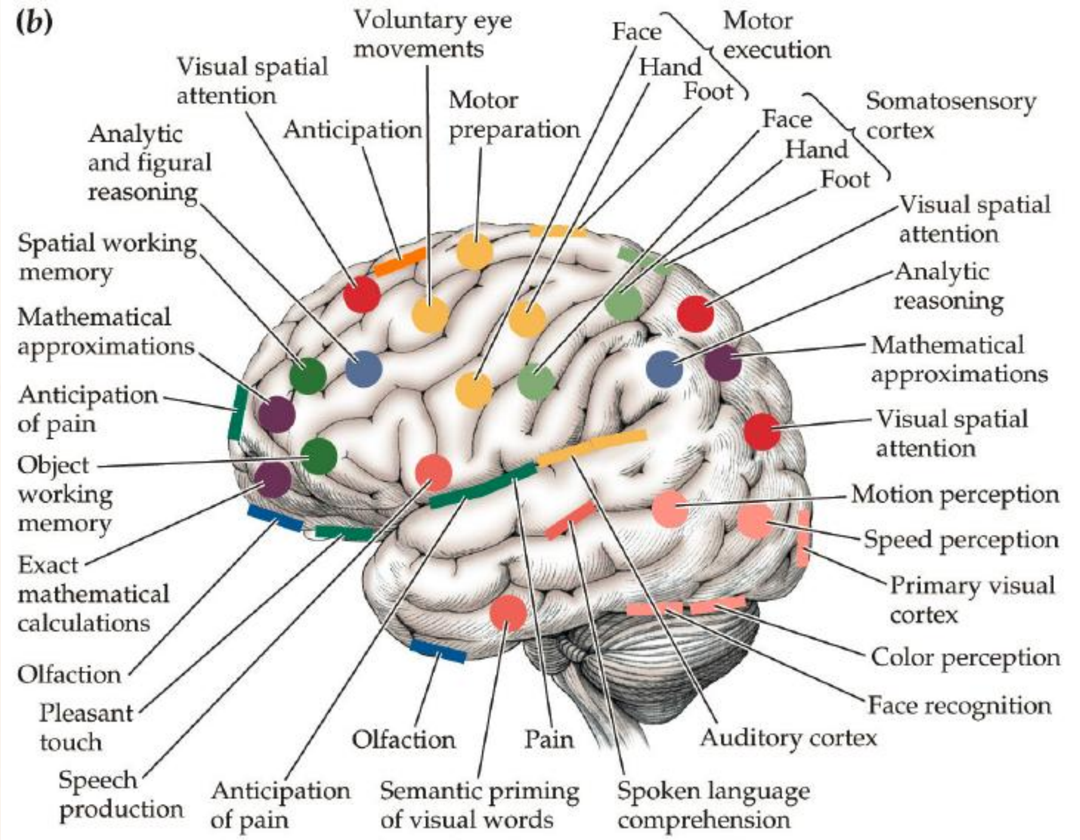
Brain I/O



Biological Psychology 5e

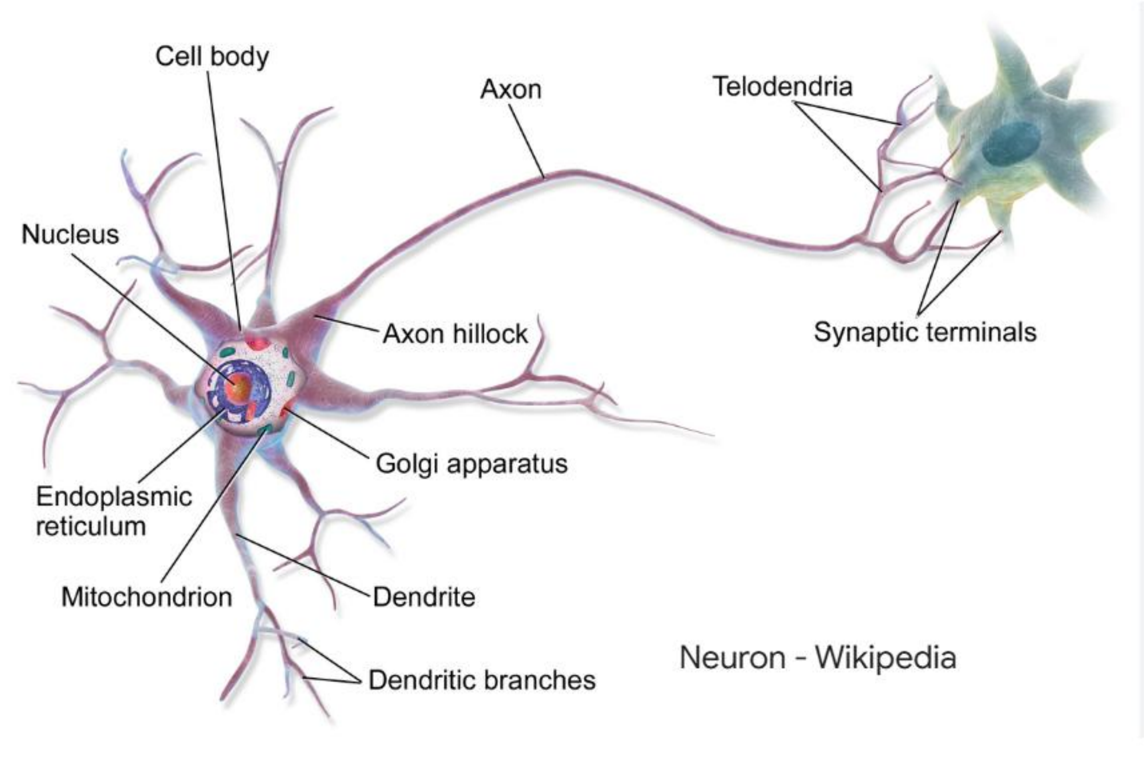
The Human Brain

Functional Organization of the Brain



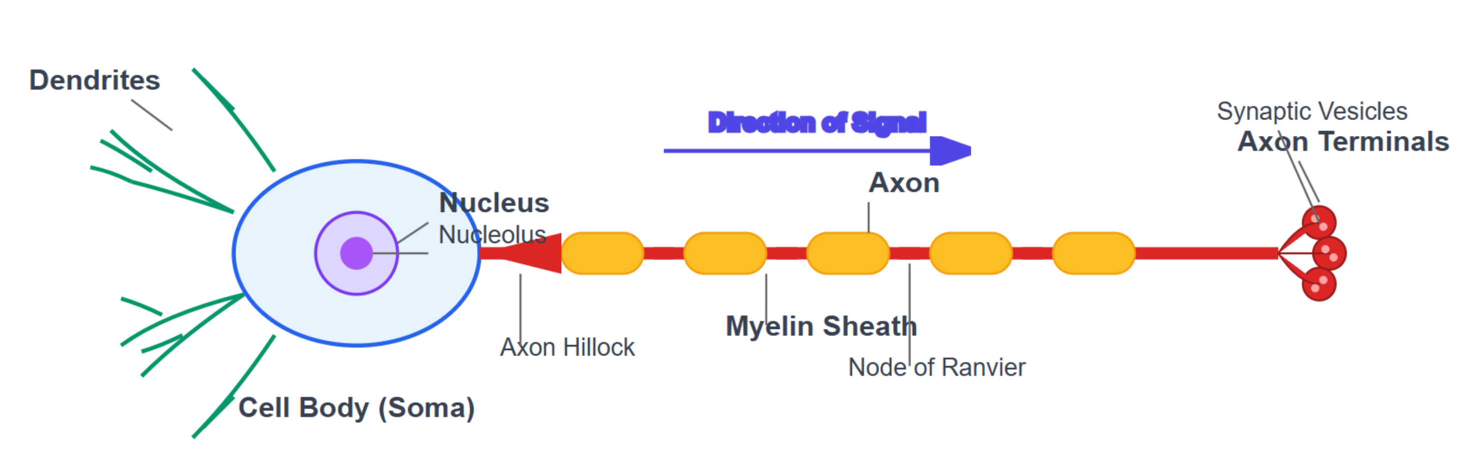
Biological Psychology 5e, Figure 1.11 (Part 2)

What is a Neuron?



A **neuron** is a specialized cell in the nervous system responsible for **receiving**, **processing**, and **transmitting** information through **electrical** and **chemical** signals.

What is a Neuron?



Main Parts:

Cell Body (Soma): Contains the nucleus and most organelles, where metabolic activities occur

Nucleus: Control center containing the cell's DNA

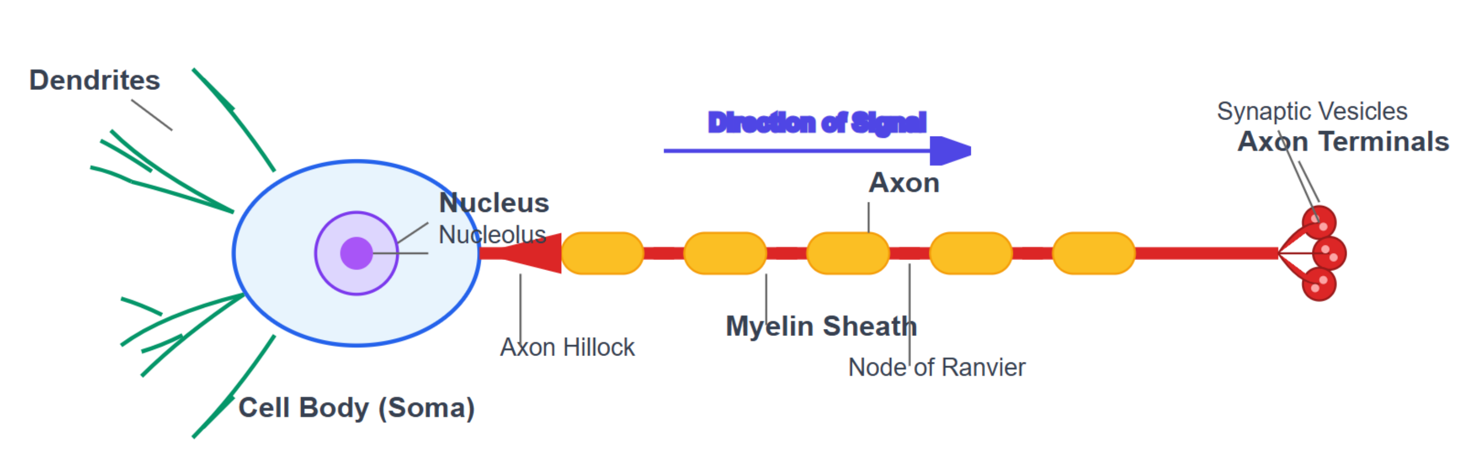
Nucleolus: Dense region within nucleus where ribosome assembly begins

Dendrites: Branched extensions that receive signals from other neurons

Axon: Long projection that carries signals away from the cell body

Axon Hillock: Where the axon begins and action potentials are initiated

What is a Neuron?

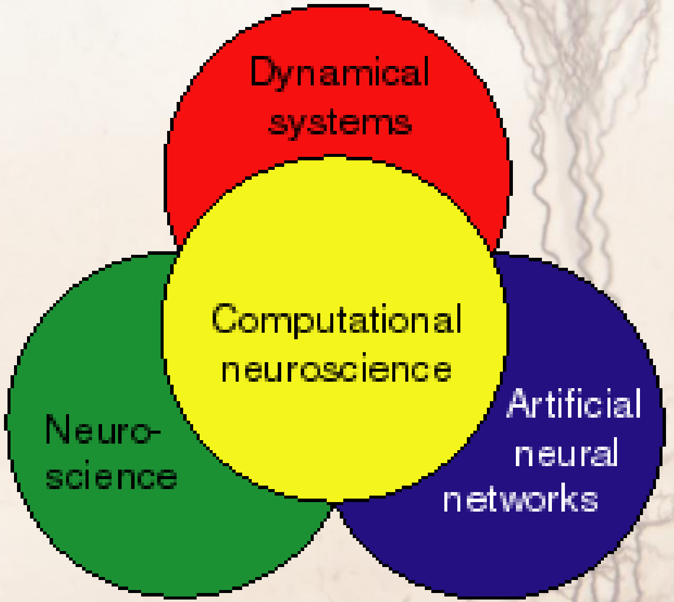


Specialized Structures:
Myelin Sheath: Fatty insulation around the axon that speeds up signal transmission
Nodes of Ranvier: Gaps in the myelin where the axon membrane is exposed
Axon Terminals: End branches of the axon that form synapses with other neurons
Synaptic Vesicles: Small sacs containing neurotransmitters for chemical signaling

- The arrow shows the typical direction of signal flow: ***from dendrites → cell body → axon → axon terminals.***
- This is how neurons communicate with each other in neural networks throughout the nervous system.

What is Computational Neuroscience

Computational neuroscience is indeed an interdisciplinary field that utilizes mathematical modeling, computer simulations, and data analysis techniques to explore the complexities of brain function.



Why Models?

Why Models?

Why Models?





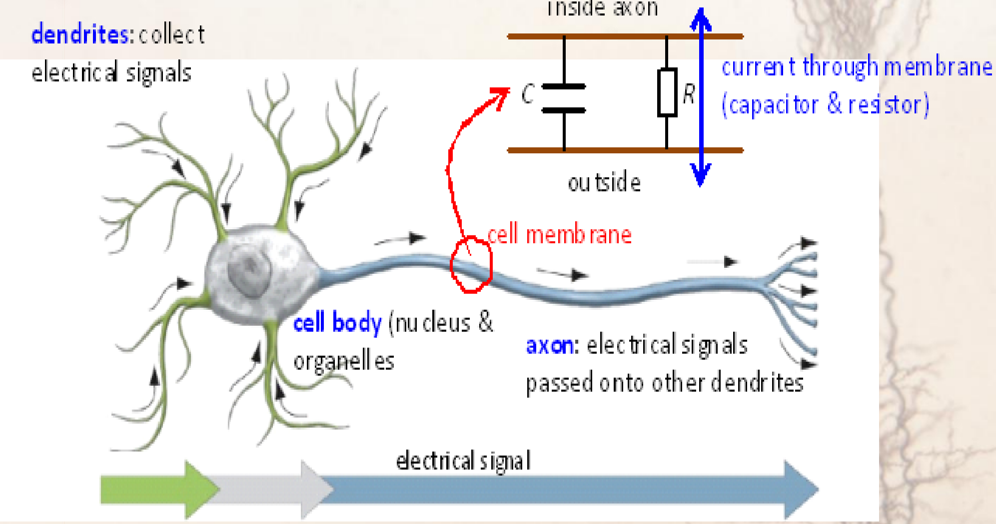

"All models are wrong, but some are useful."
— George E. P. Box

Introduction to the Passive Membrane Model

The **passive membrane model** is an electrical circuit model of a patch of neuron membrane.
It models how voltage across the membrane changes in response to electrical input.

The neuron membrane is modeled as:

- A **capacitor (C)** representing the lipid bilayer that stores charge.
- A **resistor (R)** representing ion channels that allow current to leak.
- A **voltage source (V_{rest})** representing the resting membrane potential.



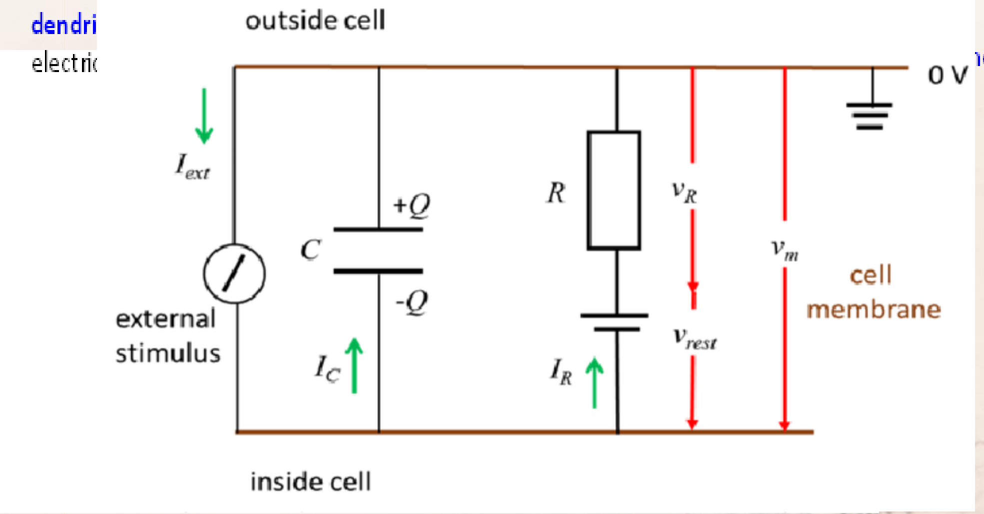
Assume no active ion channels (like voltage-gated sodium or potassium channels).
Only passive electrical properties (R & C) are considered.

Introduction to the Passive Membrane Model

The **passive membrane model** is an electrical circuit model of a patch of neuron membrane.
It models how voltage across the membrane changes in response to electrical input.

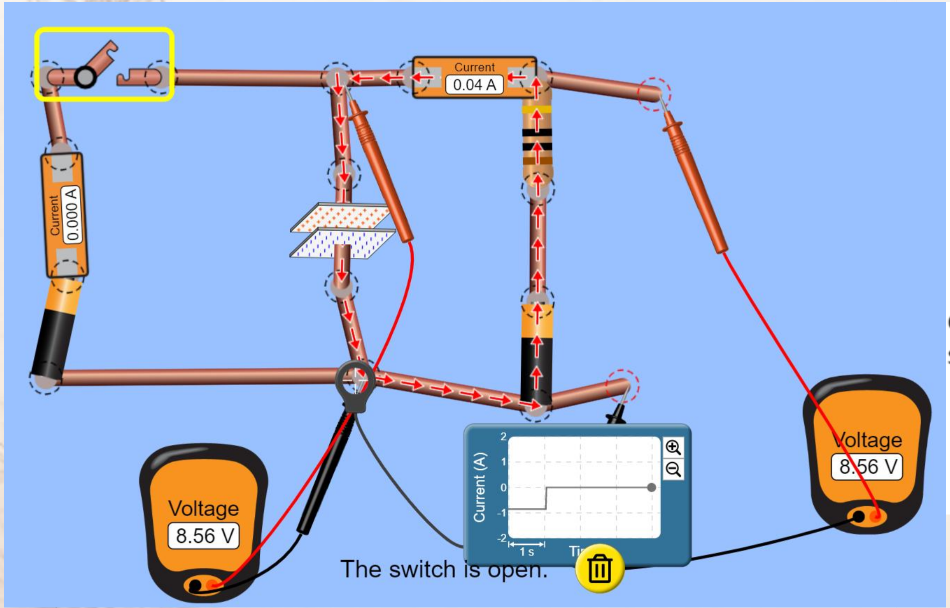
The neuron membrane is modeled as:

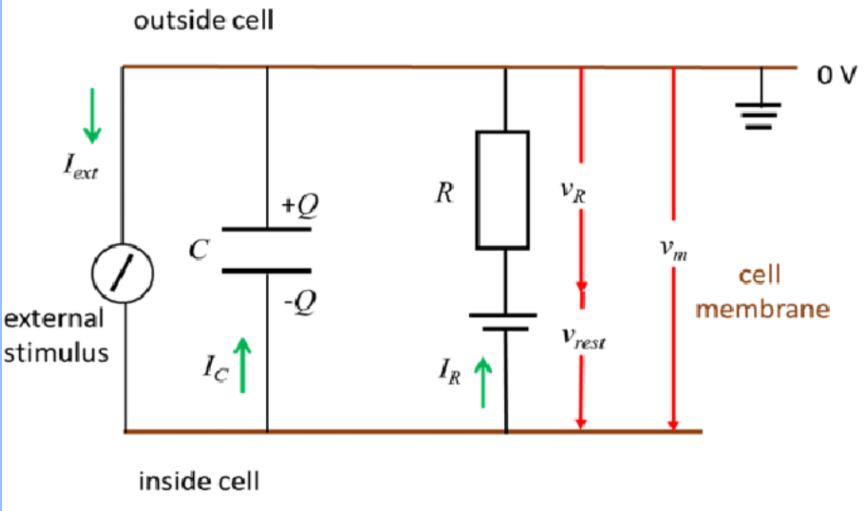
•A **capacitor (C)** representing the lipid bilayer that stores charge.
•A **resistor (R)** representing ion channels that allow current to leak.
•A **voltage source (V_{rest})** representing the resting membrane potential.



Assume no active ion channels (like voltage-gated sodium or potassium channels).
Only passive electrical properties (R & C) are considered.

Introduction to the Passive Membrane Model



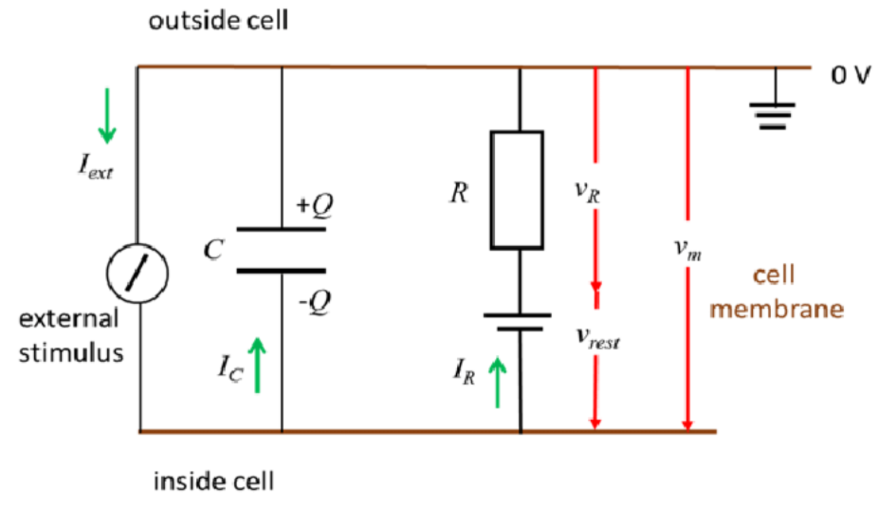


- A **resistor (R)** representing ion channels that allow current to leak.
- A **voltage source ($\mathbf{V_{rest}}$)** representing the resting membrane potential.

Assume no active ion channels (like voltage-gated sodium or potassium channels).
Only passive electrical properties (R & C) are considered.

Introduction to the Passive Membrane Model

Component	Electrical Symbol	Biological Meaning
Capacitor, **C**	Two plates storing charge	Lipid bilayer storing electrical charge
Resistor, **R**	Limits current flow	Leak channels allowing ions to flow
External current, $\mathbf{I_{ext}}$	Injected current	Synaptic or experimental input current
Voltage, $\mathbf{V_m}$	Membrane potential	Difference in potential across membrane



V_m is the Membrane potential (inside relative to outside).
V_{rest} is the Resting potential (around -70 mV).
V_R is the Voltage across the resistor. $V_R = V_m - V_{rest}$

From the circuit: $V_R = V_m - V_{rest}$

Fundamental Electrical Laws

a. Total Current Balance (Kirchhoff's Law)

All current entering the circuit must leave. Therefore:

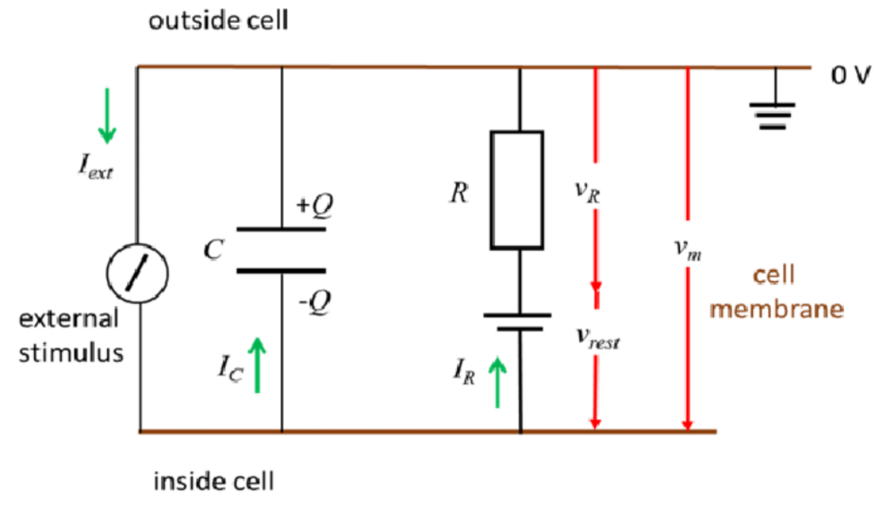
$$I_{\text{ext}} = I_C + I_R$$

b. Capacitor Law:

$$I_C = C \cdot \frac{dV_m}{dt}$$

c. Ohm's Law for the Resistor:

$$I_R = \frac{V_R}{R} = \frac{V_m - V_{\text{rest}}}{R}$$



Introduction to the Passive Membrane Model

From the circuit: $V_R = V_m - V_{rest}$

Fundamental Electrical Laws

a. Total Current Balance (Kirchhoff's Law)

All current entering the circuit must leave. Therefore:

$$I_{\text{ext}} = I_C + I_R$$

b. Capacitor Law:

$$I_C = C \cdot \frac{dV_m}{dt}$$

c. Ohm's Law for the Resistor:

$$I_R = \frac{V_R}{R} = \frac{V_m - V_{\text{rest}}}{R}$$

$$I_C = \frac{dQ}{dt}$$

Substitute $Q = C \cdot V_m$:

$$I_C = \frac{d}{dt}(C \cdot V_m)$$

If C is constant (as in most biological membranes), it comes out of the derivative:

$$I_C = C \cdot \frac{dV_m}{dt}$$

This gives us back the **capacitor law** used in the passive membrane model!

Introduction to the Passive Membrane Model

From the circuit: $V_R = V_m - V_{rest}$

Fundamental Electrical Laws

a. Total Current Balance (Kirchhoff's Law)

All current entering the circuit must leave. Therefore:

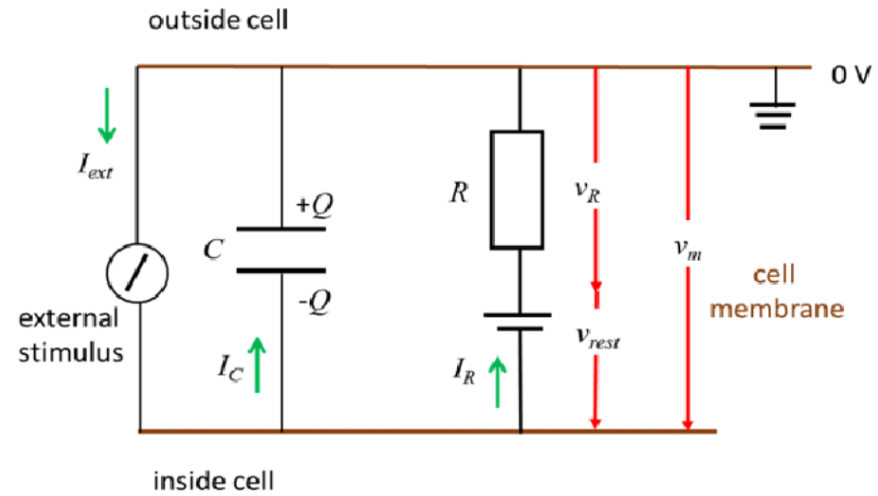
$$I_{\text{ext}} = I_C + I_R$$

b. Capacitor Law:

$$I_C = C \cdot \frac{dV_m}{dt}$$

c. Ohm's Law for the Resistor:

$$I_R = \frac{V_R}{R} = \frac{V_m - V_{\text{rest}}}{R}$$



Combine into One Equation

Introduction to the Passive Membrane Model

From the circuit: $V_R = V_m - V_{rest}$

Fundamental Electrical Laws

a. Total Current Balance (Kirchhoff's Law)

All current entering the circuit must leave. Therefore:

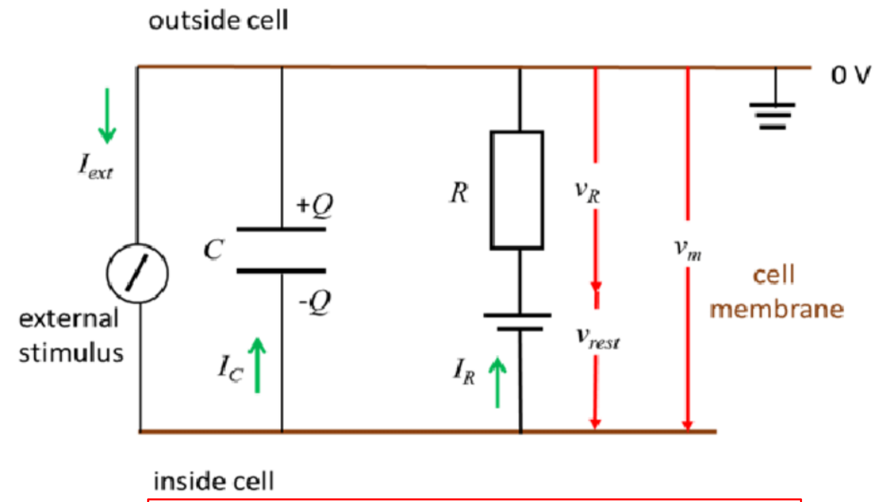
$$I_{\text{ext}} = I_C + I_R$$

b. Capacitor Law:

$$I_C = C \cdot \frac{dV_m}{dt}$$

c. Ohm's Law for the Resistor:

$$I_R = \frac{V_R}{R} = \frac{V_m - V_{\text{rest}}}{R}$$



Combine into One Equation

$$I_{\text{ext}} = C \cdot \frac{dV_m}{dt} + \frac{V_m - V_{\text{rest}}}{R}$$

This is the core differential equation describing the passive membrane.

Introduction to the Passive Membrane Model

Combine into One Equation

$$I_{\text{ext}} = C \cdot \frac{dV_m}{dt} + \frac{V_m - V_{\text{rest}}}{R}$$

This is the core differential equation describing the passive membrane.

Rearrange to make it useful for numerical simulation

$$I_{\text{ext}} = C \cdot \frac{dV_m}{dt} + \frac{V_m - V_{\text{rest}}}{R}$$

Step 1: Subtract the leak term

$$C \cdot \frac{dV_m}{dt} = -\frac{1}{R}(V_m - V_{\text{rest}}) + I_{\text{ext}}$$

Step 2: Divide by C

$$\frac{dV_m}{dt} = -\frac{1}{RC}(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{C}$$

Let: $\tau = RC$

Rearrange to make it useful for numerical simulation

$$I_{\text{ext}} = C \cdot \frac{dV_m}{dt} + \frac{V_m - V_{\text{rest}}}{R}$$

Step 1: Subtract the leak term

$$C \cdot \frac{dV_m}{dt} = -\frac{1}{R}(V_m - V_{\text{rest}}) + I_{\text{ext}}$$

Step 2: Divide by C

$$\frac{dV_m}{dt} = -\frac{1}{RC}(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{C}$$

Let: $\tau = RC$

Then the equation becomes:

$$\frac{dV_m}{dt} = -\frac{1}{\tau}(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{C}$$

This final form is ***widely used in neuron modeling.***

Introduction to the Passive Membrane Model

*This final form is **widely used in neuron modeling**.*

$$\boxed{\frac{dV_m}{dt} = -\frac{1}{\tau}(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{C}}$$

- $\frac{dV_m}{dt}$: How fast the membrane potential is changing.
- $-\frac{1}{\tau}(V_m - V_{\text{rest}})$: The **leak term**. It pulls the membrane potential back toward the resting value.
- $\frac{I_{\text{ext}}}{C}$: The **input term**. It describes how external current (from a synapse or electrode) changes the membrane potential.

Introduction to the Passive Membrane Model

This final form is ***widely used in neuron modeling***.

$$\frac{dV_m}{dt} = -\frac{1}{\tau}(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{C}$$

Initial Condition for Simulation

Usually, at $t = 0$, the membrane is at rest:

$$V_m(0) = V_{\text{rest}}$$

If $I_{\text{ext}} \neq 0$, the potential will change over time.

Introduction to the Passive Membrane Model

*This final form is **widely used in neuron modeling**.*

$$\frac{dV_m}{dt} = -\frac{1}{\tau}(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{C}$$

Numerical Solution (e.g., Euler's Method)

To simulate on a computer (Python, Matlab, etc.), use:

```
# Discrete-time update (Euler method)

V_m[t+1] = V_m[t] + dt * (-(V_m[t] - V_rest)/tau + I_ext[t]/C)
```

Introduction to the Passive Membrane Model

```
# Discrete-time update (Euler method)
V_m[t+1] = V_m[t] + dt * (-(V_m[t] - V_rest)/tau + I_ext[t]/C)
```

To simulate the system **numerically**, we approximate the derivative using **Euler's method**:

$$\frac{dV_m}{dt} \approx \frac{V_m(t+\Delta t) - V_m(t)}{\Delta t}$$

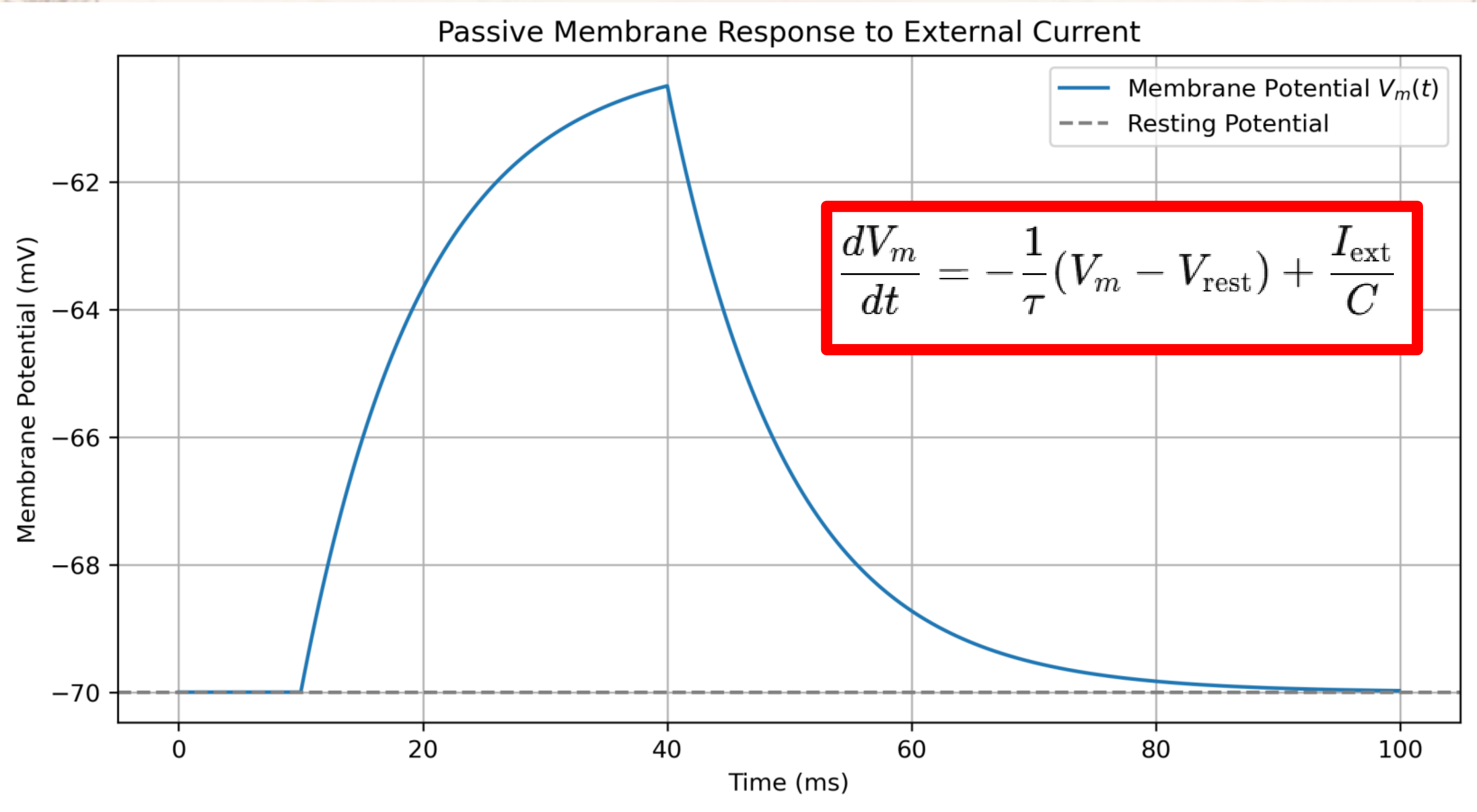
Substitute this into the differential equation:

$$\frac{V_m(t+\Delta t) - V_m(t)}{\Delta t} = -\frac{1}{\tau}(V_m(t) - V_{\text{rest}}) + \frac{I_{\text{ext}}(t)}{C}$$

Multiply both sides by Δt:

$$V_m(t+\Delta t) = V_m(t) + \Delta t \left[-\frac{1}{\tau}(V_m(t) - V_{\text{rest}}) + \frac{I_{\text{ext}}(t)}{C} \right]$$

Introduction to the Passive Membrane Model



Introduction to the Passive Membrane Model

This final form is ***widely used in neuron modeling.***

$$\boxed{\frac{dV_m}{dt} = -\frac{1}{\tau}(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{C}} \Longleftrightarrow \tau\frac{dV_m}{dt} = -(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{g}$$

Passive membrane dynamics are important for:

- Understanding subthreshold behavior.
- Modeling dendritic integration.
- Teaching how action potentials are initiated from rest.

Though simple, this model forms the **foundation** of more complex neuron models like:

- ➢ **Leaky Integrate-and-Fire (LIF)**
- ➢ **Hodgkin-Huxley model**

Integrate-and-Fire (IF) Model

Integrate-and-Fire (IF) Model

$$\boxed{\frac{dV_m}{dt} = -\frac{1}{\tau}(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{C}}$$

Recall our Passive Membrane Model

Integrate-and-Fire (IF) Model

Integrate-and-Fire (IF) Model

No leak term; the membrane potential integrates incoming current linearly.

$$\frac{dV_m}{dt} = \boxed{-\frac{1}{\tau}(V_m - V_{\text{rest}})} + \frac{I_{\text{ext}}}{C}$$

Integrate-and-Fire (IF) Model

Integrate-and-Fire (IF) Model

No leak term; the membrane potential integrates incoming current linearly.

$$\frac{dV_m}{dt} = -\frac{1}{\tau}(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{C}$$

Differential Equation:

$$\frac{dV_m}{dt} = \frac{I_{\text{ext}}}{C}$$

Threshold-Reset Rule: Spiking

$$\text{If } V_m \geq V_{\text{th}}, \quad \text{then: } V_m \leftarrow V_{\text{reset}}$$

Integrate-and-Fire (IF) Model

Integrate-and-Fire (IF) Model

No leak term; the membrane potential integrates incoming current linearly.

$$\boxed{\frac{dV_m}{dt} = \boxed{-\frac{1}{\tau}(V_m - V_{\text{rest}})} + \frac{I_{\text{ext}}}{C}}$$

Differential Equation:

$$\frac{dV_m}{dt} = \frac{I_{\text{ext}}}{C} \quad \Longleftrightarrow \quad \tau\frac{dV_m}{dt} = \frac{I_{\text{ext}}}{g}$$

Threshold-Reset Rule: **Spiking**

$$\text{If } V_m \geq V_{\text{th}}, \quad \text{then: } V_m \leftarrow V_{\text{reset}}$$

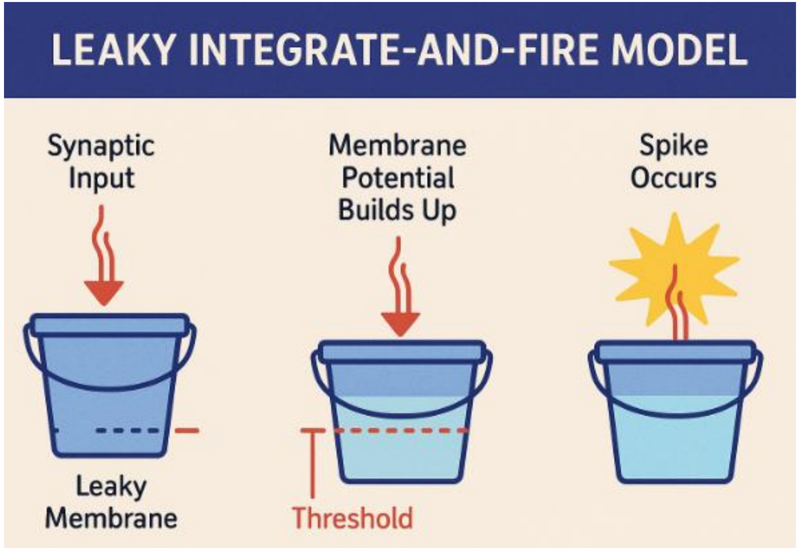
*The variable **g** typically represents **membrane conductance**. It is the inverse of **membrane resistance** R.*

Leaky Integrate-and-Fire (LIF) Model

Leaky Integrate-and-Fire (LIF) Model

The LIF model is a biologically inspired and computationally simple model that simulates how neurons:

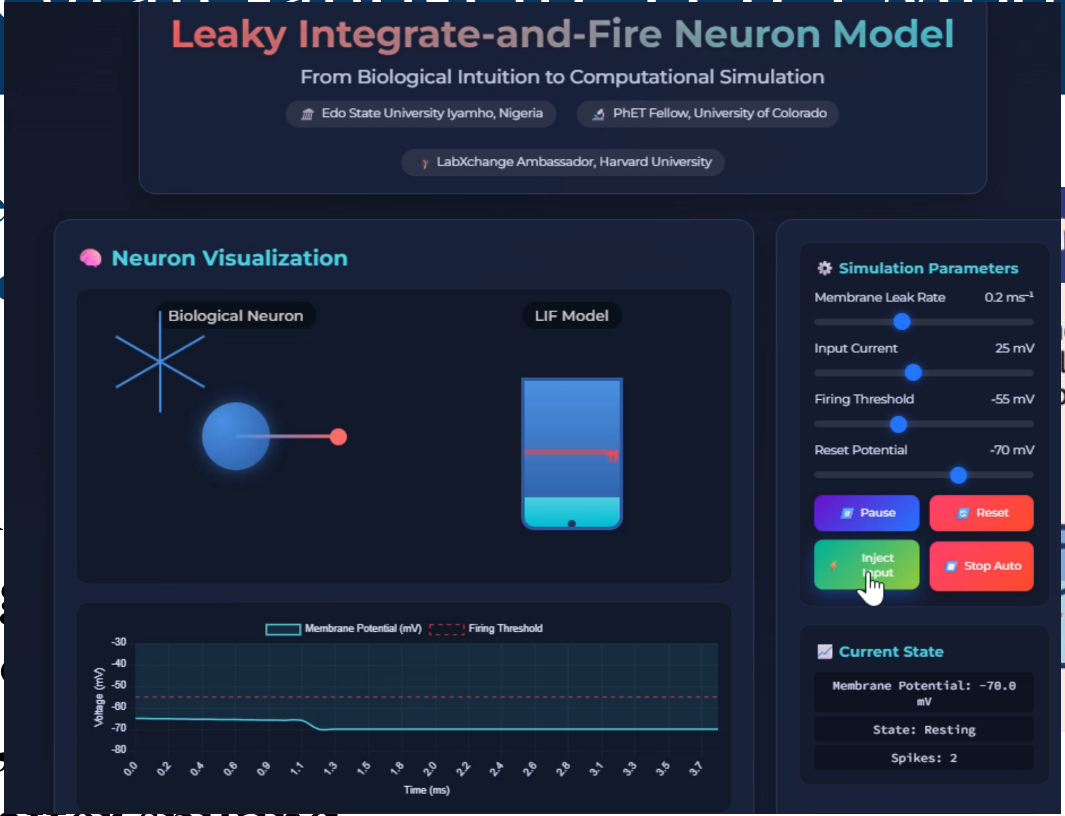
- ➢ **Integrate input over time,**
- ➢ **Leak voltage back toward rest,**
- ➢ **Fire a spike when a threshold is reached,**
- ➢ **And reset after spiking.**

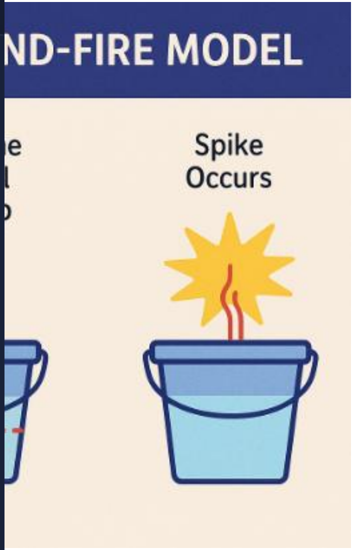


Leaky Integrate-and-Fire (LIF) Model

The LIF mode
inspired and
simple model
neurons:

- Integrate i
- Leak volta
- Fire a spik
  is reached,
- And reset after spiking.





Leaky Integrate-and-Fire (LIF) Model

*Adds a **leak** to the IF model that pulls voltage back toward rest.*

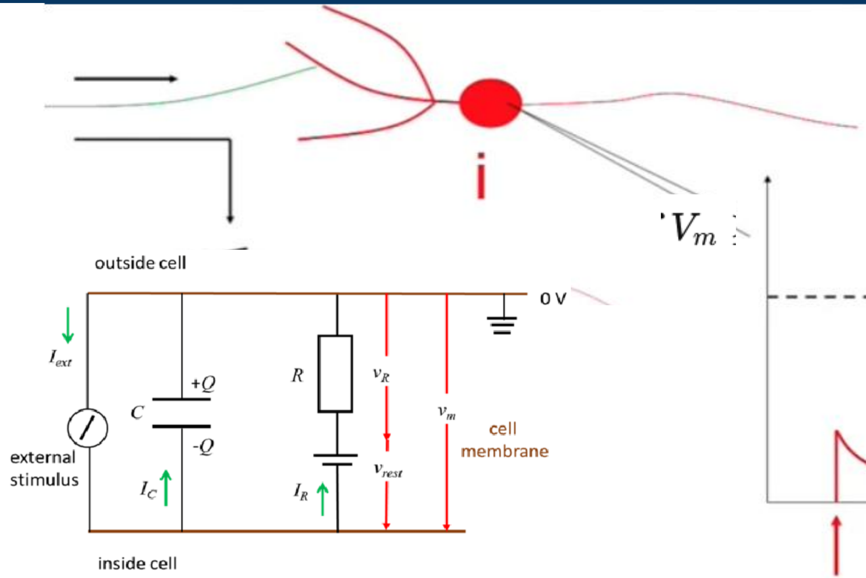
Differential Equation:

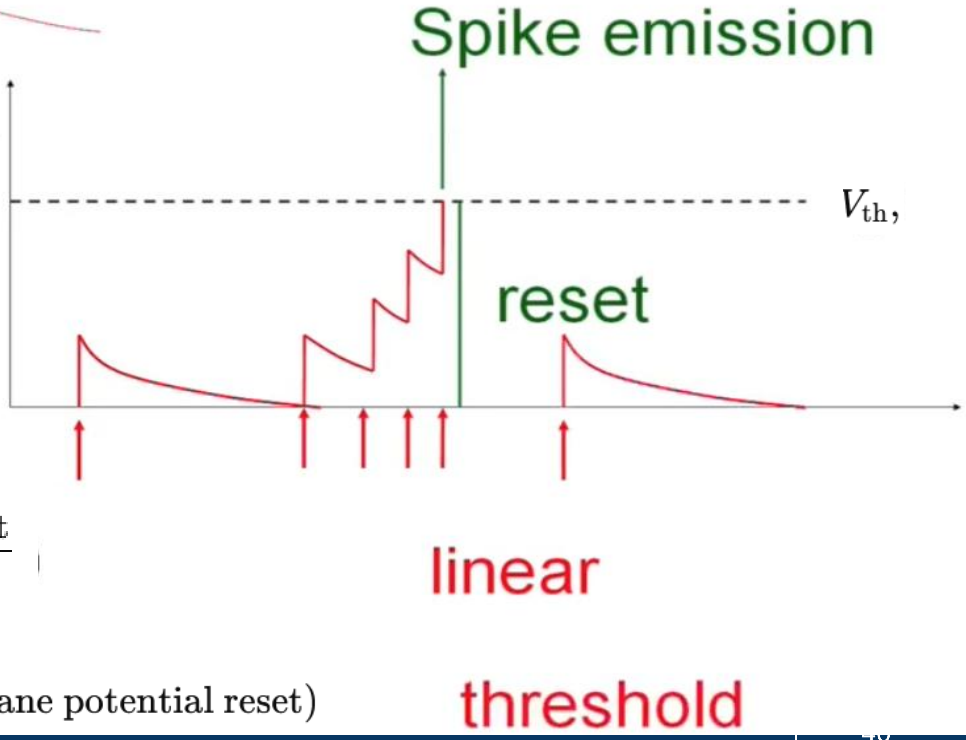
$$\frac{dV_m}{dt} = -\frac{1}{\tau}(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{C} \quad \Longleftrightarrow \quad \tau\frac{dV_m}{dt} = -(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{g}$$

Threshold-Reset Rule: **Spiking**

$$\text{If } V_m \geq V_{\text{th}}, \quad \text{then: } V_m \leftarrow V_{\text{reset}}$$

Leaky Integrate-and-Fire (LIF) Model





$$\tau \frac{dV_m}{dt} = -(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{g}$$

$$\textbf{If } V_m \geq V_{\text{th}}, \quad \textbf{then: } \begin{cases} \text{Neuron fires a spike} \\ V_m \leftarrow V_{\text{reset}} \quad (\text{membrane potential reset}) \end{cases}$$

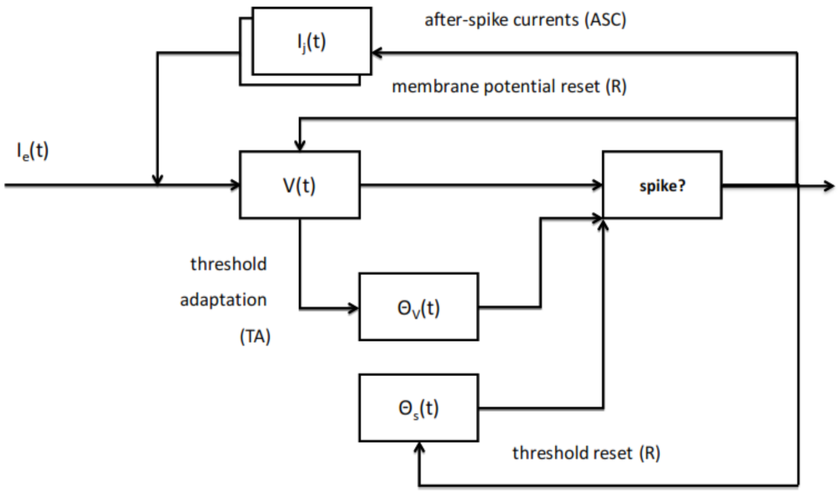
Leaky Integrate-and-Fire (LIF) Model

Summary of Passive, IF, and LIF models

Model	Differential Equation	Spiking Rule	Description
Passive	$\tau \frac{dV_m}{dt} = -(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{g}$	❌ None	Just passive charging and leaking, no threshold or reset.
IF	$\tau \frac{dV_m}{dt} = \frac{I_{\text{ext}}}{g}$	✅ If $V_m \geq V_{\text{th}}$, then $V_m \leftarrow V_{\text{reset}}$	Integrates current; no leak; spikes when threshold is crossed.
LIF	$\tau \frac{dV_m}{dt} = -(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{g}$	✅ If $V_m \geq V_{\text{th}}$, then $V_m \leftarrow V_{\text{reset}}$	Combines passive leak and integration; more biologically plausible.

Generalized Leaky Integrate-and-Fire (gLIF) Model

- From LIF to gLIF: Adding Realism to Neuron Models.
- The standard LIF model simplifies spiking, ignoring many biophysical features.
- The generalized LIF model extends LIF by adding:
  - Adaptation currents (e.g., after-spike hyperpolarization)
  - Dynamic threshold
  - Multiple time constants
  - Conductance-based inputs



Mihalas and Niebur 2009
Corinne Teeter, Stefan Mihalas

Generalized Leaky Integrate-and-Fire (gLIF) Model

gLIF Membrane Potential Dynamics

$$\tau \frac{dV_m}{dt} = -(V_m - V_{\text{rest}}) + \frac{I_{\text{ext}}}{g} - w$$

Also include (if modeling adaptation):

$$\tau_w \frac{dw}{dt} = a(V_m - V_{\text{rest}}) - w$$

$$\text{After spike: } w \to w + b$$

w: spike-triggered adaptation current

a,b: adaptation parameters

- ***a****: Strength of voltage-driven adaptation*
- ***b****: Increment added to w after each spike*

τ_w: Time constant for adaptation current

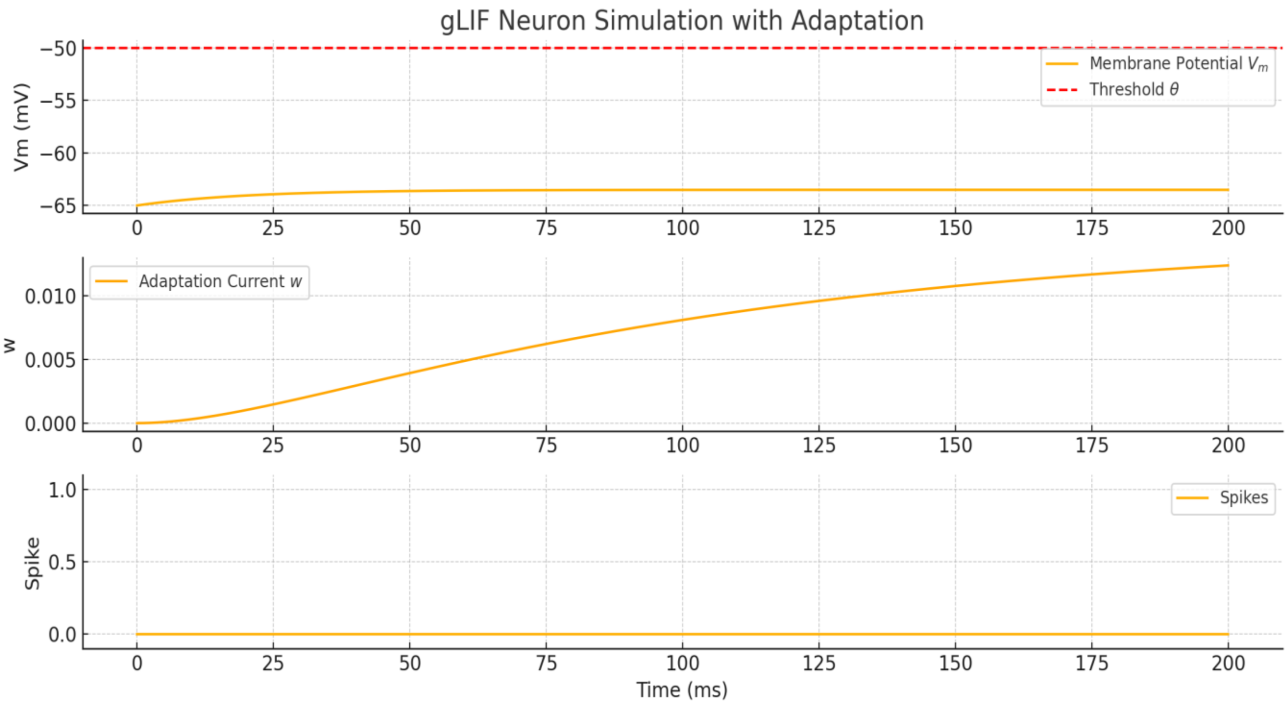
Threshold condition: when $V_m \geq V_{th}(t)$, fire and reset

Generalized Leaky Integrate-and-Fire (LIF) Model

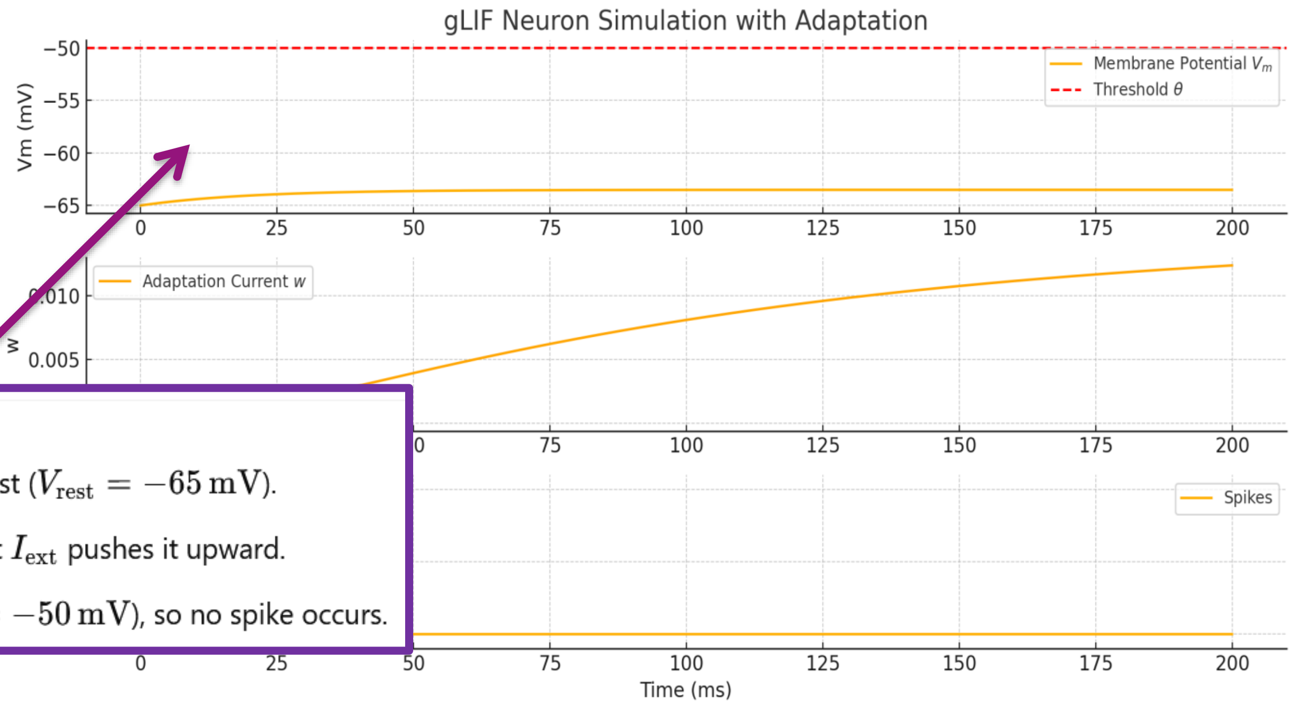
Difference between LIF and gLIF

Feature	LIF	gLIF
Leak	✓	✓
Spiking	✓	✓
Adaptation	✗	✓
Dynamic Threshold	✗	✓
Computational Load	Low	Moderate

Generalized Leaky Integrate-and-Fire (gLIF) Model



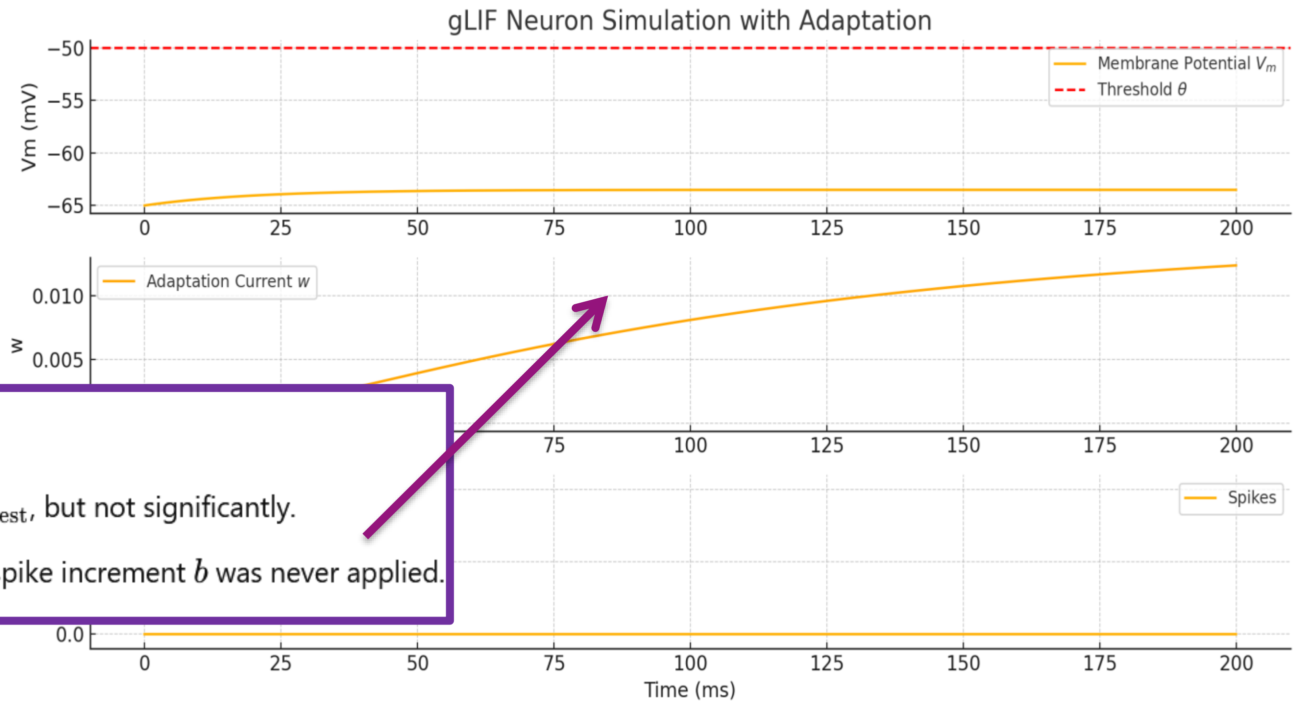
Generalized Leaky Integrate-and-Fire (gLIF) Model



1. Membrane Potential (V_m)

- The membrane potential starts at rest ($V_{\text{rest}} = -65\,\text{mV}$).
- It slowly rises as the external current I_{ext} pushes it upward.
- It **never reaches the threshold** ($\theta = -50\,\text{mV}$), so no spike occurs.

Generalized Leaky Integrate-and-Fire (gLIF) Model



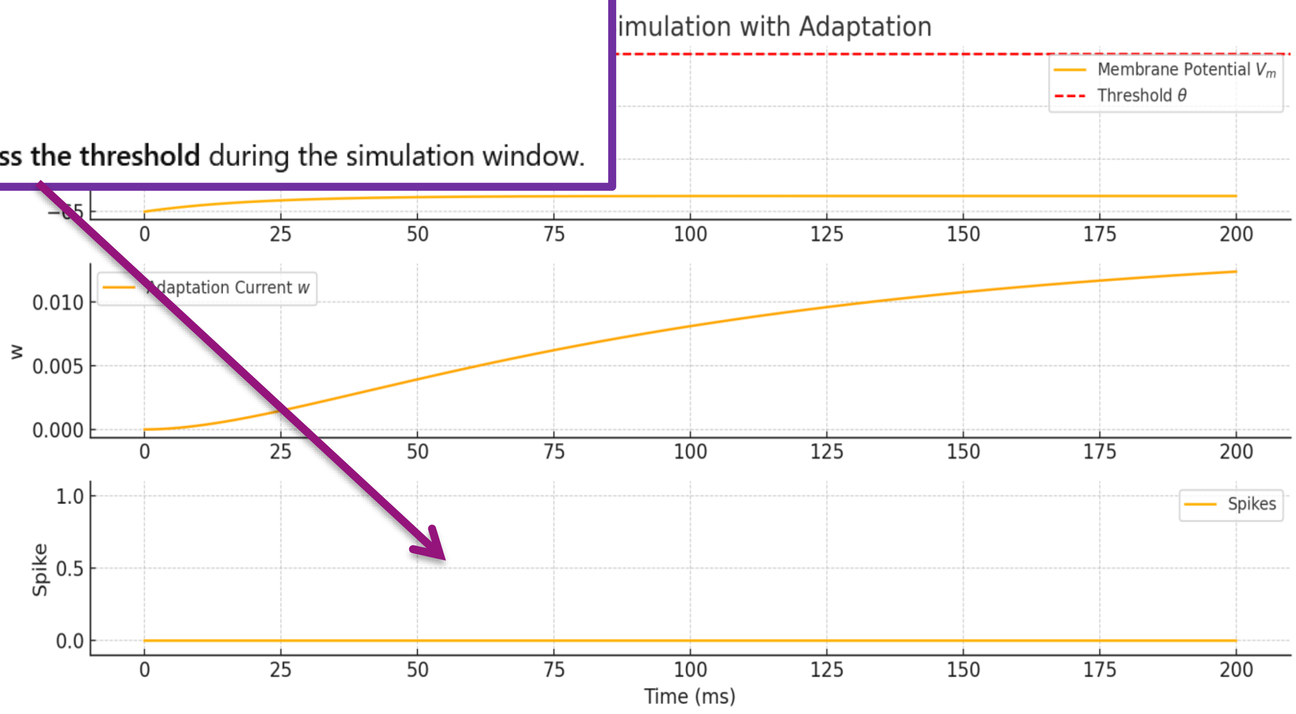
2. Adaptation Current (w)

- Increases slowly because $V_m > V_{\text{rest}}$, but not significantly.
- Since no spike occurred, the post-spike increment b was never applied.

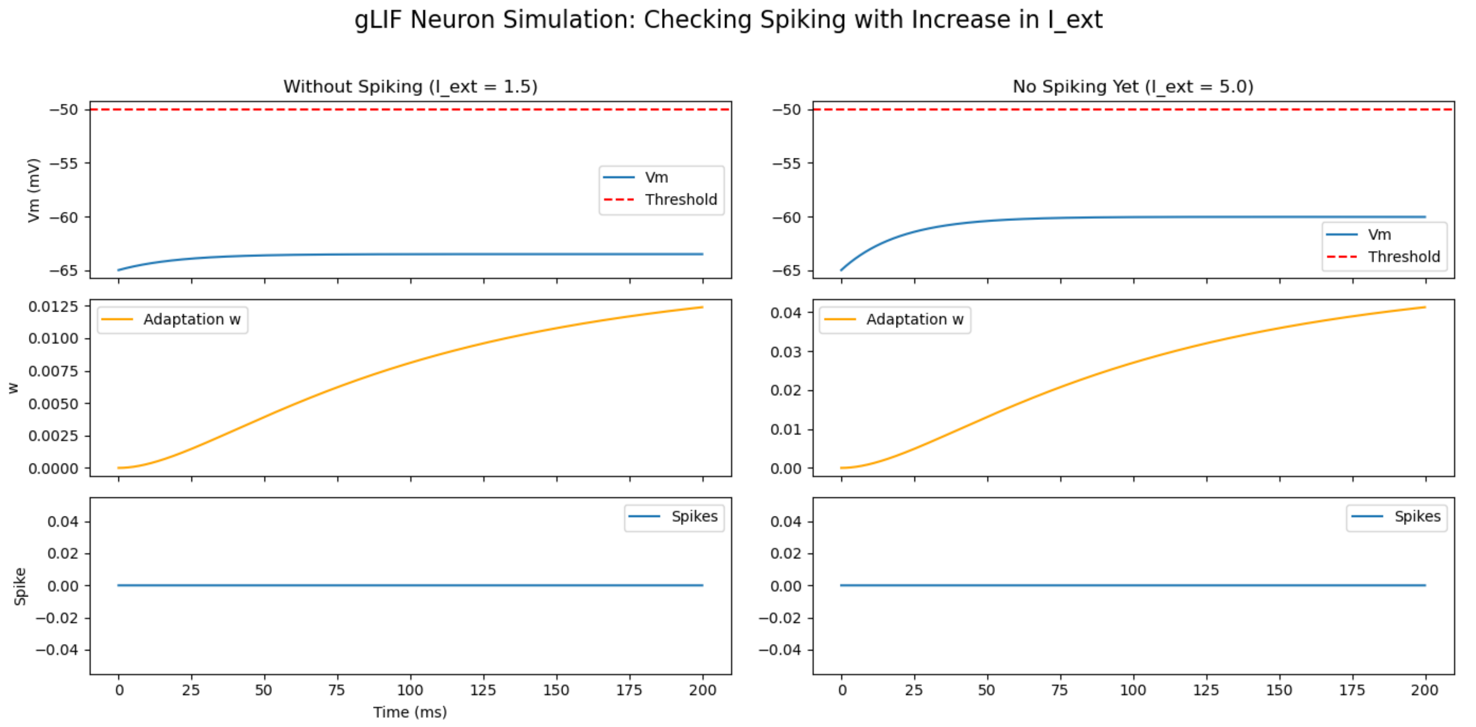
Generalized Leaky Integrate-and-Fire (gLIF) Model

3. Spike Output

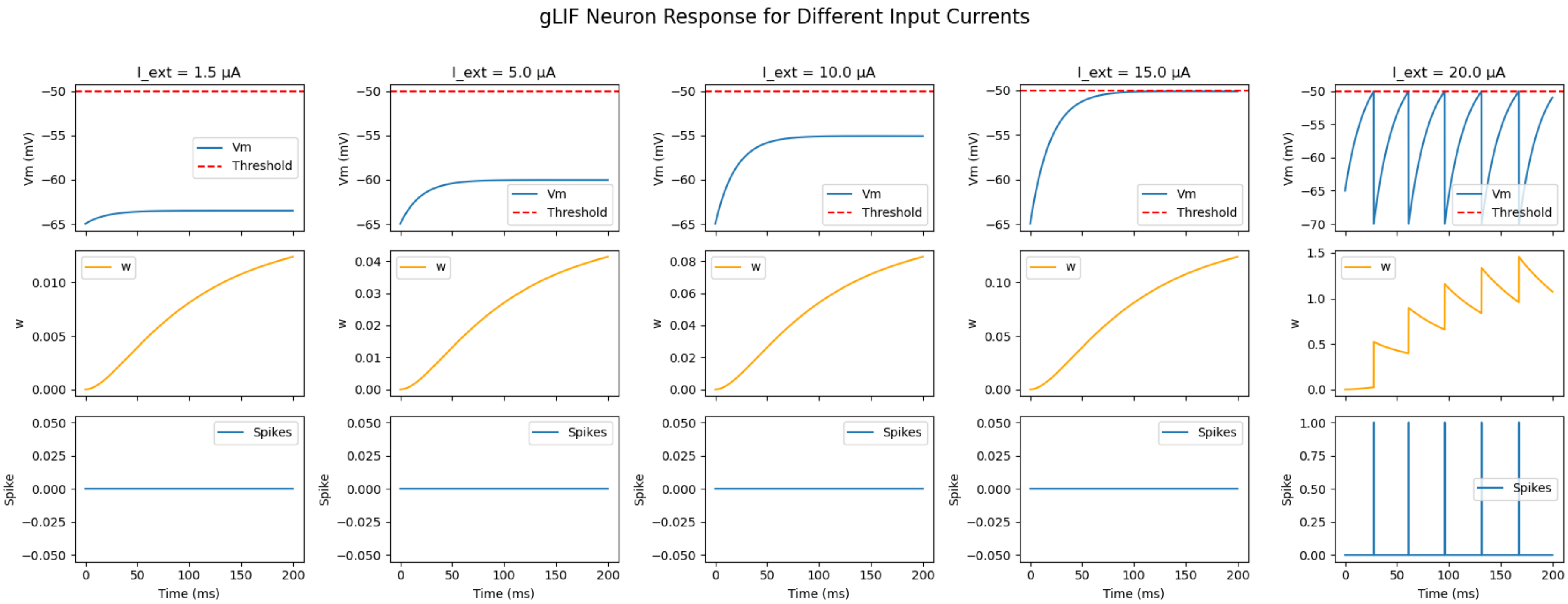
- No spikes were recorded (flat line).
- This confirms that $V_m(t)$ **did not cross the threshold** during the simulation window.



Generalized Leaky Integrate-and-Fire (gLIF) Model



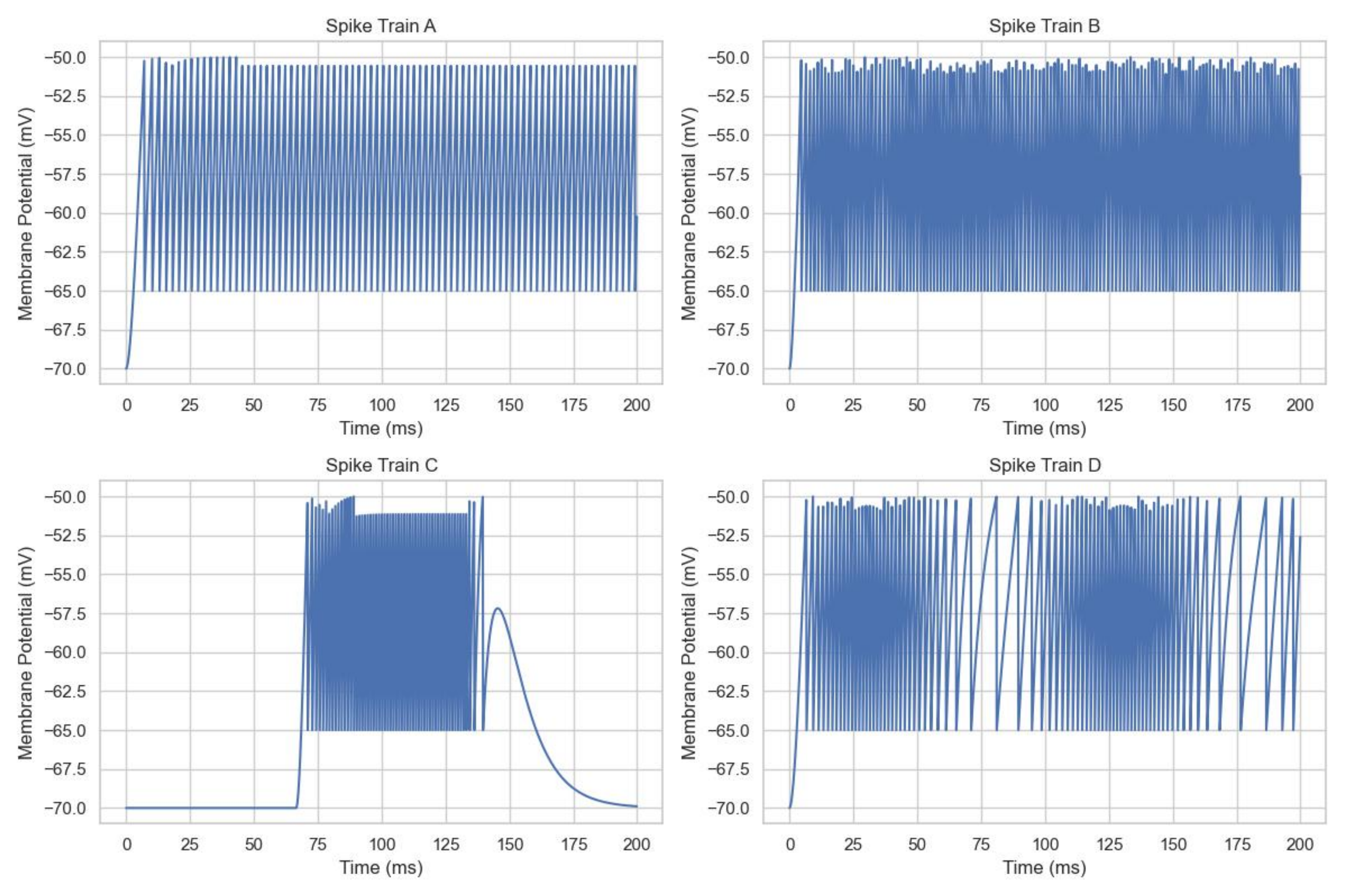
Generalized Leaky Integrate-and-Fire (gLIF) Model



You are shown 4 spike trains (A–D) from gLIF neurons. Each was generated using a different input. Can you guess which has **Noisy high-frequency current**?

https://pollev.com/akinolasamsonolayinka426

https://bit.ly/44Jj5Os



Let's see your choices:

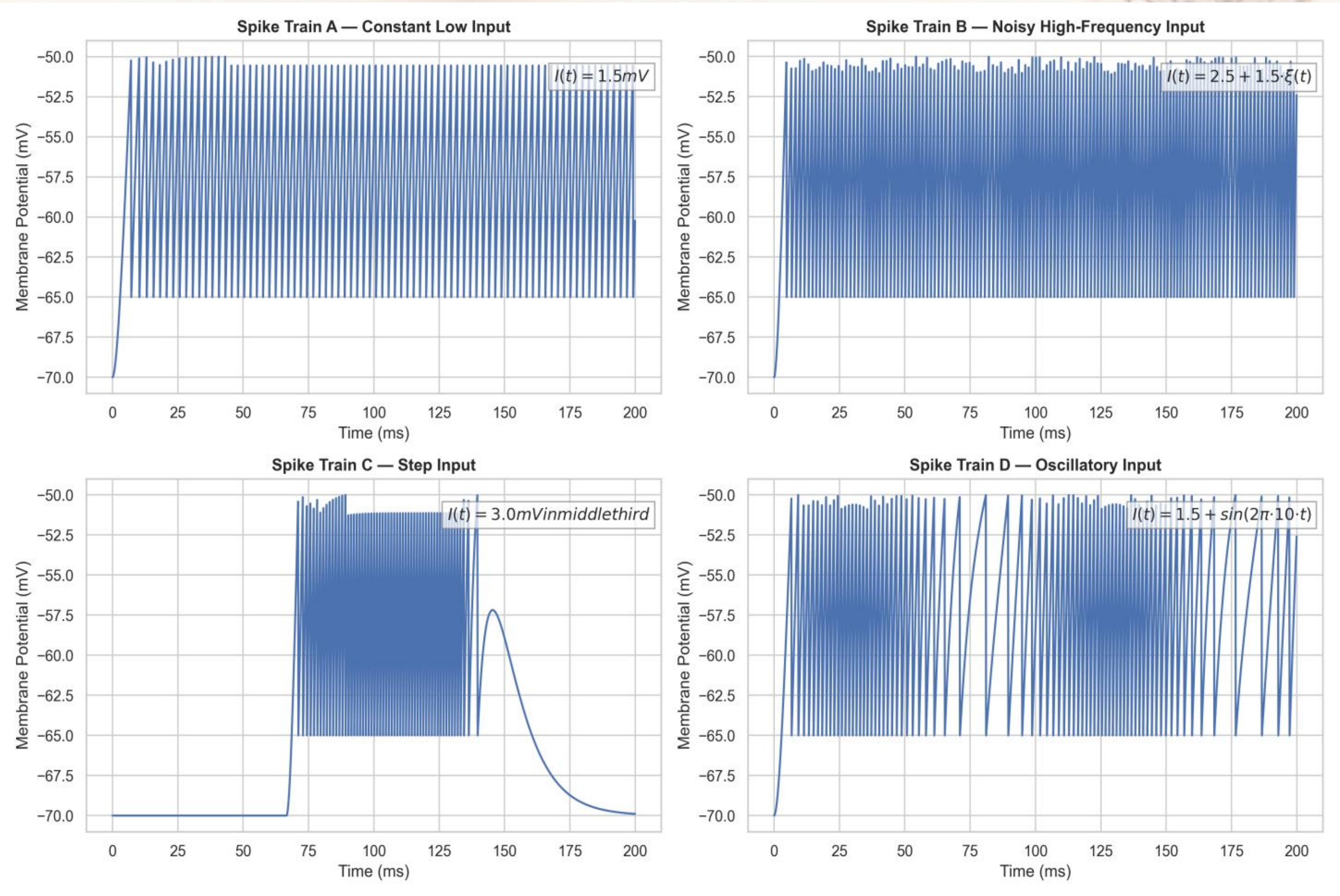
https://pollev.com/akinolasamsonolayinka426

https://bit.ly/44Jj5Os

Reveal:

https://pollev.com/akinolasamsonolayinka426

https://bit.ly/44Jj5Os



LIF Model Extensions: Beyond gLIF

🔁 **Adaptive LIF:**

- Adds a spike-triggered adaptation current
- Models neuron fatigue and firing rate adaptation
- Used in simulating sustained inputs and bursting dynamics

⚡ **Conductance-Based LIF**

- Inputs modeled as dynamic synaptic conductances
- More realistic than current-based models
- Accounts for the dependence on membrane potential and reversal potentials

⛈️ **Stochastic LIF**

- Introduces randomness in input or threshold
- Captures variability in spike timing
- Useful for modeling sensory neurons or cortical noise

Parameter Tuning: C, R, V_{th}, V_{reset}

Membrane Capacitance (C):

- Typical range: 50-500 pF
- Larger $C \rightarrow$ slower voltage changes
- Affects $\tau = RC$

Membrane Resistance (R):

- Typical range: 10-1000 MΩ
- Higher $R \rightarrow$ more sensitive to current
- Determines input resistance

Spike Threshold (V_{th}):

- Typical: -55 to -40 mV
- Lower threshold $\rightarrow$ easier to spike
- Can be adaptive in real neurons

Reset Potential (V_{reset}):

- Typical: -80 to -60 mV
- Deeper reset $\rightarrow$ longer inter-spike intervals
- Affects spike frequency adaptation

Effect of Time Constant on Response

Time Constant: $\tau = RC$

Fast Neurons (τ small, e.g., 5 ms):

- Rapid voltage changes
- Quick response to inputs
- Higher maximum firing rates
- Good temporal resolution

Slow Neurons (τ large, e.g., 100 ms):

- Smooth voltage changes
- Temporal integration of inputs
- Lower firing rates
- Memory-like properties

$$V_{m,\text{steady}} = V_{\text{rest}} + \frac{I_{\text{ext}}\tau}{C}$$

$$\text{Rise time} \propto \tau$$

$$\text{Decay time} = \tau$$

Effect of Current on Response

What Happens with Stronger Current?

Current Strength Effects:

Current Level	Behavior	Firing Rate
$I_{\text{ext}} < I_{\text{rheo}}$	No spikes	0 Hz
$I_{\text{ext}} = I_{\text{rheo}}$	Threshold	$\sim$0 Hz
$I_{\text{ext}} > I_{\text{rheo}}$	Regular spiking	Increases
$I_{\text{ext}} \gg I_{\text{rheo}}$	High-frequency spiking	Saturates

Firing Rate Formula: For constant current above threshold:

$$f = \frac{1}{\tau \ln\left(\frac{I_{\text{ext}}\tau/C-(V_{\text{reset}}-V_{\text{rest}})}{I_{\text{ext}}\tau/C-(V_{\text{th}}-V_{\text{rest}})}\right)}$$

Effect of Leak

What If the Leak Is Too Large?

Large Leak (small τ):

Effects:

- Fast decay to resting potential
- Requires stronger current to reach threshold
- Shorter membrane time constant
- Reduced temporal integration

Mathematical Analysis:

$$I_{\text{rheo}} = \frac{C(V_{\text{th}} - V_{\text{rest}})}{\tau}$$

Refractory Period: Why and How?

Why Refractory Periods?

- • Real neurons can’t spike infinitely fast
- • Sodium channels need time to recover
- • Prevents unrealistic high-frequency firing
- • Adds biological realism

Types of Refractory Periods:

1. Absolute Refractory Period:
- • Complete inability to spike
- • Typical duration: 1-2 ms
- • Voltage clamped at Vreset

2. Relative Refractory Period:
- • Elevated threshold
- • Gradual recovery
- • Duration: 5-10 ms

Biological Relevance of LIF

- LIF captures essential dynamics of real neurons.
- Used in large-scale brain simulations.

Key Advantage & Limitation of LIF

Advantage:

- Simple, computationally efficient, captures key dynamics.

Limitations:

- Ignores detailed ion channel dynamics, spatial structure.

Though Hodgkin-Huxley (HH) model is more detailed but computationally expensive than LIF.

Key Application Areas of LIF

- 🔬 **Neural Circuit Modeling**
e.g., using **Brian2**, **NEST**, **NEURON**
*(Simulating realistic spiking behavior in **cortical networks, sensory processing**)*

- 🤖 **Spiking Neural Networks (SNNs)**
Integrated into **machine learning** models for temporal pattern recognition and low-power **AI**

- ⚙️ **Neuromorphic Computing**
Hardware implementations on platforms like:
– **Intel Loihi**
– **SpiNNaker**

https://www.eenewseurope.com/en/spinnaker-neuromorphic-supercomputer-reaches-one-million-cores-2/

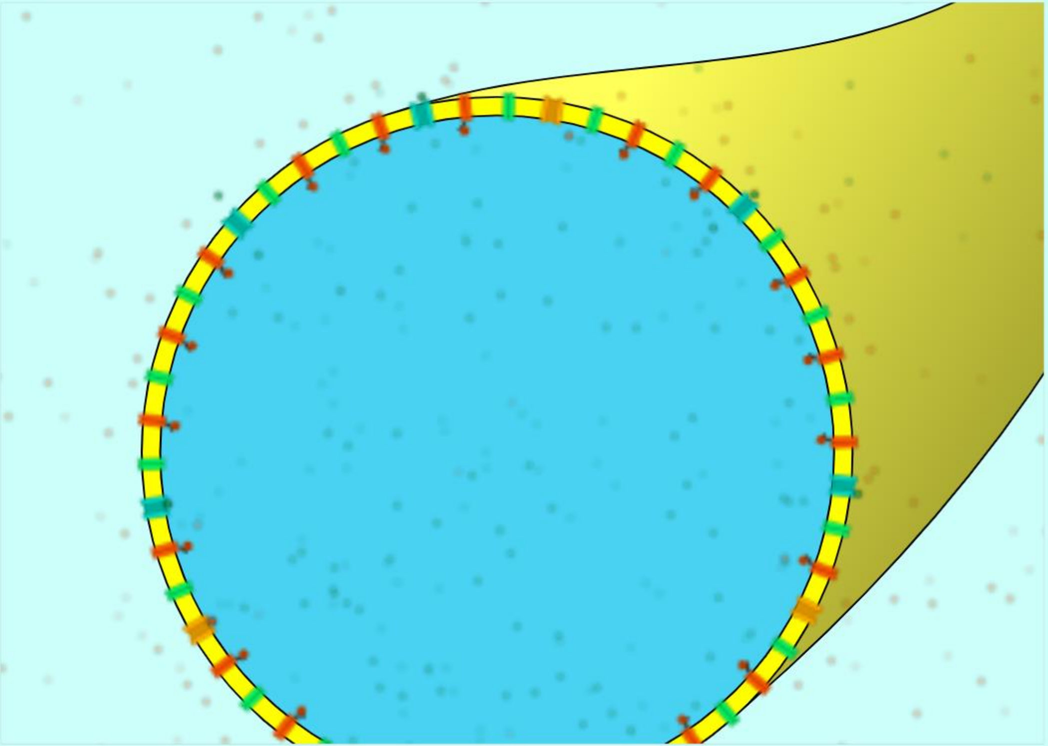
Neuron Simulation using PhET Sim

https://phet.colorado.edu/sims/html/neuron/latest/neuron_all.html

https://phet.colorado.edu/services/download-servlet?filename=/activities/4217/phet-contribution-4217-7617.pdf

https://phet.colorado.edu/en/activities/3608

https://phet.colorado.edu/en/activities/5081



Legend

- Sodium Ion (Na+)
- Potassium Ion (K+)
- Sodium Gated Channel
- Potassium Gated Channel
- Sodium Leak Channel
- Potassium Leak Channel

Show

- [x] All Ions
- [ ] Charges
- [ ] Concentrations
- [ ] Potential Chart

- Fast Forward
- Normal
- Slow Motion

Stimulate Neuron

Conclusion

- The **Generalized Leaky Integrate-and-Fire (gLIF)** model is a powerful abstraction of how real neurons behave.

- gLIF bridges the gap between **neuroscience data** and **applied computing**, enabling simulations of the brain's dynamics.

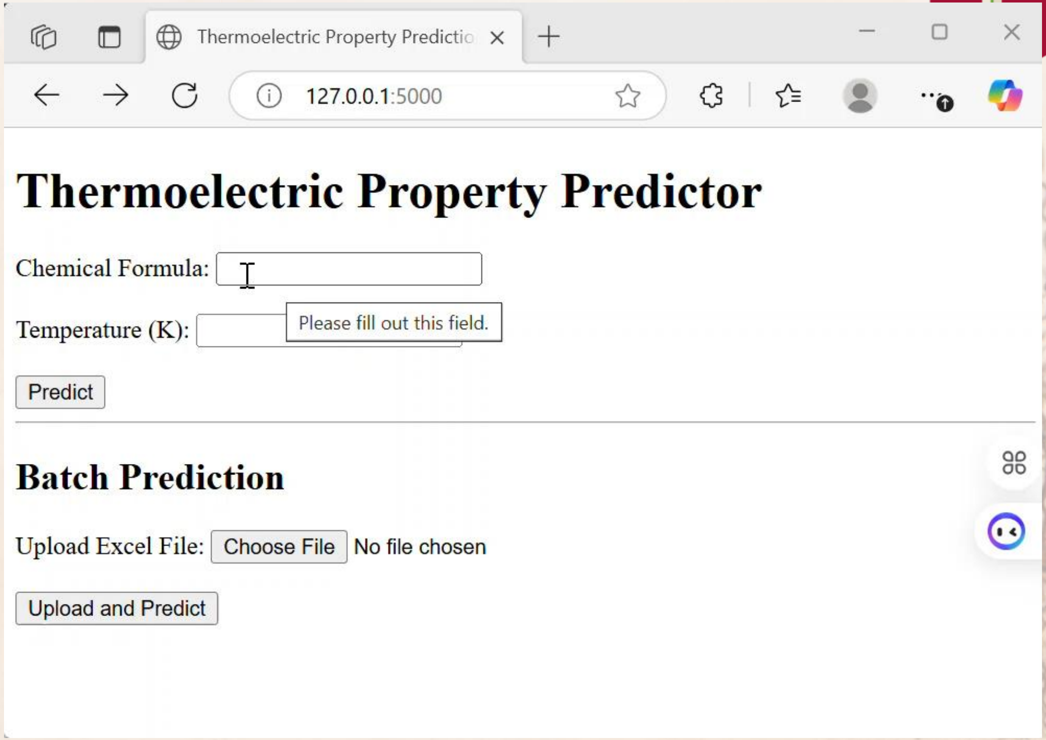
- It supports real-world applications: from **disease modeling** and **neuroprosthetics** to **spiking AI systems** and **robotics**.

Understanding spiking neuron dynamics is not just academic but foundational to the future of AI, healthcare, and human-machine symbiosis.

Recent Works

Physics and Materials Science

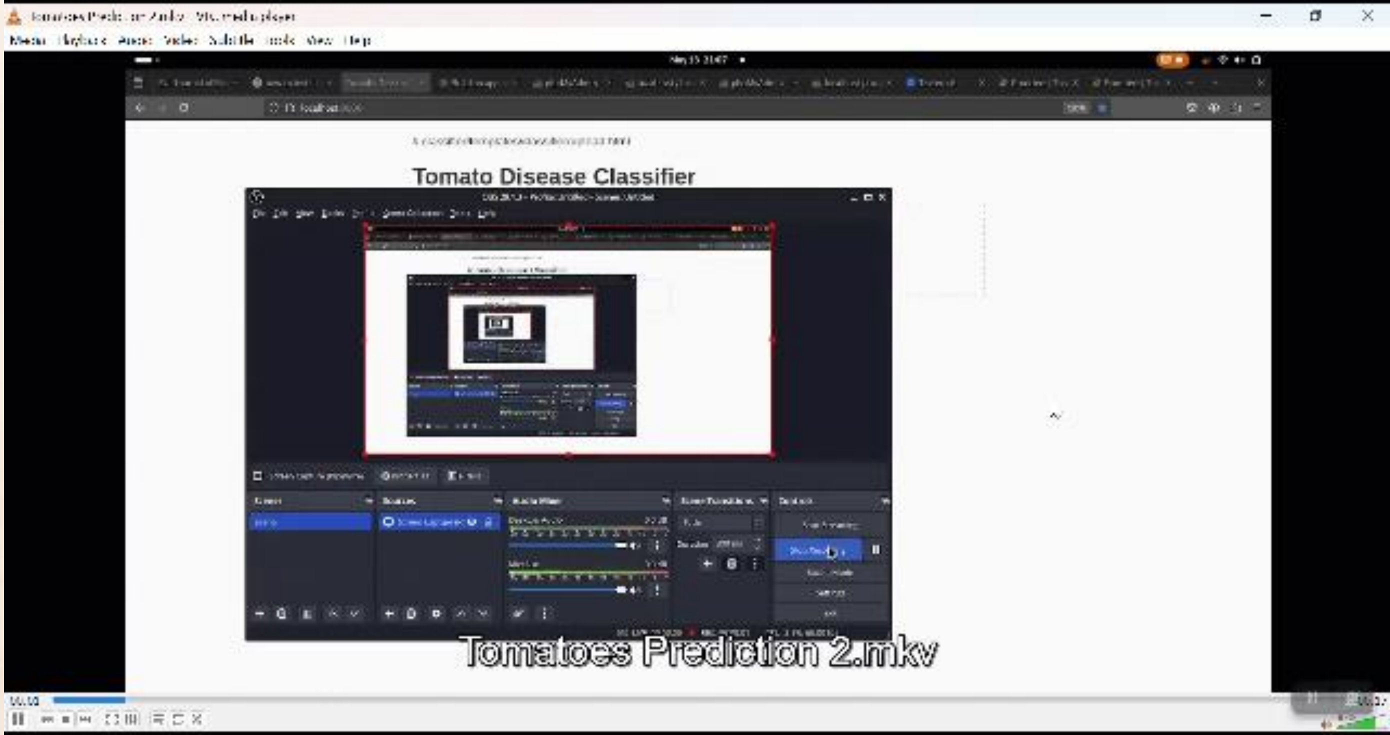
Featured Research: Thermoelectric Property Prediction from Chemical Formulas



Recent Works

Featured Computer Vision Deep Learning for Tomatoes Disease Prediction

(My M.Sc Student's Work)



Myself & Prof Larry Abbott (author: Theoretical Neuroscience book) at isiCNI 2018, Cape Town, SA.

Supporting stakeholders & Partners

EDO STATE UNIVERSITY UZAIRUE
QUALITY EDUCATION FOR DEVELOPMENT

BENSON IDAHOSA UNIVERSITY

CERTIFICATE OF COMPLETION

This certifies that

Akinola Olayinka

is a **LabXchange Teacher Ambassador.**

The participant has successfully completed 20 hours of professional learning through the completion of the LabXchange Teacher Ambassador Program.

LabXchange
Learning without limits.

Andrew Minor 29/02/24
LabXchange, Harvard University

https://www.labxchange.org/akinolaO

HARVARD Online

INTERNATIONAL BRAIN
IBRO
RESEARCH ORGANIZAT

AUCHI POLYTECHNIC AUCHI
BRAIN FOR DEV.

JÜLICH
Forschungszentrum

ICTP
The Abdus Salam
International Centre for Theoretical Physics

UNESCO
United Nations
Educational, Scientific and
Cultural Organization

EAIFR
ICTP - East African Institute
for Fundamental Research
under the auspices of UNESCO

Utrecht University

Centre for Computational Science

isiCNI
STIFUG

University of Colorado Boulder

PhET Global

Yidan Prize
一 丹 獎

PhET
INTERACTIVE SIMULATIONS
University of Colorado Boulder

Centre for Computational Science

LUX TERRA LEADERSHIP FOUNDATION

IBRO Simons Imbizo

https://github.com/solayinka
a/BioRTC-Simons-
Presentation

“The brain is wider than the sky.”

— Emily Dickinson

Thank you for listening