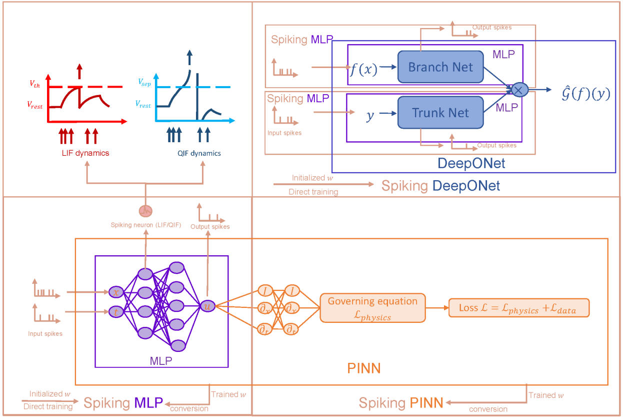
Quadratic Integrate-and-Fire Neurons as Differentiable Units for Scientific Machine Learning

Ruyin Wan George Em Karniadakis Panos Stinis

March 24, 2026

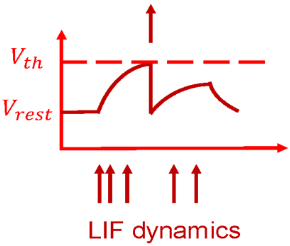
Spiking neuron networks (SNNs)

- Spiking neural networks (SNNs) process information through discrete spike events rather than continuous activations.



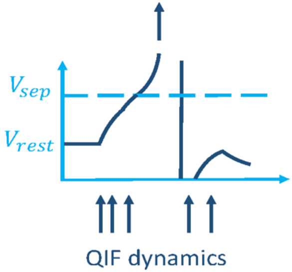
Training of SNNs

- Exact spike gradient $\in \{0, +\infty\}$ does not enable learning.
- Spike-time gradient: gradient of spike time is taken, giving a continuous function.
- Request: each neuron must fire ONCE AND ONLY ONCE for a gradient.
- Problem: relies on *ad hoc* measures to deal with spike (dis-)appearances and gradient divergence.



Why QIF Neurons?

- QIF neurons possess inherently smooth dynamics that allow the variation of membrane potential $\dot{V}(t)$ to grow unbounded during spike generation.
- Small perturbations in synaptic weights w/input spike times t influence only the precise timing of a spike rather than its occurrence.
- *Pseudospike* and *pseudodynamics* deal with any output spike time shifted outside the trial.
- Continuous and non-disruptive spike timing behavior yields smooth gradient of output spikes with respect to w and t, essential for gradient-based optimization and learning.



Spiking Neurons: LIF vs QIF dynamics

- **LIF (leaky integrate-and-fire) equation:**

$$\tau_m \dot{V}(t) = -\lambda\big(V(t) - V_{\text{rest}}\big) + I(t) \tag{1}$$

- **QIF (quadratic integrate-and-fire) equation:**

$$\dot{V}(t) = V(t)\big(V(t) - 1\big) + I(t) \tag{2}$$

- Input current:

$$I(t) = I_0 + \tau_m \sum_i w_i \sum_{t_i} \delta(t - t_i) \tag{3}$$

- **LIF dynamics:** spike emitted once membrane potential V exceeds threshold V_{th}, then V reset to the resting state V_{rest}.
- **QIF dynamics:** spike emitted once membrane potential V surpasses separation point V_{sep}, V tends to $+\infty$ and after spike, V reset to $-\infty$.

Core mechanism: smooth spike dynamics

Phase representation:

$$\phi = \Phi(V) = \frac{\tau_m}{\sqrt{I_0 - \frac{1}{4}}} \left[\arctan\left(\frac{V - \frac{1}{2}}{\sqrt{I_0 - \frac{1}{4}}} \right) + \frac{\pi}{2} \right], \quad \dot{\phi} = 1$$

Spike interaction:

$$\phi^+ = H_w(\phi^-) = \Phi\Big(\Phi^{-1}(\phi^-) + w\Big)$$

Output spike time:

$$t_{sp} = t_i + \phi_\Theta - H_w(\phi_0 + t_i)$$

Key result:

$$\frac{\partial t_{sp}}{\partial w}, \; \frac{\partial t_{sp}}{\partial t} \quad \text{are continuous}$$

Making it work for SciML

1. Regression output

- spike time alone is restrictive
- use **difference between spike times of two neurons**

$$f(x) = t_2 - t_1$$

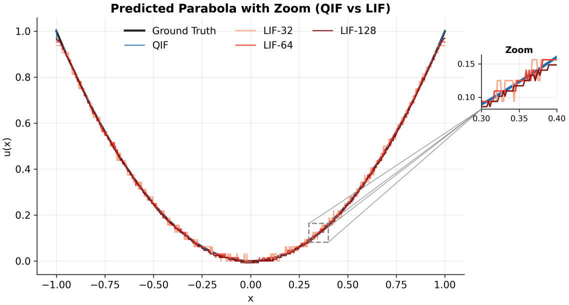
2. Input encoding

- direct transformation to spike times (cheap + differentiable)

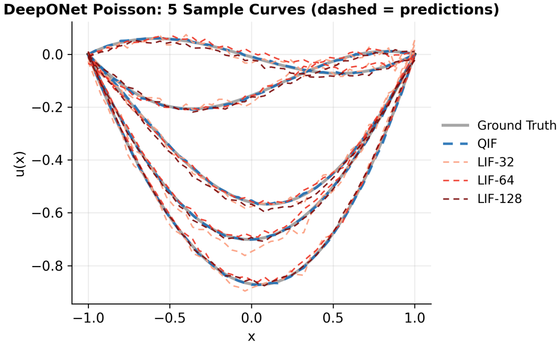
Result

Event-driven SNN becomes compatible with DeepONet, PINNs

Results I: regression + operator learning



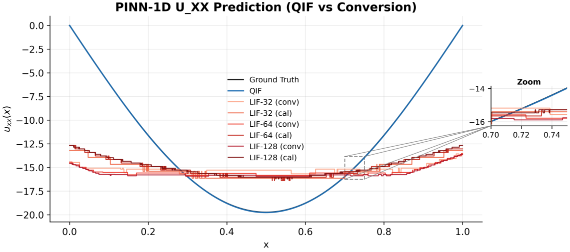
Function regression



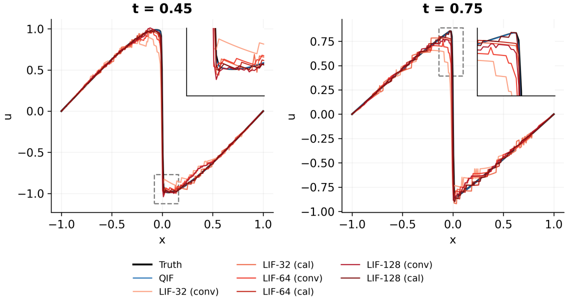
DeepONet (Poisson)

- Smoother prediction
- Better approximation accuracy

Results II: physics-informed learning



1D PINN derivatives (Poisson)



Burgers equation

- QIF gives **correct derivative**
- QIF aviods over/undershoots

Takeaway

QIF is especially powerful when **(higher-order) derivatives matter**

Key takeaways and future work

- QIF neuron allows smooth and differentiable dynamics in SciML tasks through phase representation.
- Both smoothness and accuracy are promising across MLPs, DeepONets and PINNs.
- Future work: Neuromorphic hardware implementations for QIF-based SNN and exploring their capabilities in large-scale, event-driven computation for physics-informed and operator-learning tasks.

Thank you

Questions welcome