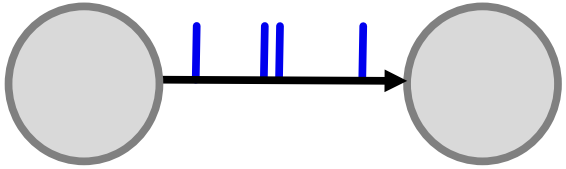
Delays everywhere: Lessons from Neuromorphics and Vanilla Machine Learning

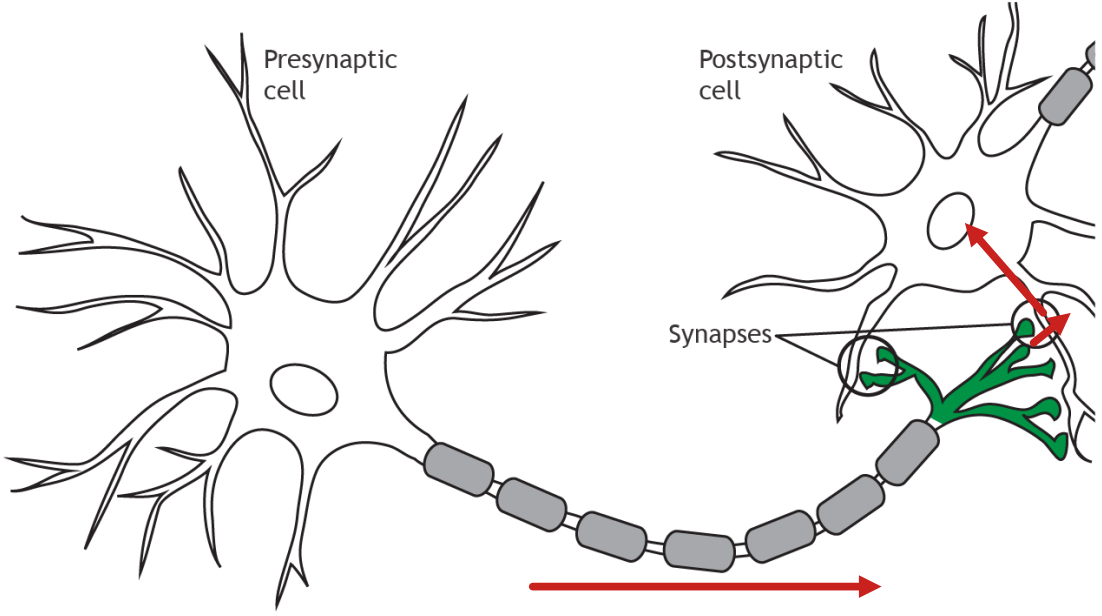
Neuromorphic Hardware and Algorithms Workshop
June 10, 2026

Laura Kriener

EIS - LAB
institute of neuroinformatics
ETH zürich
University of Zurich[UZH]

Spike transmission delays in biology

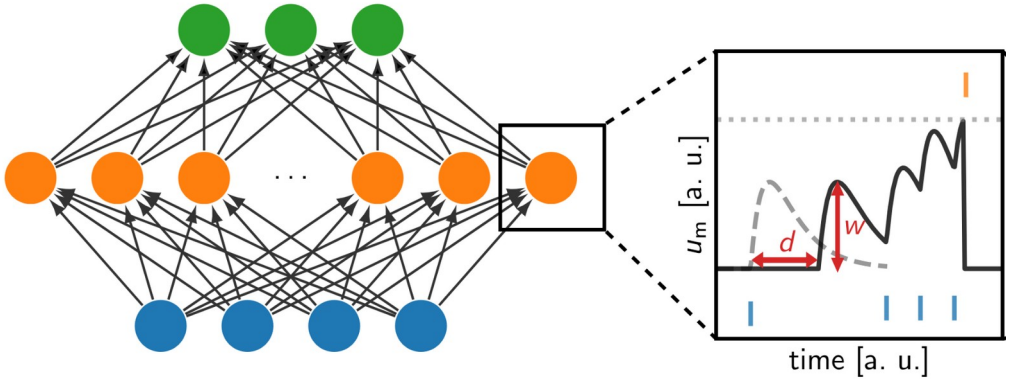




The usual question: Is it a bug or a feature?

Figure taken from Casey L. Henley (CC-BY-NC-SA) at https://openbooks.lib.msu.edu/neuroscience/chapter/the-neuron/

Delays provide computational advantages



Does this hold on neuromorphic hardware?

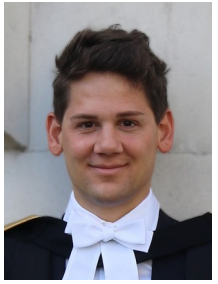
- Feed-forward SNNs + delays can replace RNNs
- Better performance with fewer parameters:
  D'Agostino et al., 2024 and Hammouamri et al., 2024

Part 1: Training delays on neuromorphic hardware

- Fast and energy-efficient neuromorphic deep learning with first-spike times

  J. Göltz & L. Kriener et al., 2021

  https://www.nature.com/articles/s42256-021-00388-x

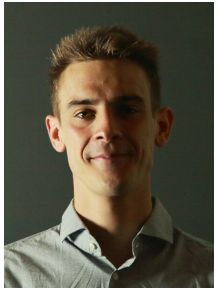


Julian Göltz

- DelGrad: Exact event-based gradients in spiking networks for training delays and weights

  J. Göltz & J. Weber & L. Kriener et al., 2025

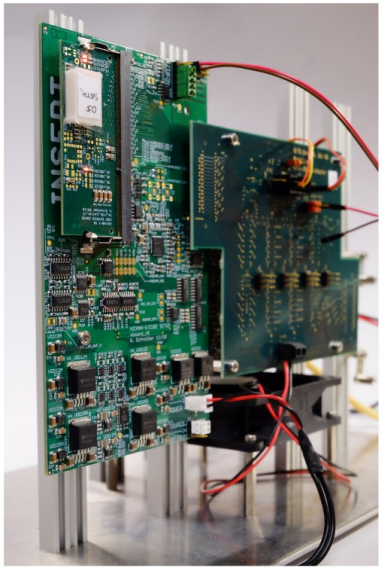
  https://www.nature.com/articles/s41467-025-63120-y

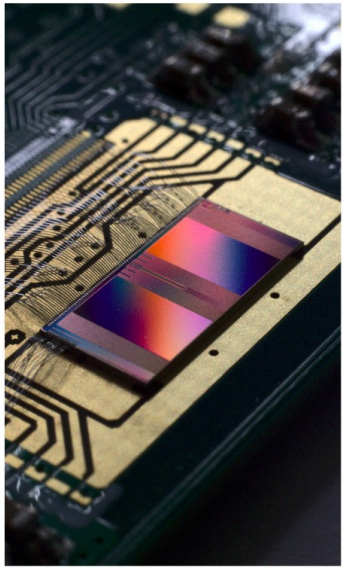


Jimmy Weber

The setup

- BrainScaleS-2 hardware
- Developed by Electronic Vision(s) Group in Heidelberg
- Mixed-signal chip
- 1000x accelerated compared to real-time





Pehle et al., 2022 & Billaudelle et al., 2020

Training SNNs (on hardware) with backprop

Problem: Did the neuron spike? → Discrete output → Bad gradients

Commonly used approach:

- Surrogate gradient (Neftci et al., 2019)
- BPTT-based
- **Difficult hardware requirements:**
  - Recording of all membrane voltages
  - Storing and transporting all of that

Spike-time based approaches:

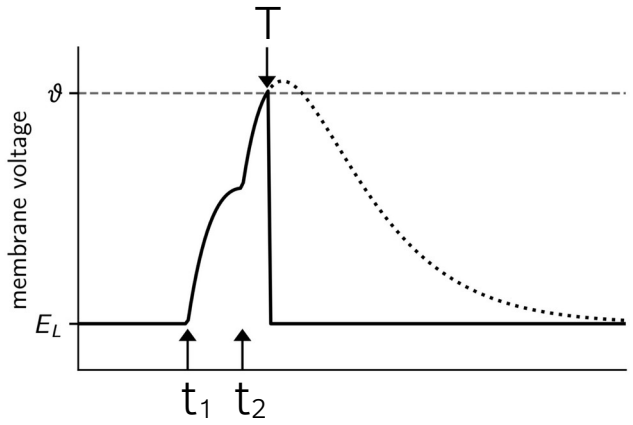
- "When did the neuron spike?"

  → continuous output → gradients ok
- **Easier hardware requirements:**
  - Only spike times recorded
  - Much less data to transport

“Just solve for T”

Membrane voltage is sum of PSPs:

$$u_{\mathrm{m}}(t) = E_\ell + \sum_{\text{input times } t_i} \mathrm{PSP}\,(t - t_i)$$



Spike if voltage is equal to threshold:

$$u_{\mathrm{m}}(T) = \vartheta$$

Can be solved for T under certain conditions:

- LIF neurons
- Specific ratio of time constants

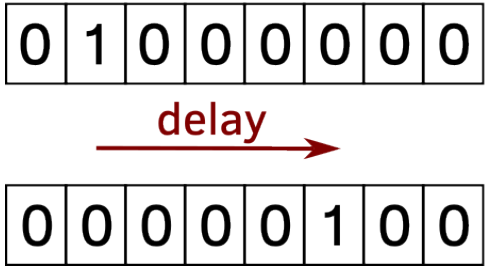
$$T = \tau_{\mathrm{s}} \left\{ \frac{b}{a_1} - \mathcal{W}\left[-\frac{g_\ell \vartheta}{a_1} \exp\left(\frac{b}{a_1}\right) \right] \right\}$$

$$\frac{\partial T}{\partial w_i}(\mathbf{w}, \mathbf{t}, T) = -\frac{1}{a_1} \frac{1}{\mathcal{W}(z)+1} \exp\left(\frac{t_i}{\tau_{\mathrm{s}}}\right) (T - t_i)$$

$$\frac{\partial T}{\partial t_i}(\mathbf{w}, \mathbf{t}, T) = -\frac{1}{a_1} \frac{1}{\mathcal{W}(z)+1} \exp\left(\frac{t_i}{\tau_{\mathrm{s}}}\right) \frac{w_i}{\tau_{\mathrm{s}}} (T - t_i - \tau_{\mathrm{s}})$$

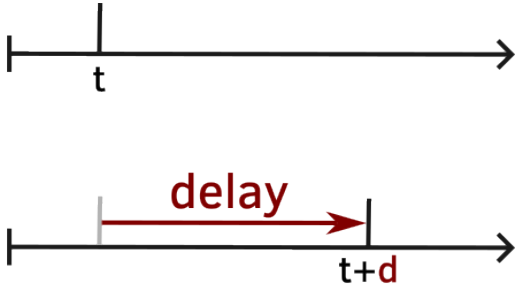
Göltz & Kriener et al., 2021

Training delays



Time-stepped setting:

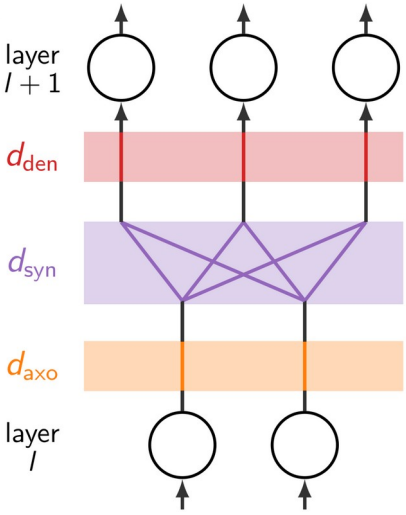
- Delay results in discrete shift
- Workaround via convolutions



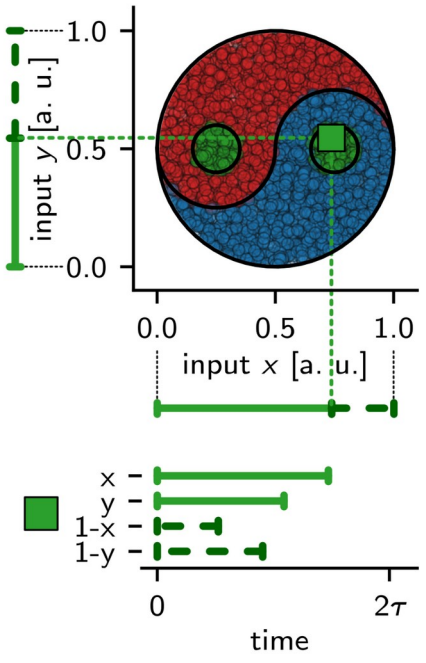
Spike-time-based setting:

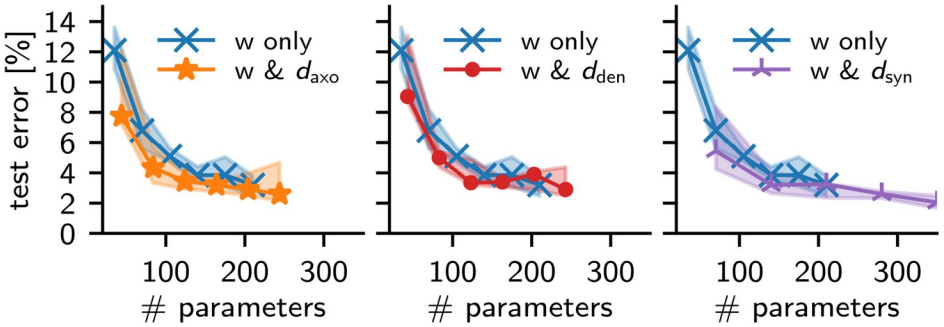
- Delay is addition to spike time
- Gradients to learn the delays are easy to include

Different types of delays



Kriener et al., 2022





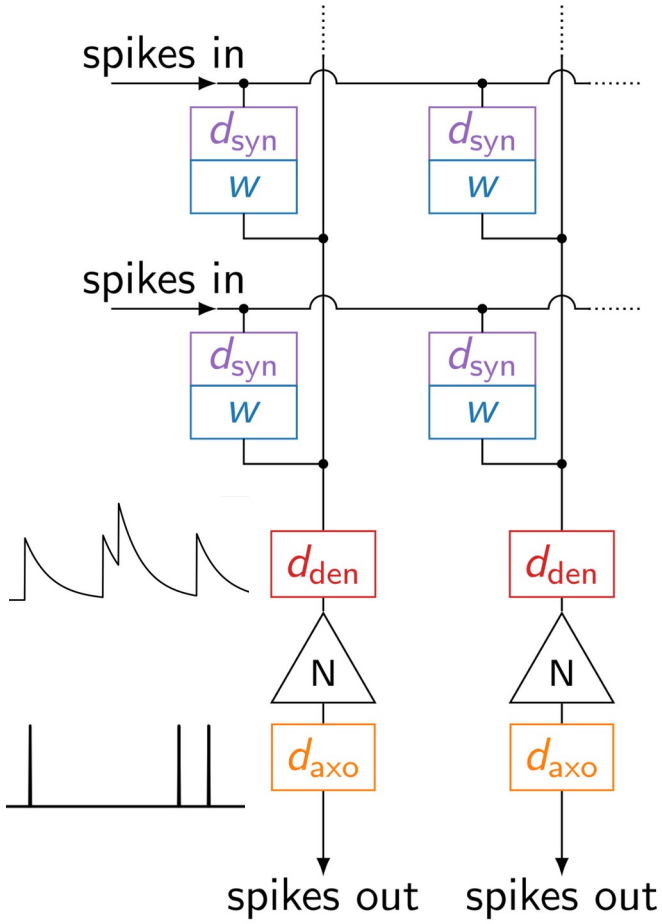
- Delays enhance parameter space
- Better performance for fewer parameters
- Delay type does not matter

Göltz & Weber & Kriener et al., 2025

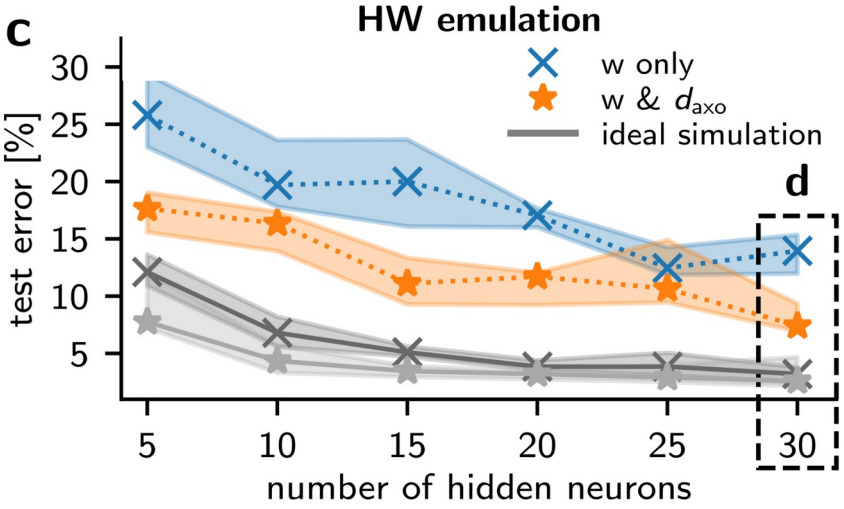
Where to put the delays on hardware?

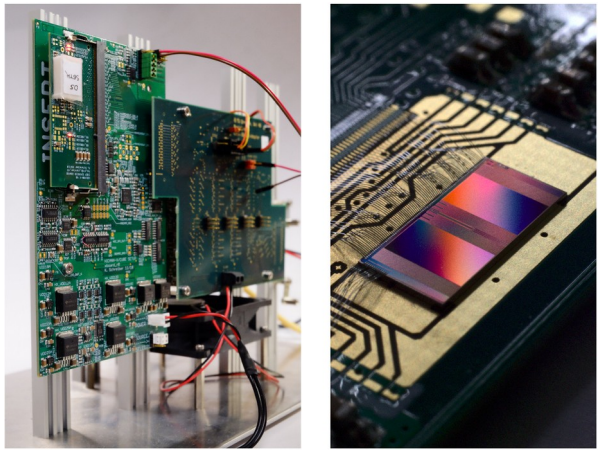
Delay type matters a lot:

- Synaptic delays scale quadratically
- Dendritic delays might require delaying an analog quantity (synaptic current)
- Axonal delays can delay binary signal



Training weights and delays on BrainScaleS-2





- Small networks on hardware suffer from noise effects
- Bigger benefit of delays than in simulations

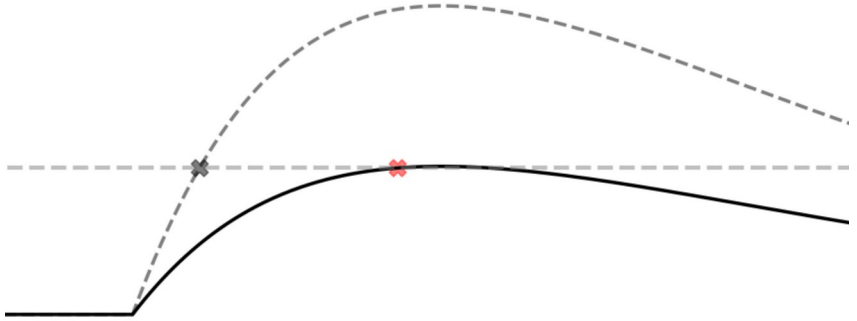
  → Delays add robustness against noise

Göltz & Weber & Kriener et al., 2025

Hypothesis: Delays allow steep threshold crossing

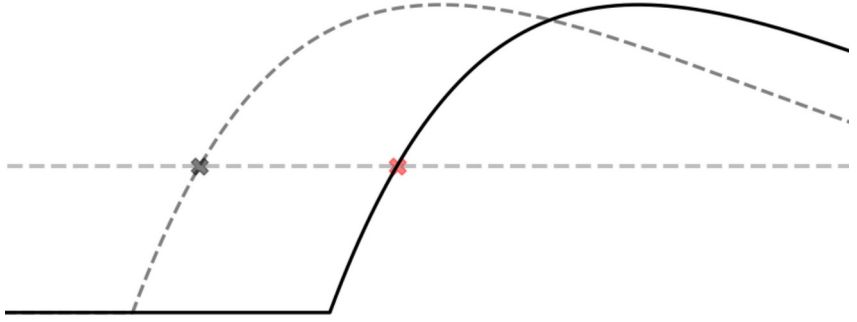
Weight-only:

- Decrease weight to delay output spike
- Shallow threshold crossing



With delays:

- Use delay instead of weight
- Steep threshold crossing



Does the delay advantage transfer to hardware?

Yes!

And not only that, it also stabilizes against hardware noise!

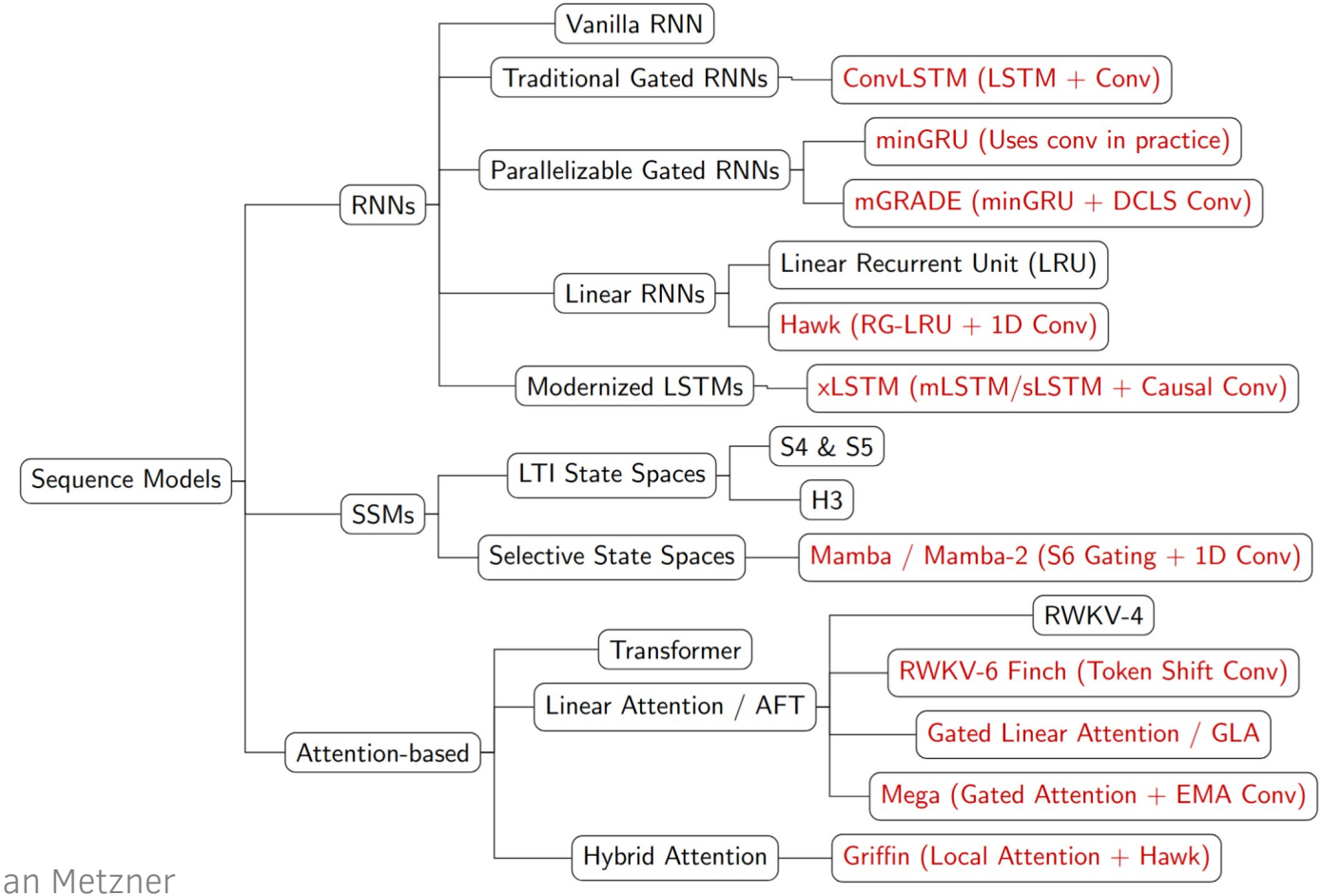
A little excursion...

Machine learning models always contain more components than described in the main paper:

```
self.v_conv1d = ShortConvolution(
    hidden_size=self.value_dim,
    kernel_size=conv_size,
    bias=conv_bias,
    activation='silu',
)
```

https://github.com/fla-org/flash-linear-attention/blob/main/fla/layers/gated_deltanet.py

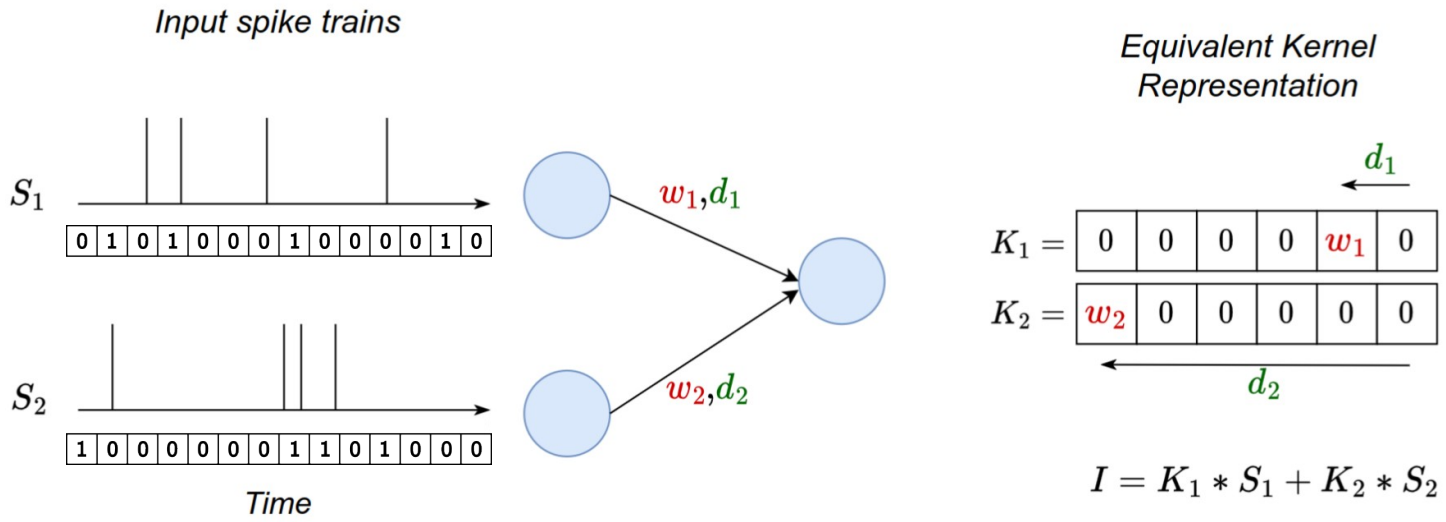
Conv1D in many sequence model backbones!



Created by Christian Metzner

What does Conv1D have to do with delays?

"DCLS: Dilated Convolution with Learnable Spacings"



Spiking world: sparse S_i and K_i

Machine learning world: dense S_i and K_i

Adapted from Hammouamri et al., ICLR 2024

Understand and use for efficient model!

- Why is this combination of delays and recurrency so prevalent?
- What is a minimal & efficient combination of recurrency and delays for powerful sequence processing?

Tristan Torchet

- mGRADE: Minimal Recurrent Gating Meets Delay Convolutions for Lightweight Sequence Modeling

  T. Torchet & C. Metzner et al., 2025

  https://arxiv.org/abs/2507.01829

Christian Metzner

Minimally **G**ated **R**ecurrent **A**rchitecture with **D**elay **E**mbedding

Gated recurrency:

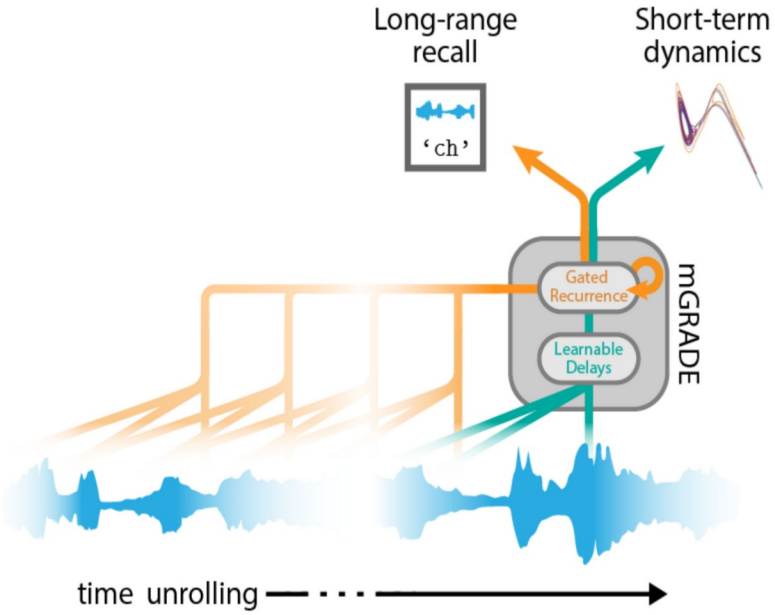
- minGRU for efficiency
- Long-range selectivity and memory
- **But:** slow-feature bias

Delay-embedding layer:

- DCLS for sparse and learnable kernels

  → parameter efficient

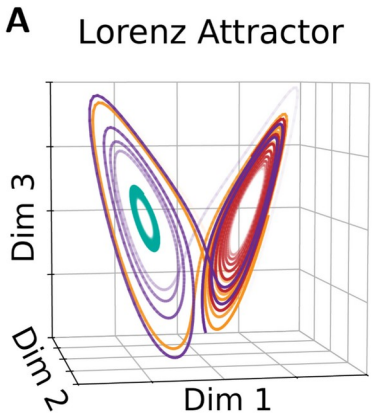
- “Pre-process” fast features before entering recurrent unit



Delay embeddings are powerful

Next-step prediction of **unobserved dimension:**

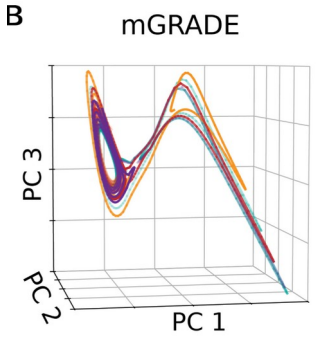
- mGRADE: ✔
- minGRU: ✘

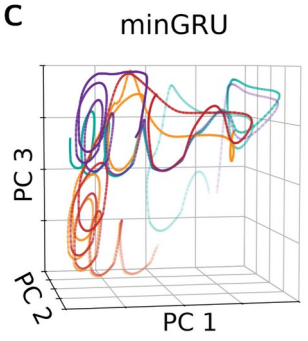


Taken's Embedding Theorem (1981):

- d-dimensional dynamical system
- m = 2d + 1 delays allow reconstruction

  from single observed dimension

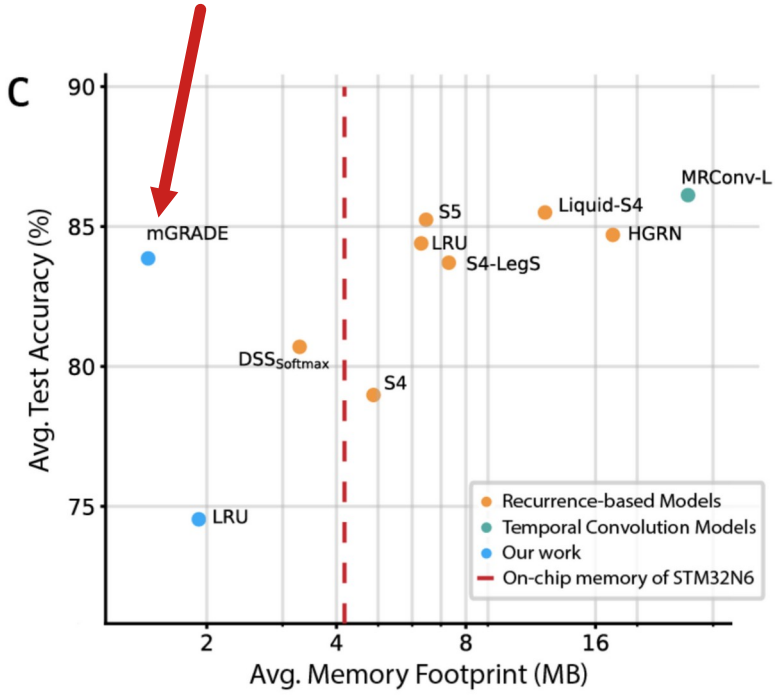




mGRADE is versatile and parameter efficient

Long Range Arena (LRA) benchmark:

- Collection of multi-time-scale benchmarks
- mGRADE shows competitive performance with much larger models



(Avg. Test Accuracy excluding PathX)

Summary

Part 1:

- Spike-time-based exact gradients for LIF make delay learning easy
- Delays increase performance in simulation and on hardware
- On hardware delays seem to stabilize against noise

Part 2:

- Learnable delays complement classical gated-recurrent sequence models
- The combination of both results in versatile and parameter-efficient hybrid model

Fast and energy-efficient neuromorphic deep learning with first-spike times
J. Göltz & L. Kriener et al., 2021
https://www.nature.com/articles/s42256-021-00388-x

DelGrad: Exact event-based gradients in spiking networks for training delays and weights
J. Göltz & J. Weber & L. Kriener et al., 2025
https://www.nature.com/articles/s41467-025-63120-y

mGRADE: Minimal Recurrent Gating Meets Delay Convolutions for Lightweight Sequence Modeling
T. Torchet & C. Metzner et al., 2025
https://arxiv.org/abs/2507.01829

Acknowledgements

People:

Julian Göltz

Jimmy Weber

Tristan Torchet

Christian Metzner

Melika Payvand

Mihai A. Petrovici

Labs:

EIS-Lab (Zurich)

NeuroTMA-Lab (Bern)

Electronic Visions (Heidelberg)

Funding:

Swiss National Science Foundation