KirchhoffNet: Hardware Acceleration of ODE-Driven Models

Zhengqi Gao[1], Fan-keng Sun[1], Ron Rohrer[2], and Duane S. Boning[1*]

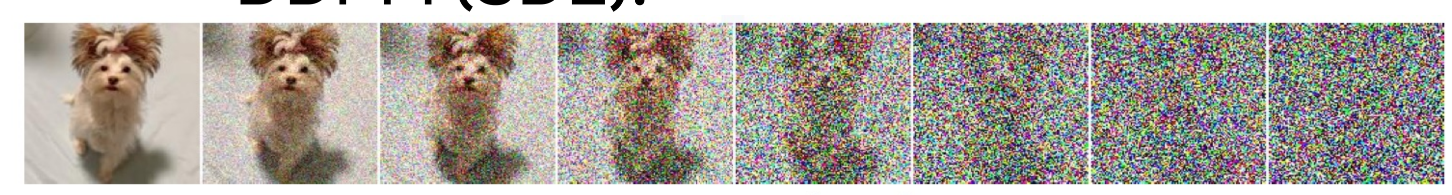
[1]Dept. EECS MIT, [2]Dept. ECE CMU, [*]accepted by ICCAD'24

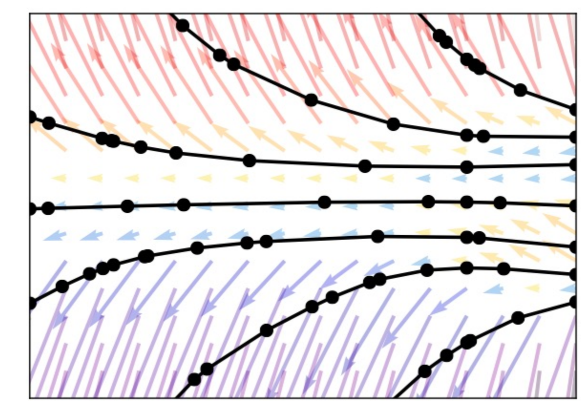
Arxiv Homepage

Introduction

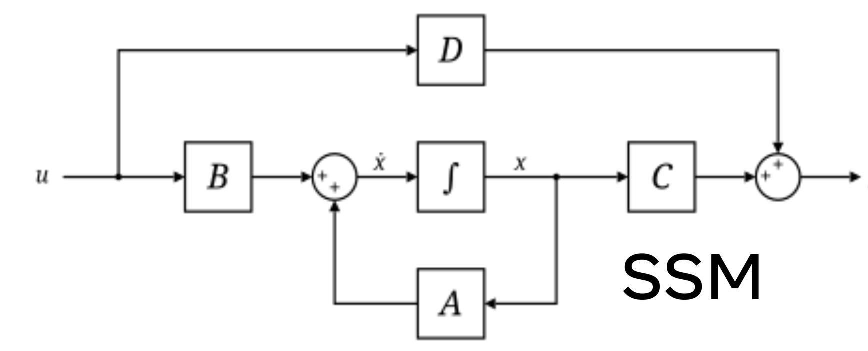
- ODE-Driven Neural Networks attract huge interests
  - Neural ODE (NODE), State-Space Model (SSM)
  - Denoising Diffusion Probabilistic Model (DDPM)
- These models are usually time consuming to run on GPUs
  - Reason: Discretization along 'time' axis; Step-by-step marching
  - Here 'time' is a unitless numerical variable, i.e., t = 1 is not 1 sec
- Motivation: make 'time' a real physical quantity, acceleration!

DDPM (SDE):





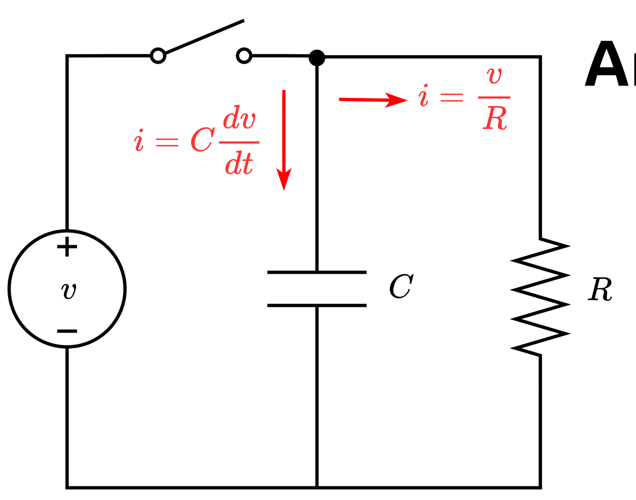
NODE $\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$



SSM

$$x'(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$
$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t)$$

KirchhoffNet: Kirchhoff Current Law (KCL) Enabled Computing



An illustrative RC example from high school

- Prep: Closed switch for long time; Voltage at black dot is v
- Run: Open the switch; KCL states net current flow = 0
- R for Resistor with i=v/R; C for Capacitor with i = Cdv/dt
- RC ODE: Cdv/dt+v/R = 0; dv/dt = av, where a = -1/RC

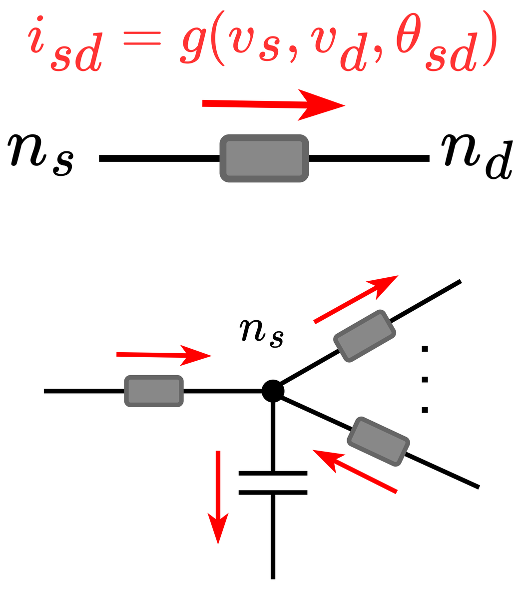
Property: Adjust Capacitor C to be smaller, then on-chip forward can be very fast

$$v(t+\Delta t) = v(t) + \frac{\Delta t}{C}\Big[\sum_{\text{into } n_s} g(v_s, v_d, \theta_{sd}) - \sum_{\text{out of } n_s} g(v_s, v_d, \theta_{sd})\Big]$$

Shrink C by Const ⇔ Shrink t by Const

(C = 1 Farad, T = 1s) ⇔ (C = 1 pF, T = 1 ps)

$i_{sd} = g(v_s, v_d, \theta_{sd})$

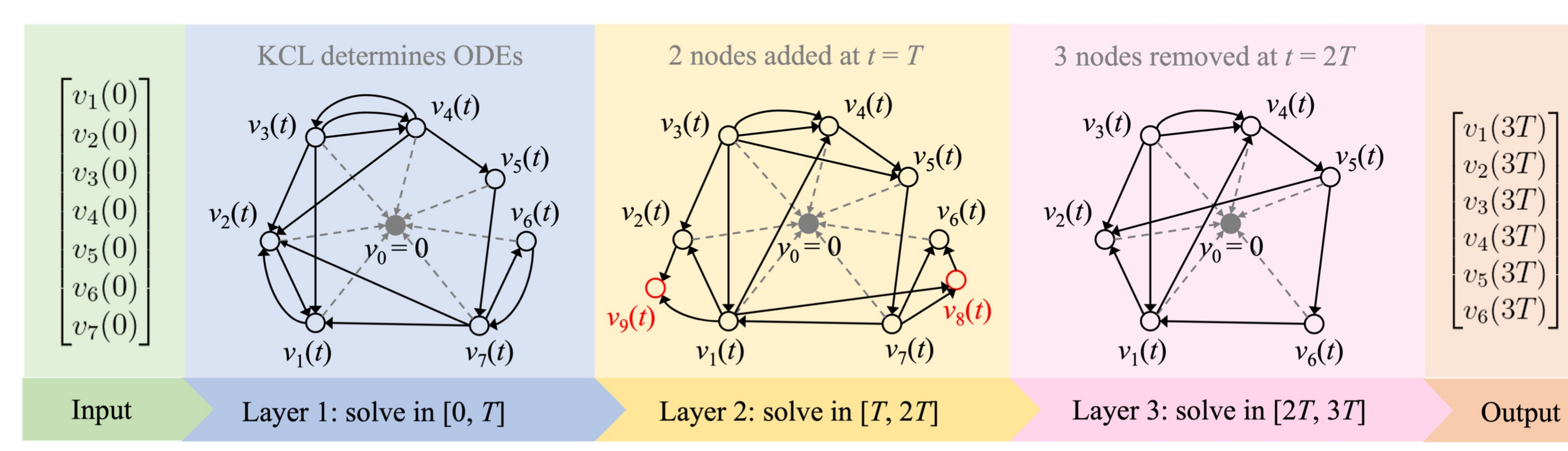


Extend to complex ODE right-hand side

- Assume a nonlinear I-V device/element g
- Write KCL for the node shown:

(*) $$C\frac{dv}{dt} = \sum_{n_s \to n_d \text{ flow into } n_s} g(v_s, v_d, \theta_{sd}) - \sum_{n_s \to n_d \text{ flow out of } n_s} g(v_s, v_d, \theta_{sd})$$

- Deriving all nodes yields a set of ODEs

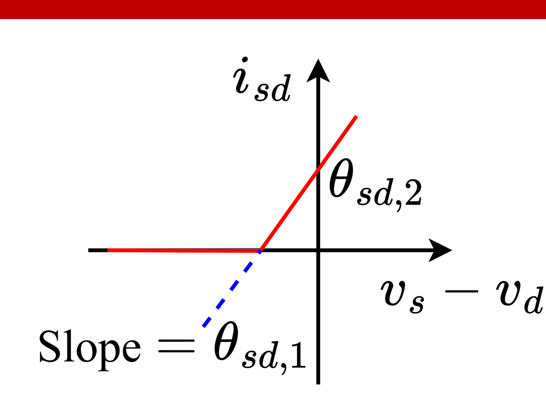


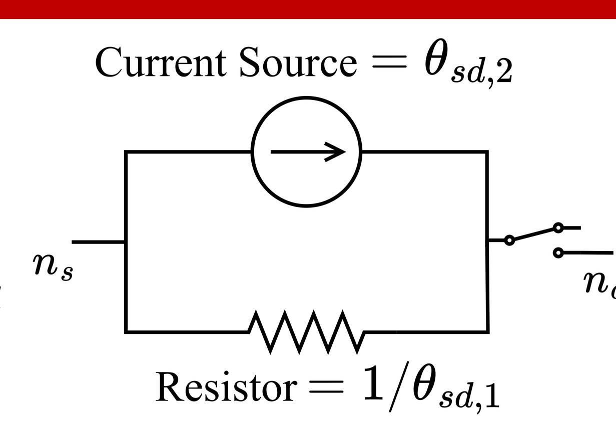
Black solid arrow: non-linear I-V device
Gray dashed arrow: fixed capacitor

T: hyper-parameter controls run-time

Numerical Results

- Example: A ReLU-shape nonlinear I-V
  - Adjustable resistor:
    - MOS working in linear region
    - Gate-controlled voltage as theta
  - Ideal-switch: diode-connected MOS
  - Adjustable current source: Current mirror, paralleled connected MOS
- KirchhoffNet limits ODE form, doesn't have traditional layers (linear, conv, …)
  - NODE on GPU $\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$: can choose f to be a NN made up of linear, conv…
  - KirchhoffNet (*): the ODE right-hand side form determined by circuit topology
- Pytorch-based circuit-level simulation/verification on GPU w/ V100s





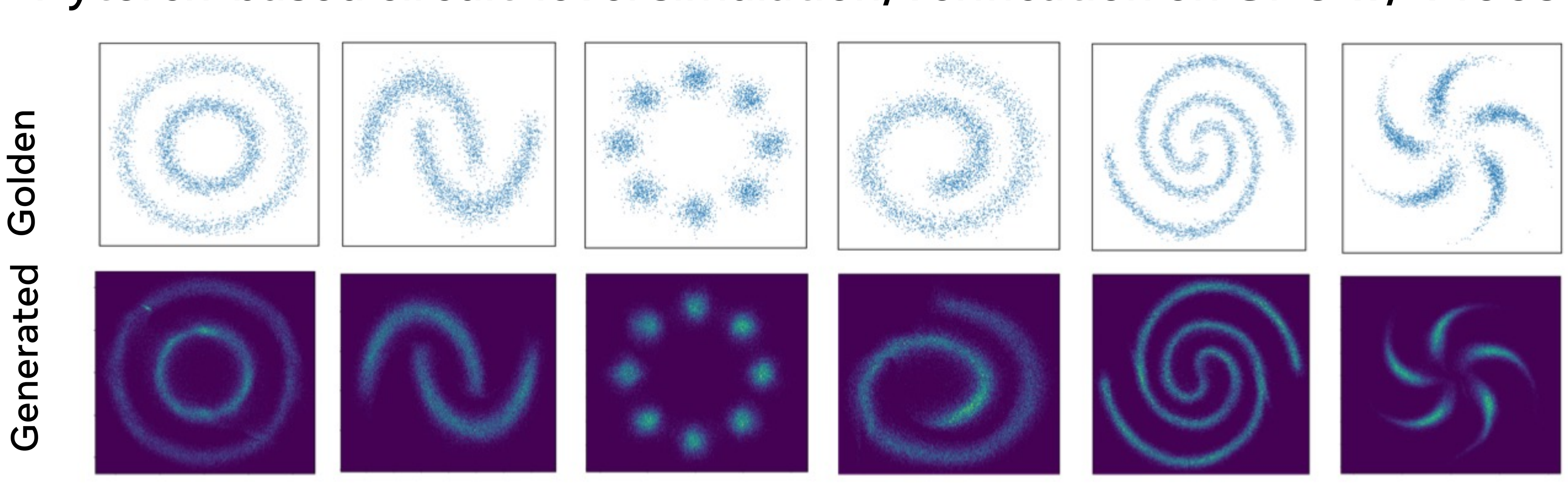
$$g(v_s, v_d, \theta_{sd}) = \text{ReLU}(\theta_{sd,1}(v_s - v_d) + \theta_{sd,2})$$

Table I: Regression Acc comparable to MLP

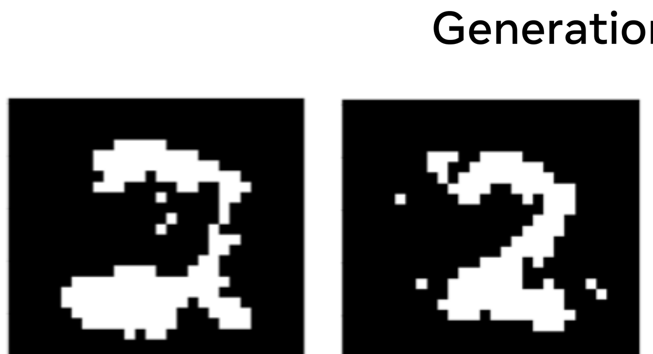
	Friedman	Housing	Diabete
MLP-s	9.97e-4 / 181	1.46e-2 / 391	2.71e-2 / 421
KirchhoffNet-s	8.81e-4 / 168	1.55e-2 / 360	2.91e-2 / 528
MLP-m	1.35e-4 / 561	1.32e-2 / 841	2.61e-2 / 889
KirchhoffNet-m	1.95e-5 / 528	1.25e-2 / 840	3.09e-2 / 840
MLP-l	6.28e-5 / 1596	1.27e-2 / 1551	2.67e-2 / 1581
KirchhoffNet-l	1.11e-5 / 1520	1.16e-2 / 1520	3.48e-2 / 1520

Table II: Image Classification needs meticulous circuit topology design to attain SOTA CNN; but our effort already comparable to SOTA NODE

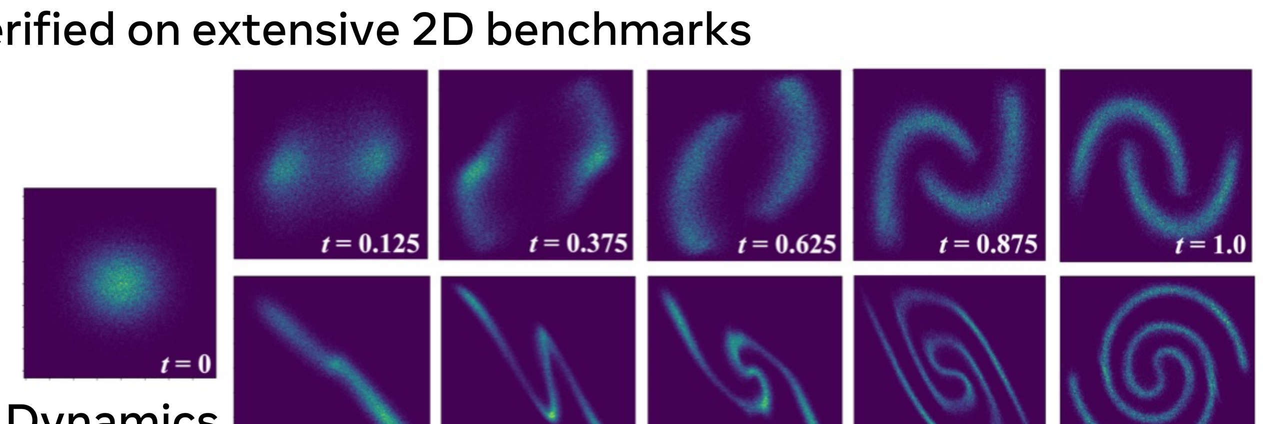
	Model	MNIST	CIFAR-10
Proposed	KirchhoffNet	99.39 ± 0.06	73.41 ± 0.12
Compared Baselines	Neural ODE [34]	96.40 ± 0.50	53.70 ± 0.20
	Aug Neural ODE [34]	98.20 ± 0.10	60.60 ± 0.40
	IL Neural ODE [38]	99.10	73.40
	2nd Neural ODE [38]	99.20	72.80



Generation: verified on extensive 2D benchmarks



Generation on MNIST



Dynamics

Conclusions

- KirchhoffNet accelerates ODE-driven models
- Run slowly on GPU (so is NODE)
- Thus, experiments limited in scale
- The promised fast run-time is on the fabricated hardware, not on GPU
- Future works: (i) Real analog circuit hardware design, (ii) SPICE simulation, (iii) Chip tapeout
- Read the Arxiv paper for more results!