IJCAI 2019 Conference

ANODE: Unconditionally Accurate Memory-Efficient Gradients for Neural ODEs

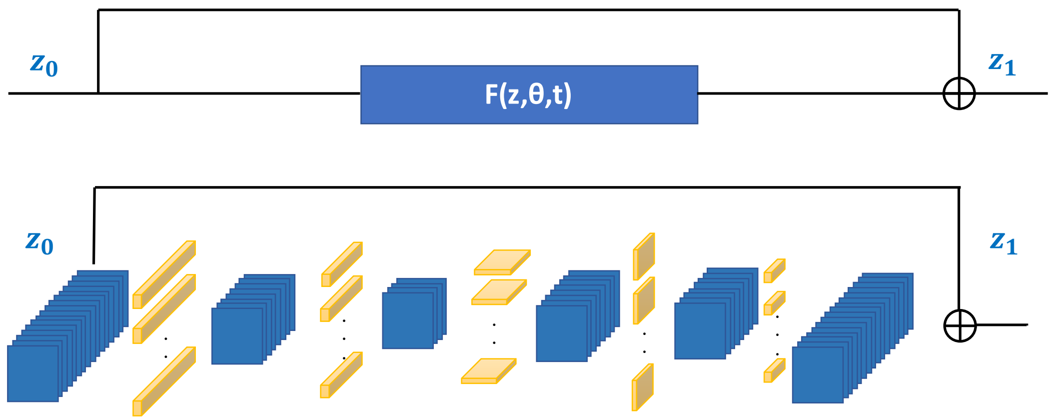
Zhewei Yao, Amir Gholami, Kurt Keutzer, George Biros

University of California, Berkeley

Residual Networks as ODEs

$$z_1 = z_0 + f(z_0, \theta) \qquad \text{ResNet}$$



We can view ResNet as an Euler discretization of a Neural ODE

$$z_1 = z_0 + \int_0^1 f(z(t), \theta) dt \qquad \text{ODE}$$

$$z_1 = z_0 + f(z_0, \theta) \qquad \text{ODE forward Euler}$$

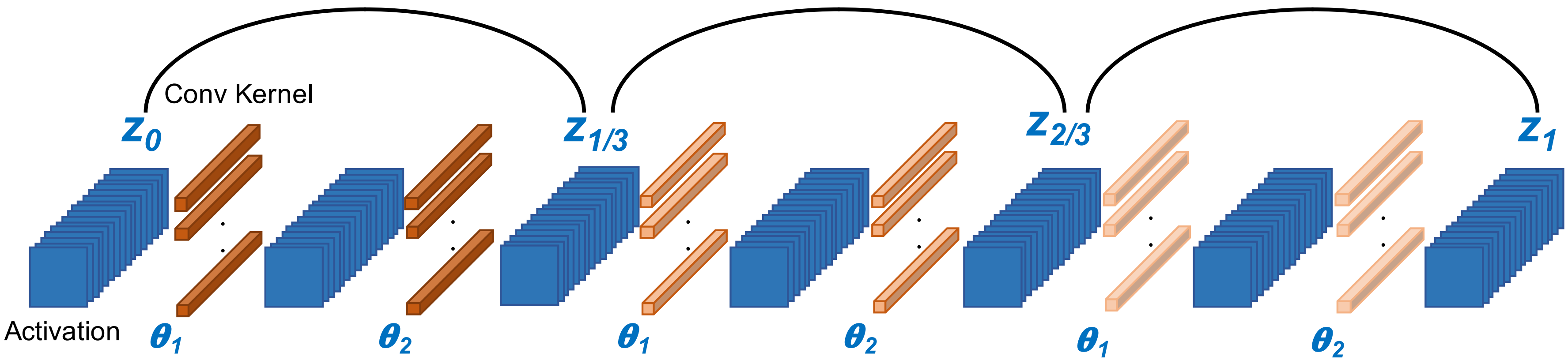
Weinan (2017), Haber (2017), Lu et al (2017), Ciccone et al. (2018), Ruthotto (2018)

Neural ODEs

- In Neural ODEs, the forward solve is equivalent to solving the following integration:

$$z_1 = z_0 + \int_0^1 f(z(t), \theta)dt \qquad \text{ODE}$$

$$z_1 = z_0 + f(z_0, \theta) \qquad \text{ODE forward Euler}$$



- **How do we backpropogate gradients?**

Gradient Backpropogation

- We need to first form the Lagrangian and find its saddle points (KKT conditions). This leads to the following system:

$$
\begin{aligned}
\frac{\partial z}{\partial t} + f(z,\theta) = 0, \quad & t \in (0,1] \\
-\frac{\partial \alpha(t)}{\partial t} - \frac{\partial f}{\partial z}^T \alpha = 0, \quad & t \in [0,1) \\
\alpha_1 + \frac{\partial J}{\partial z_1} = 0, & \\
g_\theta = \frac{\partial R}{\partial \theta} - \int_0^1 \frac{\partial f(z(t),\theta)}{\partial \theta}^T \alpha(t) dt &
\end{aligned}
$$

Gradient Backpropogation

- We need to first form the Lagrangian and find its saddle points (KKT conditions). This leads to the following system:

$$
\begin{aligned}
\frac{\partial z}{\partial t} + f(z,\theta) = 0, \quad & t \in (0,1] \\
-\frac{\partial \alpha(t)}{\partial t} - \frac{\partial f}{\partial z}^T \alpha = 0, \quad & t \in [0,1) \\
\alpha_1 + \frac{\partial J}{\partial z_1} = 0, & \\
g_\theta = \frac{\partial R}{\partial \theta} - \int_0^1 \frac{\partial f(z(t),\theta)}{\partial \theta}^T \alpha(t) dt &
\end{aligned}
$$

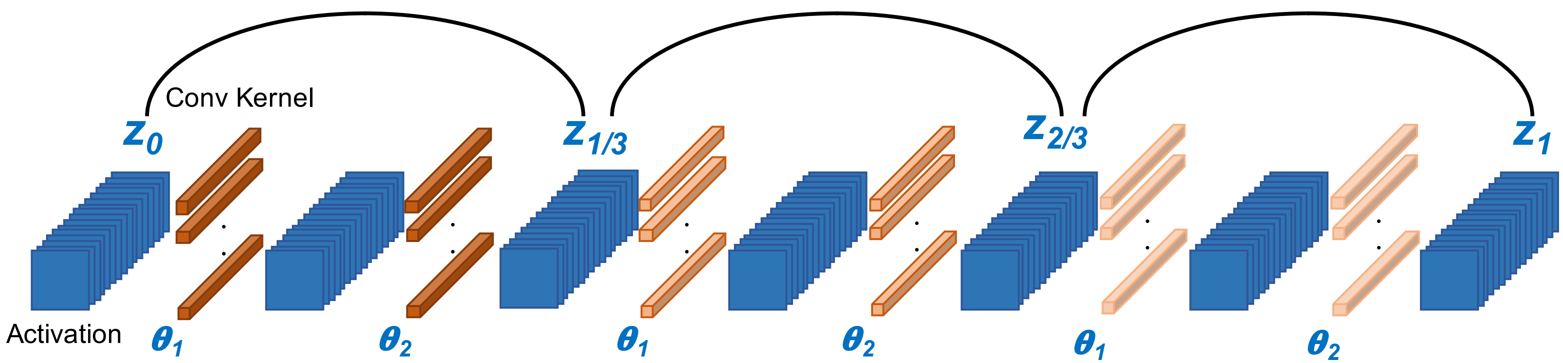
- To backpropogate the gradient we need to store intermediate activations in time z(t) -> **O(LNt)** memory footprint
  - **The memory requirement is increased by a factor of Nt**

Reverse ODE Solve

- A recent solution was proposed by Chen et al. to reverse ODE solve and avoid storing z(t)
  - Reduces memory cost from O(LNt) -> O(L)

But can ODEs be reversed in time?

Given z_0 we can perform forward pass to computer z_1



But can we recover z_0 by only storing z_1 and reversing ODE?

Reversibility of ODEs

- Reversing ODEs is in general ill-conditioned.
- Consider the following example:
- Solving this ODE is **stable in forward mode**

$$\frac{dz}{dt} = -\lambda^2 z(t)$$

$$z_t = z_0 \exp(-\lambda^2 t)$$

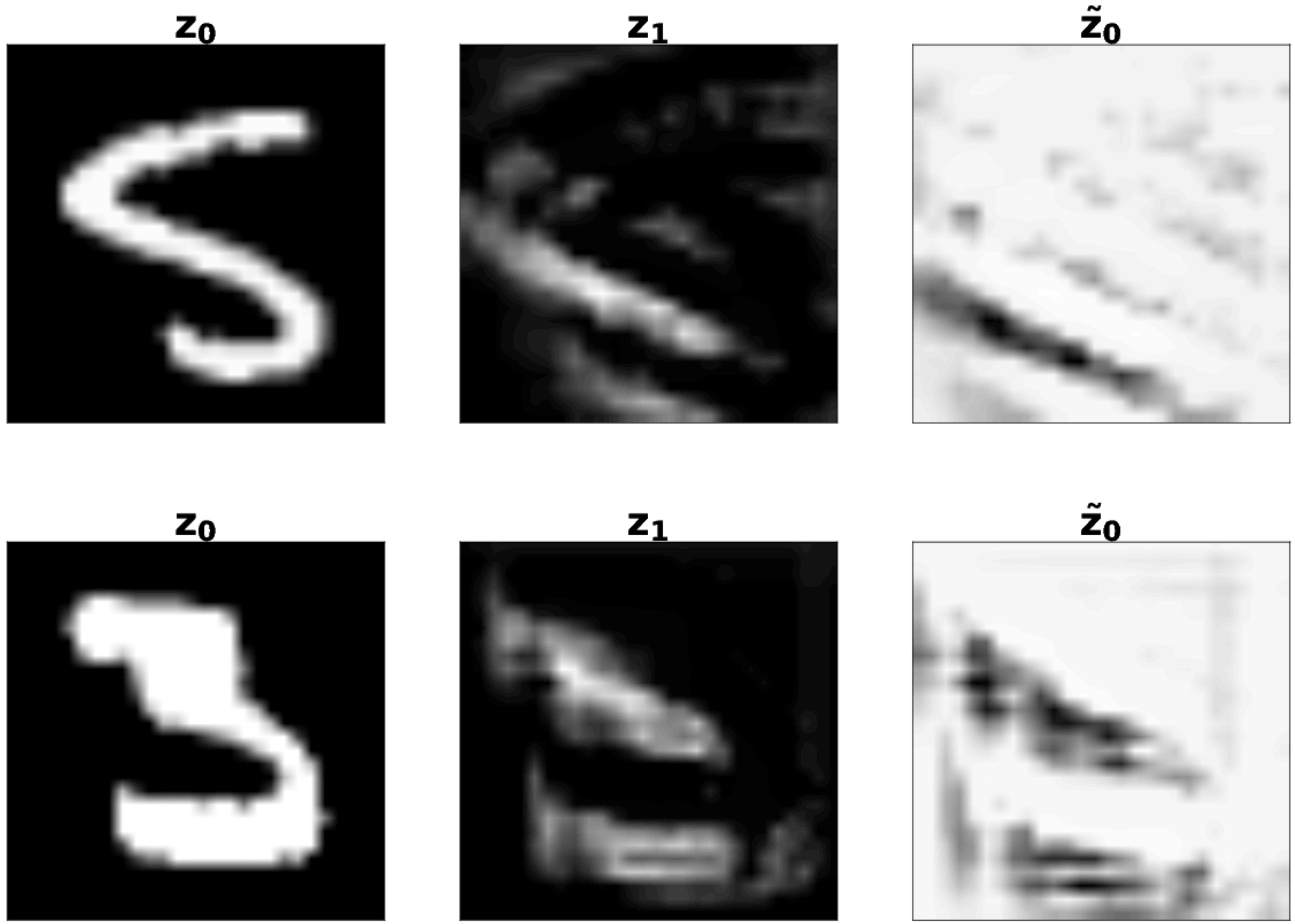
- However, reverse mode solution would **exponentially amplify noise**

$$z_0 = z_t \exp(\lambda^2 t)$$

Irreversibility of ODEs

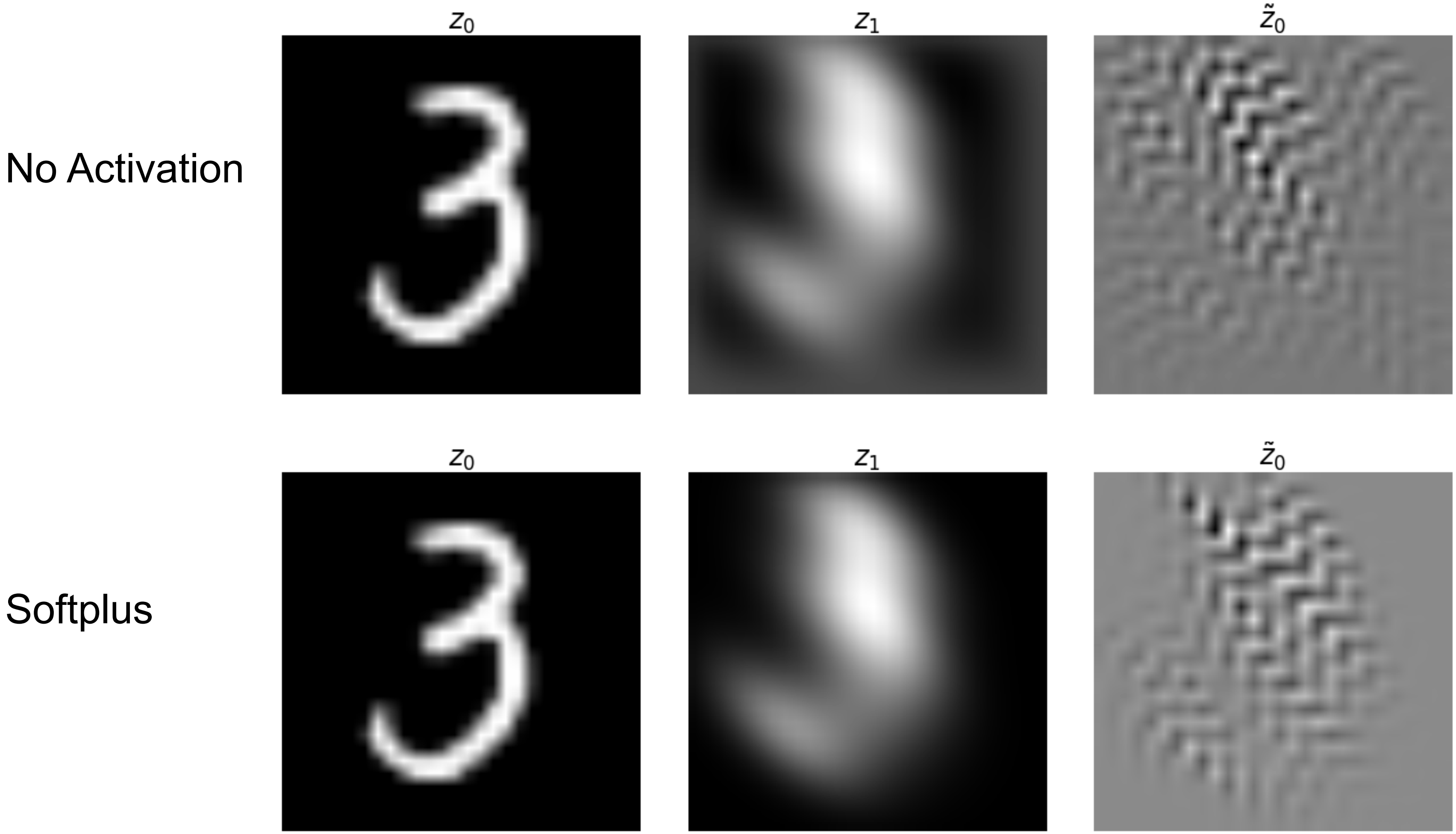
Leaky ReLu

ReLu



Demonstration of irreversibility with Euler solver

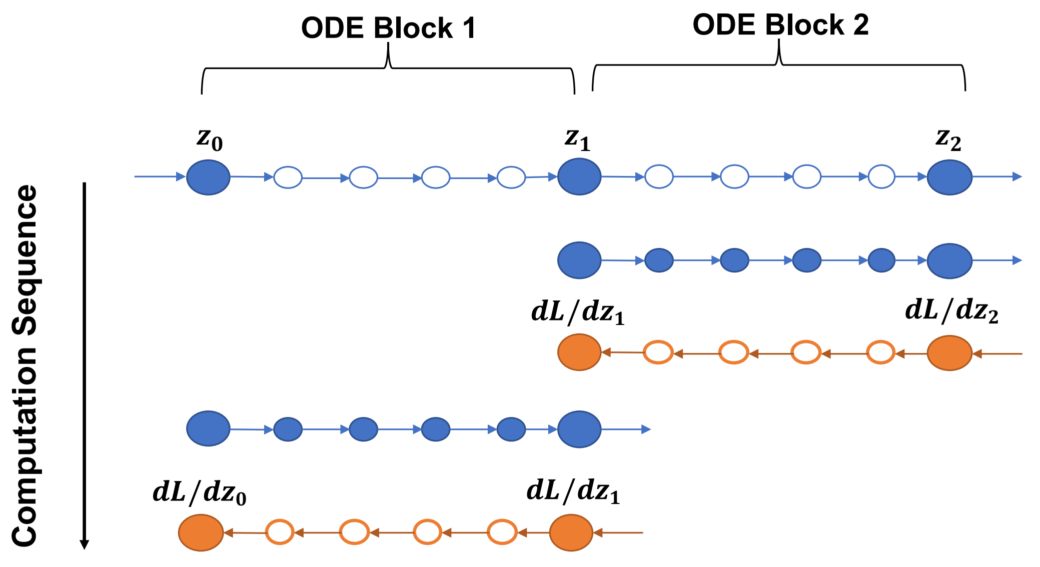
Irreversibility of ODEs



Demonstration of irreversibility with adaptive RK45 solver

ANODE: Addressing Challenges with Neural ODEs

- The memory footprint challenge could be simply addressed via checkpointing
  - O(LNt) to **O(L + Nt)** without the stability issue of Neural ODE



- Cannot use **continuous form** of optimality conditions
  - ANODE uses "**Discretize-Then-Optimize**" approach to obtain correct gradient information

A. Griewank. "Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation". Optimization Methods and software (1992), pp. 35–54.

A. Gholami, K. Keutzer, G. Biros. "ANODE: Unconditionally Accurate Memory-Efficient Gradients for Neural ODEs", arxiv-1902.10298

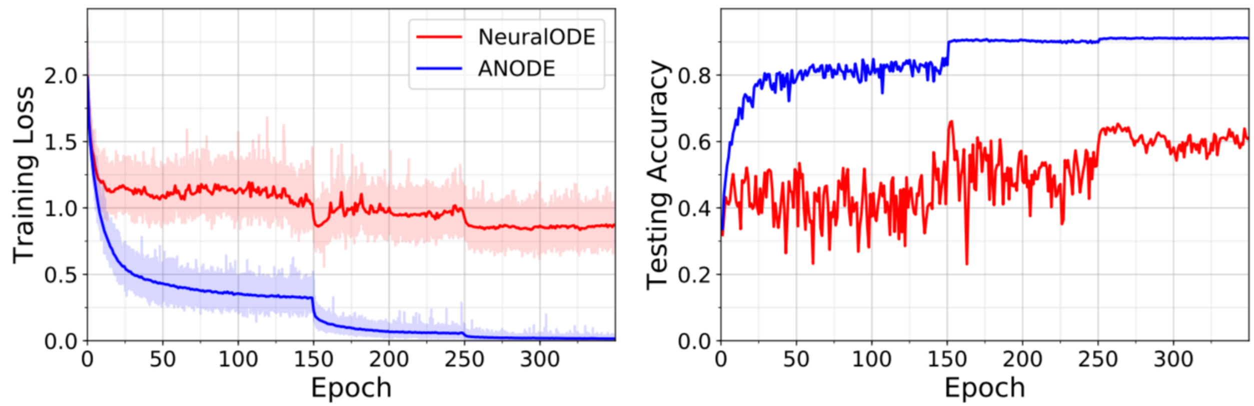
ANODE vs Neural ODE

Consider a network with L ODE layers each with Nt time steps

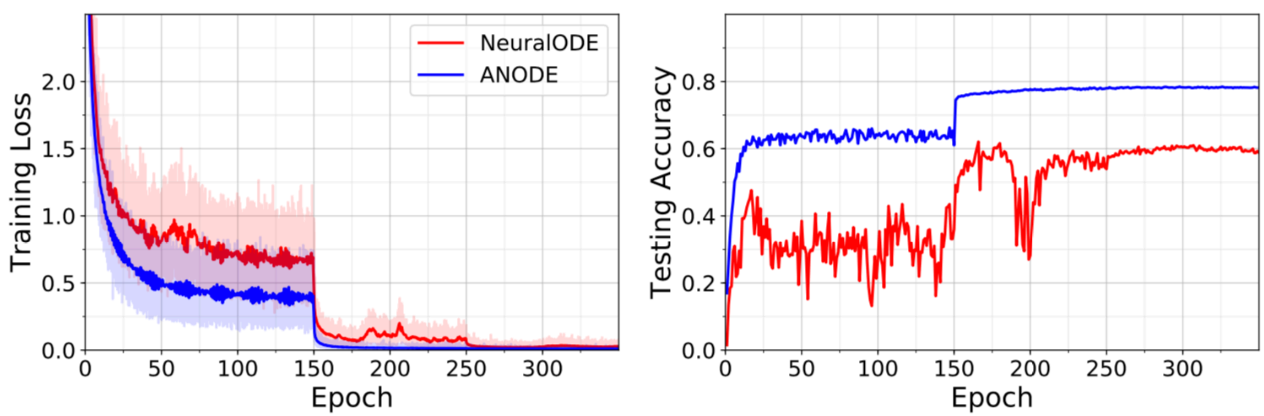
	Baseline	ANODE	Neural ODE
Memory Footprint	O(LNt)	O(L + Nt)	O(L)
FLOPS	O(LNt)	O(LNt)	O(LNt)
Stability	Stable Backprop	Stable Backprop	Unstable Backprop

A. Gholami, K. Keutzer, G. Biros. "ANODE: Unconditionally Accurate Memory-Efficient Gradients for Neural ODEs", IJCAI'19
Chen TQ, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. NeurIPS'18

Challenges with Neural ODEs



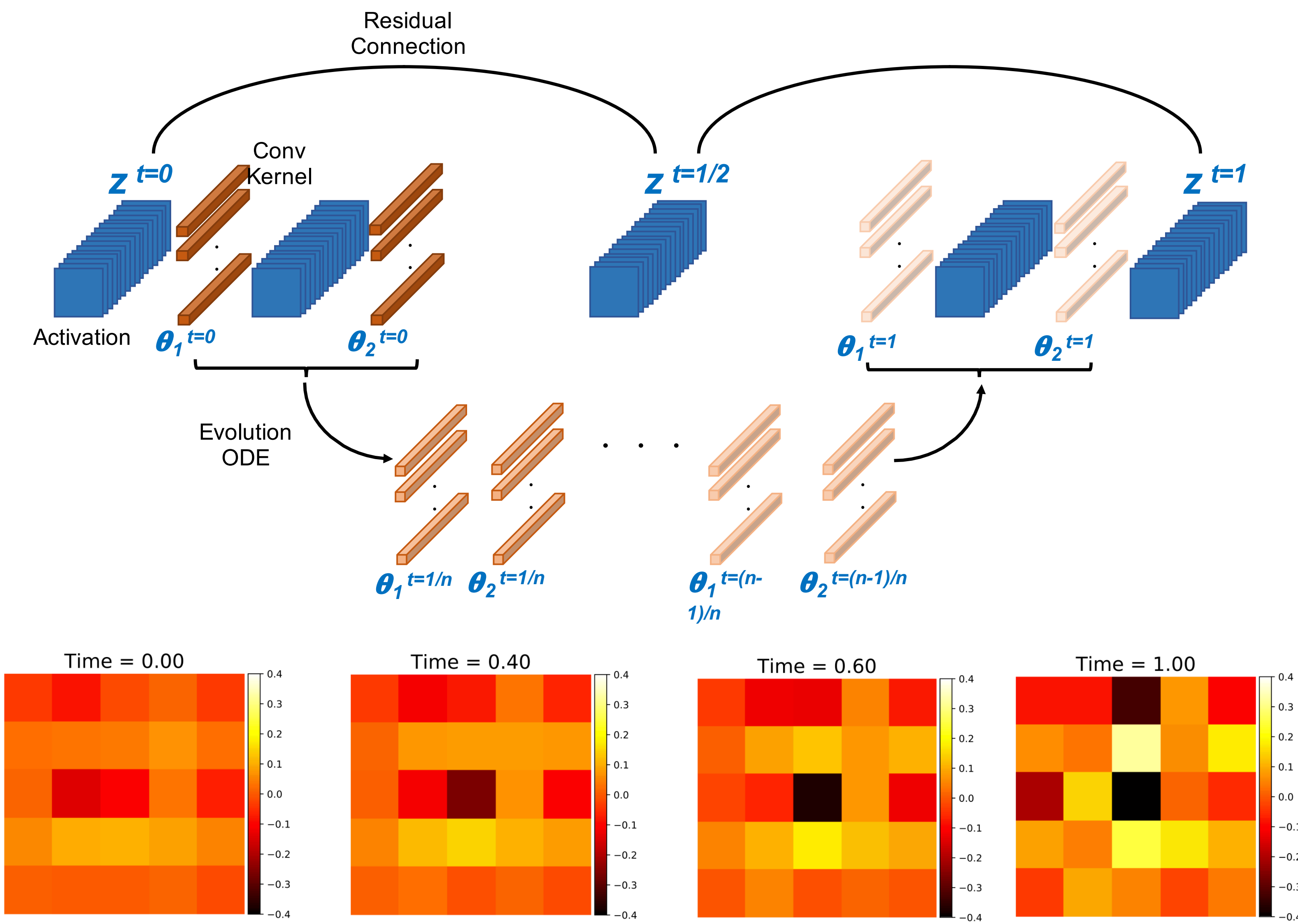
Results on Cifar-10 using SqueezeNext



Results on Cifar-100 using ResNet-18

A. Gholami, K. Keutzer, G. Biros. “ANODE: Unconditionally Accurate Memory-Efficient Gradients for Neural ODEs”, IJCAI’19

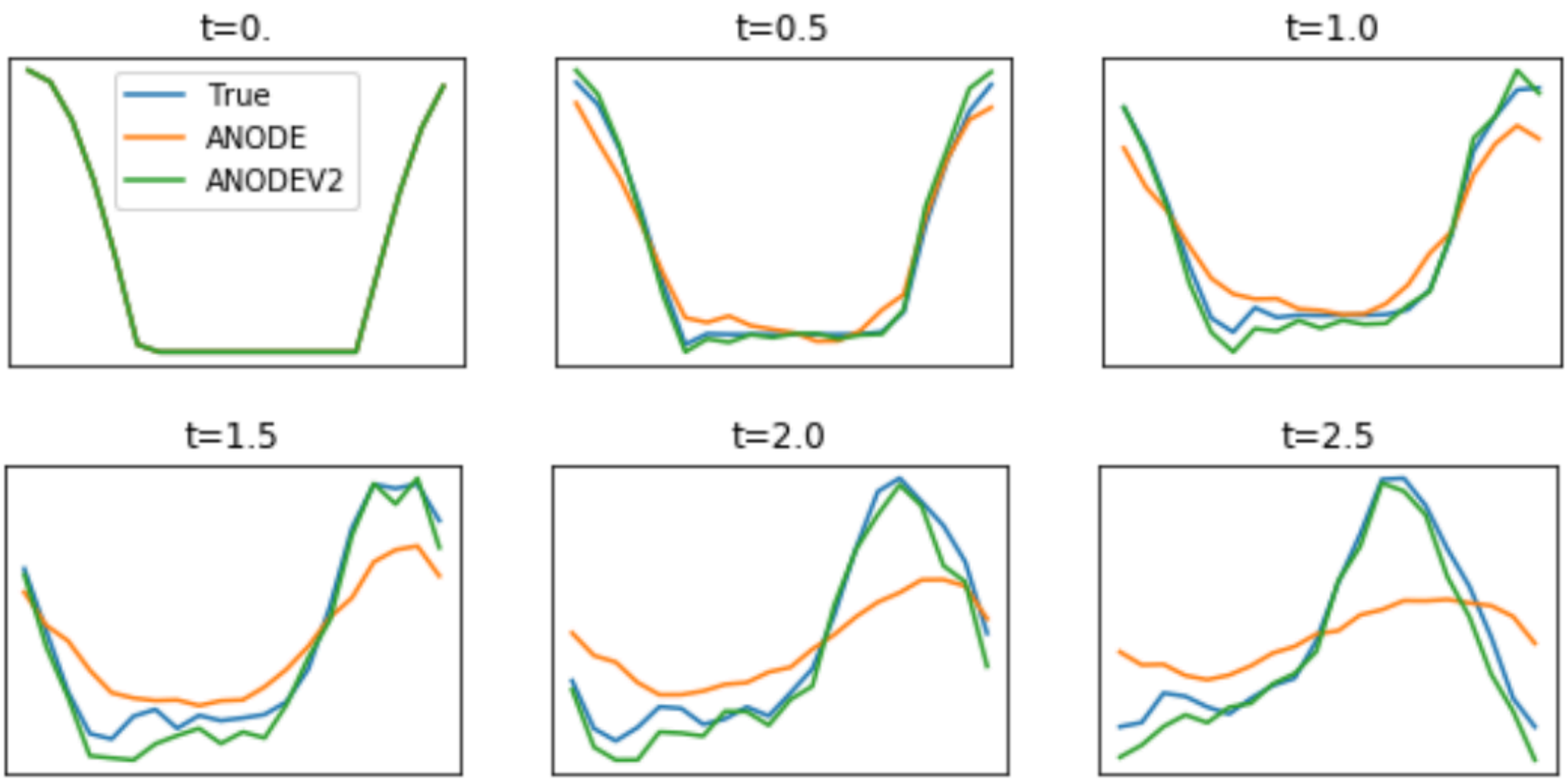
ANODEV2: A Coupled Neural ODE Framework



Zhang T, Yao Z, Gholami A, Keutzer K, Gonzalez J, Biros G, Mahoney M. "ANODEV2: A Coupled Neural ODE Evolution Framework" arXiv:1906.04596.

ANODEV2: A Coupled Neural ODE Framework

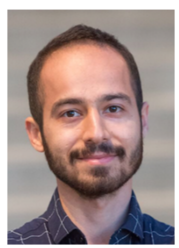
$$\frac{dz}{dt} + c(t)\frac{dz}{dx} = 0 \qquad \text{1D Wave Equation}$$



Zhang T, Yao Z, Gholami A, Keutzer K, Gonzalez J, Biros G, Mahoney M. "ANODEV2: A Coupled Neural ODE Evolution Framework" arXiv:1906.04596.

THANK YOU

Code:
github.com/amirgholami/anode

Berkeley
UNIVERSITY OF CALIFORNIA

IJCAI
2019
MACAO

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH