Collapsing Taylor Mode Automatic Differentiation

Felix Dangel*, Tim Siebert*, Marius Zeinhofer, Andrea Walther

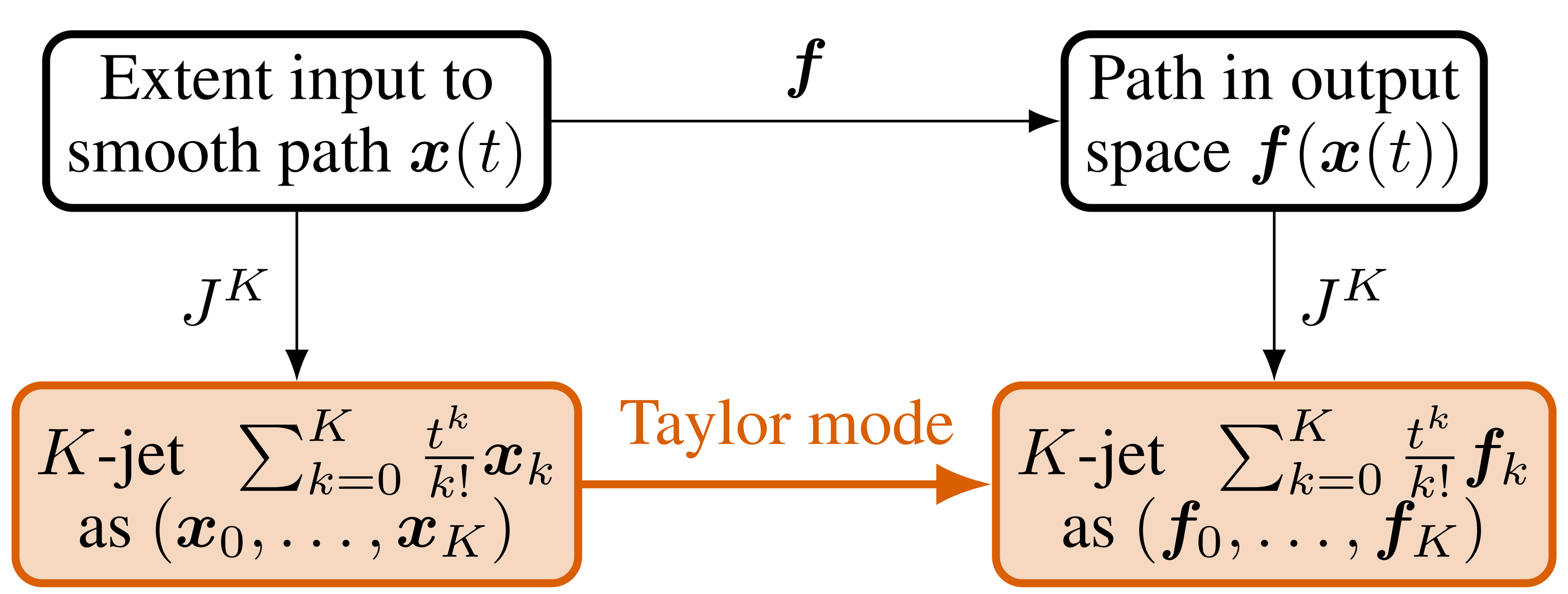
Vector Institute (Canada), Humboldt University & Zuse Institute Berlin (Germany), ETH Zürich (Switzerland)

TL;DR: We accelerate Taylor mode for practical differential operators.

How: By linearity-based graph rewrites that collapse the propagation.

Background: What is Taylor Mode Autodiff?

Taylor mode generalizes forward mode to higher-order derivatives.

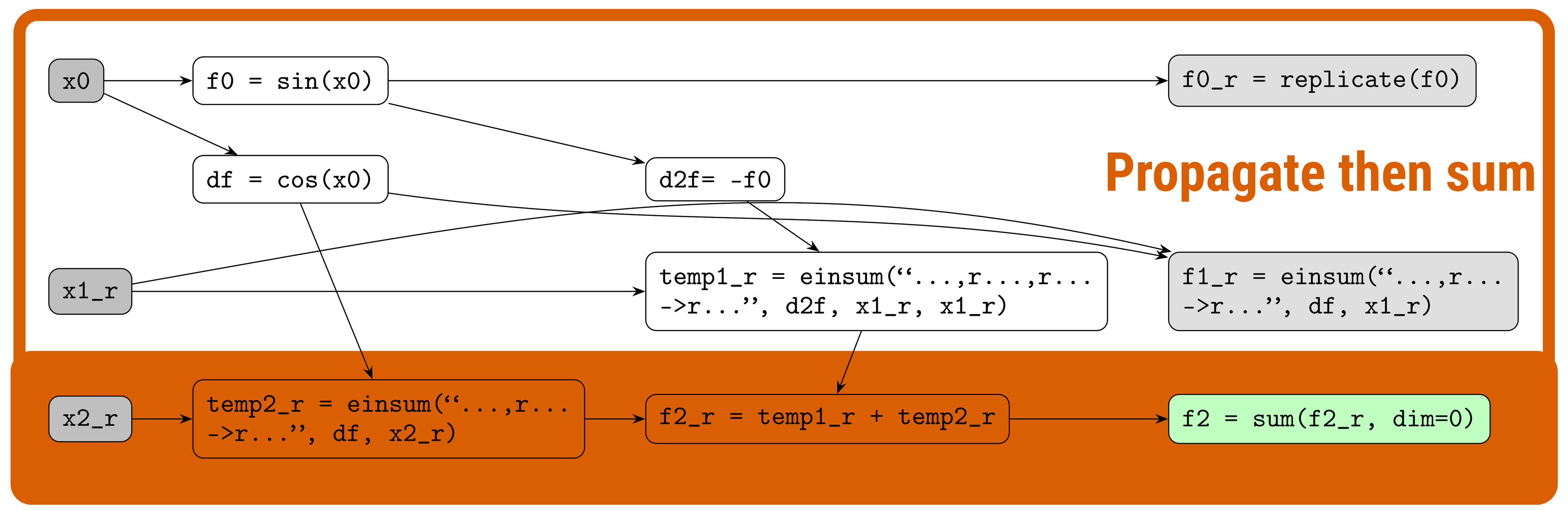


Faà di-Bruno's formula describes the propagation of Taylor coefficients (jets). Choosing different Taylor coefficients computes different derivatives.

Only JAX implements Taylor mode. We implement it for PyTorch.

```
pip install jet-for-pytorch
```

For PDE operators, we typically need to run Taylor mode along multiple directions, then sum the resulting derivatives.



Interested in similar topics? Felix is hiring for 2026! Université Concordia University · Mila

Main Idea: Use Linearity to Collapse the Propagation

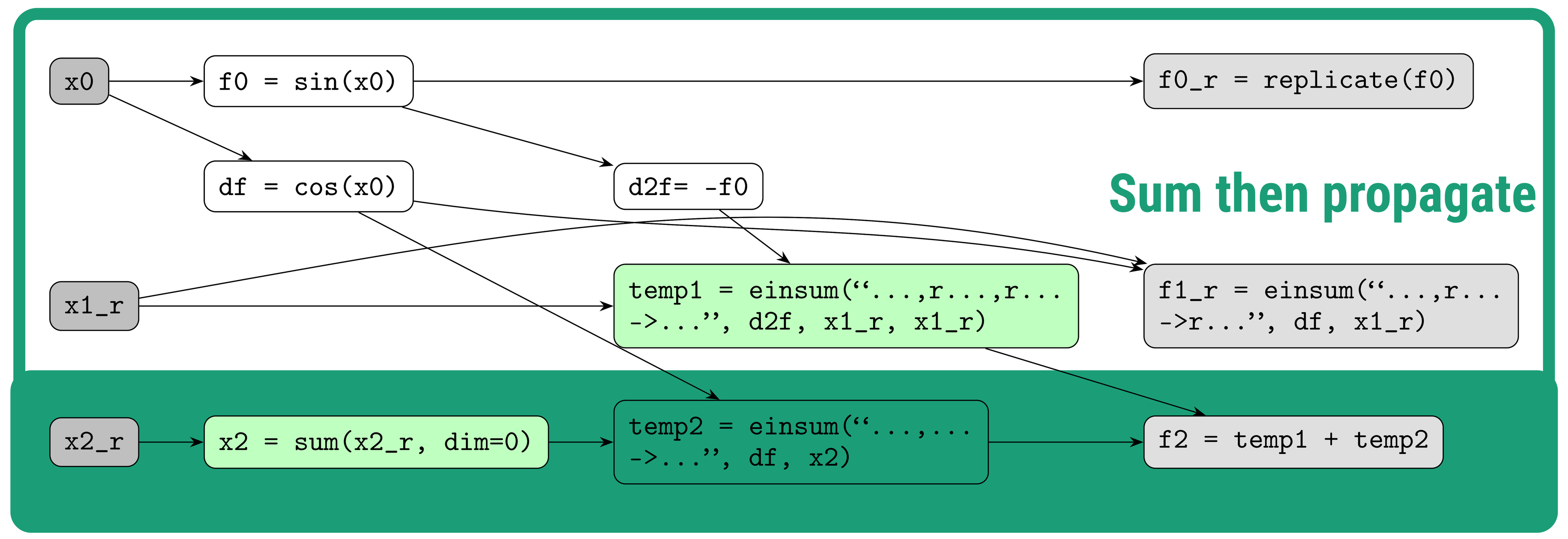
Linearity is nice because order does not matter (associativity).

Quiz: Given a matrix $\boldsymbol{W}$ and vectors $\{\boldsymbol{x}_n\}$, what is better?

$\sum_n (\boldsymbol{W}\boldsymbol{x}_n)$ **(act then sum)** or $\boldsymbol{W}\left(\sum_n \boldsymbol{x}_n\right)$ **(sum then act)**

Our insight: Faà di-Bruno's formula exhibits linearity in the sum of highest Taylor components. We can choose the better order and propagate the summed Taylor coefficient. **We call this 'collapsing' Taylor mode.**

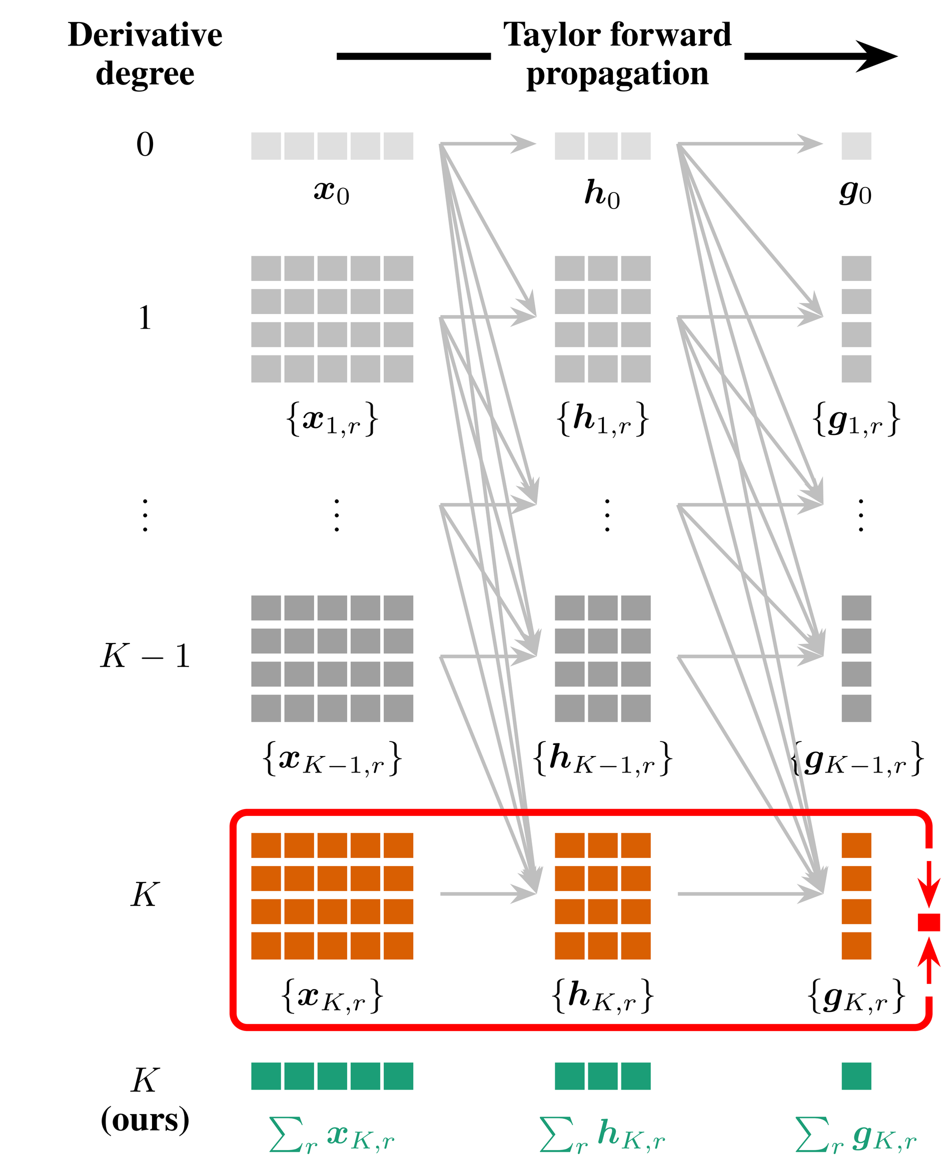
Collapsing reduces FLOPs, memory, and can be automated (e.g. in `jit`).



Scope & Performance

Collapsed Taylor mode contains and improves existing methods.

- **Laplacian** ⇔ **forward Laplacian**
- **Stochastic Taylor mode** → **collapses samples**
- **General linear PDE ops** → **collapses interpolation**



Collapsing improves vanilla Taylor mode and outperforms nesting first-order autodiff.

Mode	Per-datum/sample	Implementation	Laplacian	Weighted Laplacian	Biharmonic
Exact	Time [ms]	Nested 1st-order	0.61 (1.0x)	0.60 (1.0x)	1.2 (1.0x)
		Standard Taylor	0.56 (0.93x)	0.57 (0.94x)	0.90 (0.72x)
		Collapsed (ours)	**0.29 (0.48x)**	**0.29 (0.48x)**	**0.69 (0.55x)**
	Mem. [MiB]	Nested 1st-order	4.4 (1.0x)	4.4 (1.0x)	7.9 (1.0x)
		Standard Taylor	4.6 (1.0x)	4.6 (1.0x)	7.7 (0.98x)
		Collapsed (ours)	**2.1 (0.47x)**	**2.1 (0.47x)**	**4.8 (0.61x)**
Stochastic	Time [ms]	Nested 1st-order	24 (1.0x)	24 (1.0x)	44 (1.0x)
		Standard Taylor	23 (0.97x)	23 (0.97x)	6.6 (0.15x)
		Collapsed (ours)	**12 (0.49x)**	**12 (0.49x)**	**4.9 (0.11x)**
	Mem. [MiB]	Nested 1st-order	180 (1.0x)	180 (1.0x)	210 (1.0x)
		Standard Taylor	200 (1.2x)	200 (1.2x)	64 (0.30x)
		Collapsed (ours)	**89 (0.50x)**	**89 (0.50x)**	**38 (0.18x)**