Green-Kubo, Message-Passing Neural Networks, and Automatic Differentiation

NOMAD Laboratory
Coffee Talk
31st May 2021

Marcel F. Langer
Florian Knoop, Christian Carbogno,
Matthias Scheffler, Matthias Rupp

Technische Universität Berlin
FRITZ-HABER-INSTITUT
MAX-PLANCK-GESELLSCHAFT
NOMAD
NOVEL MATERIALS DISCOVERY

Outline

- Background
  - Green-Kubo Method
  - Message-Passing Neural Networks
  - Automatic Differentiation
- Challenges
  - Heat Flux Formulation
  - Infrastructure
- Results!

Green-Kubo Method

Thermal conductivity

Ensemble Average

$$\kappa(T, p) \propto \frac{1}{T^2 V} \lim_{t \to \infty} \int_0^t \mathrm{d}\tau \; \underbrace{\langle J(\tau) \otimes J(0) \rangle}_{\text{Heat Flux Autocorrelation Function}}{}_{T,p}$$

Temperature Pressure

Heat Flux
Autocorrelation
Function

Development:
M. Green, J. Chem. Phys. **20**, 8 (1957)
R. Kubo, M. Yokota, S. Nakajima, J. Phys. Soc. Jpn. **12**, 11 (1957)
L. Onsager, Phys. Rev. **37**, 4 405 (1931)

***Ab initio* Green-Kubo:**
C. Carbogno, R. Ramprasad, and M. Scheffler, Phys. Rev. Lett, **118**, 175901 (2017)
A. Marcolongo, P. Umari, and S. Baroni, Nat. Phys. **12**, 80 (2015)

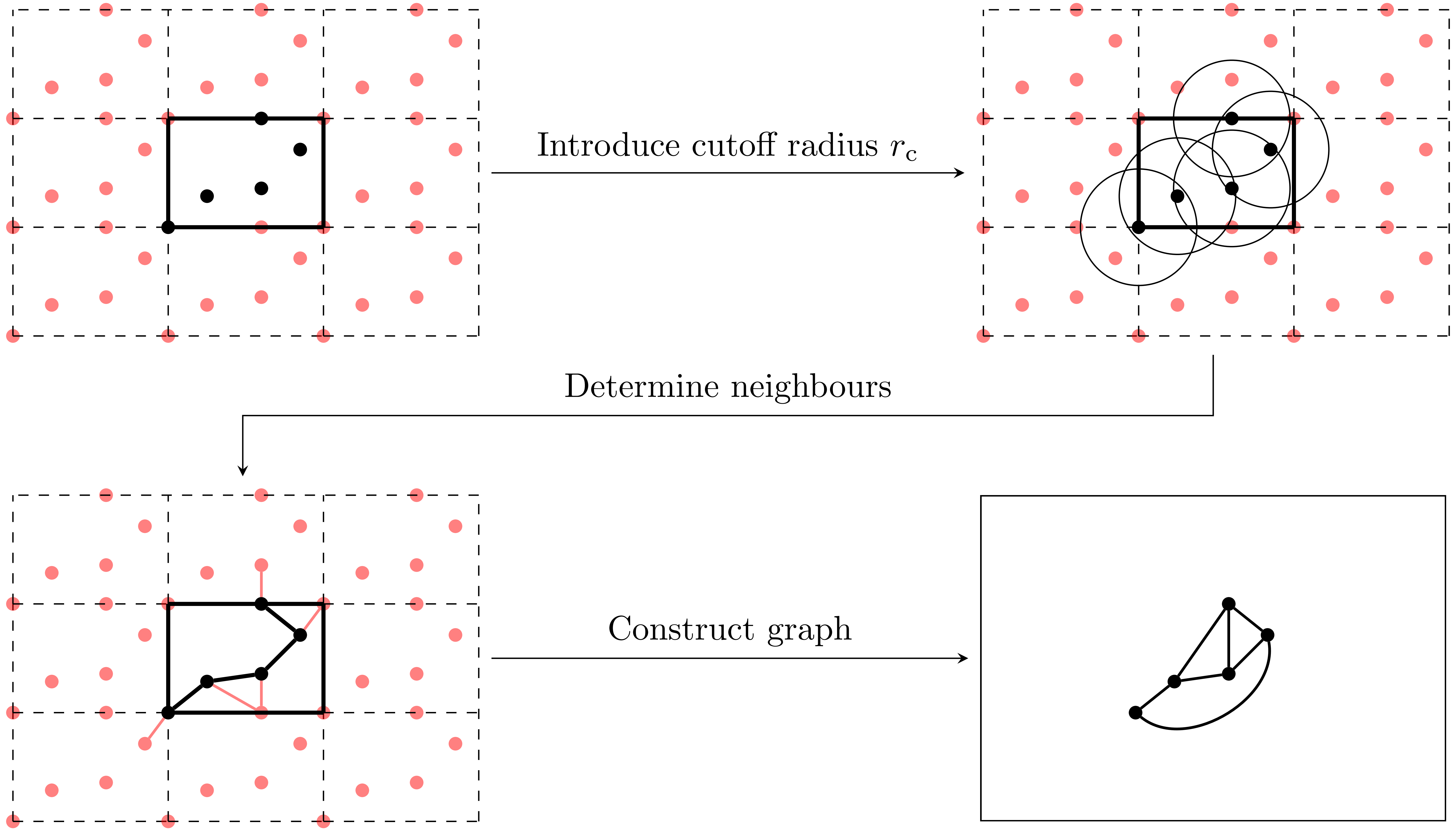
Green-Kubo Method: Practice

- For each temperature/pressure:
    - Determine unit cell
    - Thermalise
    - Run ensemble of NVE MD, compute heat flux at every (*n*-th) step
    - Post-processing: compute heat flux autocorrelation and integrate (+ noise reduction)
- Difficulty: **High computational cost** of *ab initio* calculations
    - Limited to small unit cells
    - Limited to short simulation times
    - Limited to few temperatures

Machine Learning Potentials!

Green-Kubo with Machine Learning Potentials:
P. Korotaev, I. Novoselov, A. Yanilkin, and A. Shapeev, Phys. Rev. B **100**, 144308 (2019)
C. Mangold *et al.*, J. Appl. Phys. **127**, 244901 (2020)
R. Li, E. Lee, and T. Luo, Mater. Today Phys. **12**, 100181 (2020)
at APS: **C. Verdi *et al.* B22.00003, A. Johansson *et al.* S20.00004**

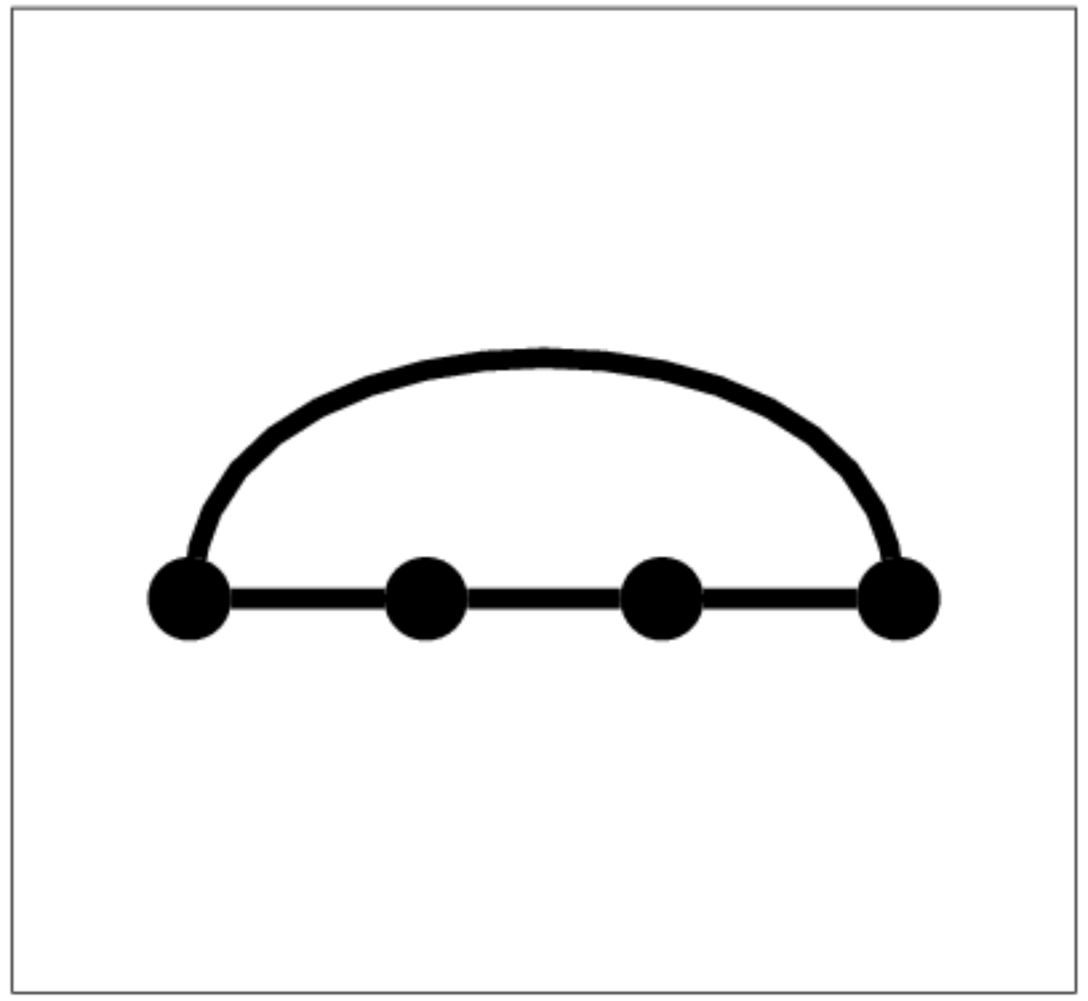
Message-Passing (Graph) **Neural Networks**



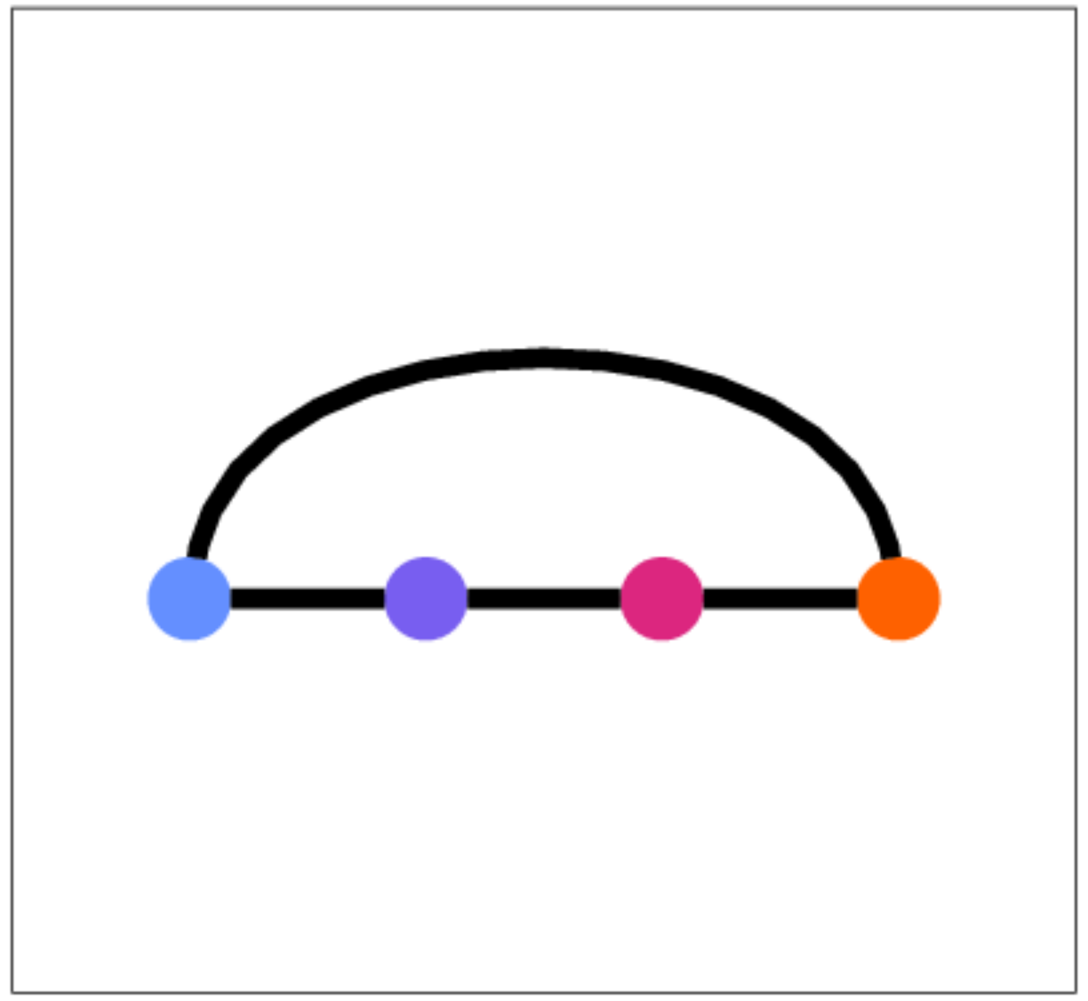
GNNs: P. W. Battaglia *et al.,* arXiv 1806.01261 (2018)
GNNs for materials: T. Xie and J.C. Grossman, Phys. Rev. Lett. **120**,145301(2018); C. Chen *et al.*, Chem. Mater. **31**, 9 (2019); C.W. Park and C. Wolverton, Phys. Rev. Mater. **4**, 063801 (2020)
MPNNs: J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, and G.E. Dahl, in Proc. of Mach. Learn. Res., Vol. 70 (2017)

Message-Passing (Graph) **Neural Networks**

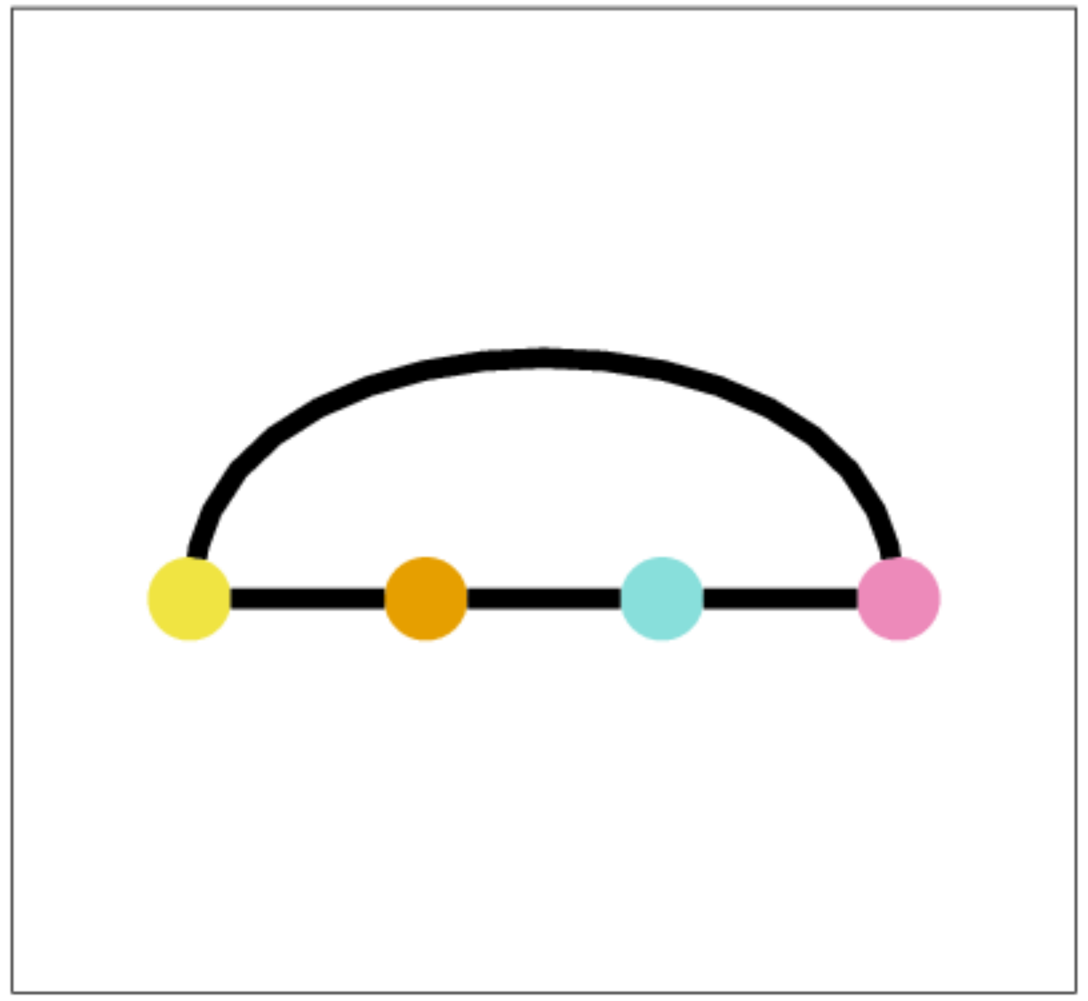
Initialisation



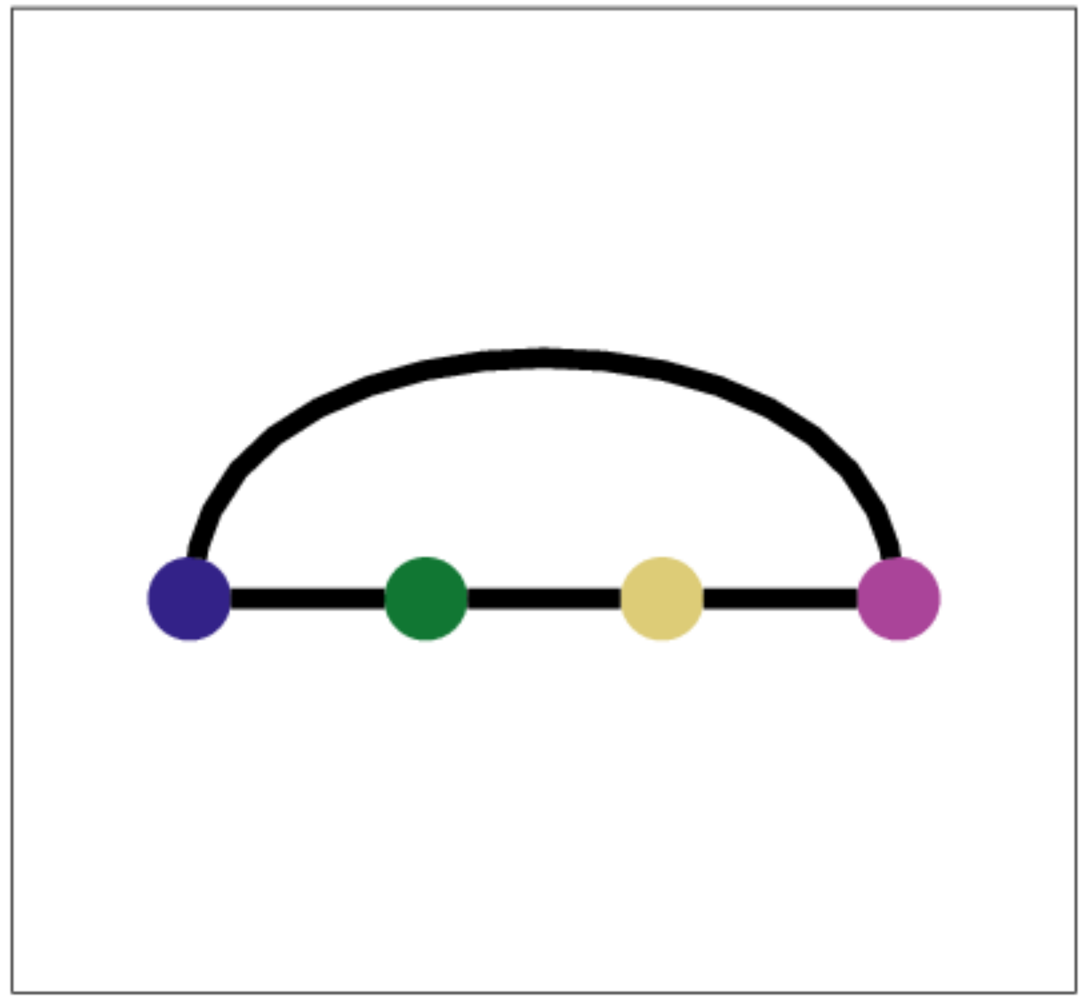
Message-passing step 1



Message-passing step 2



Readout, pooling



SchNet:
K. T. Schütt *et al.*, in *Advances in Neural Information Processing Systems 30, Los Angeles, California* (2017)
K. T. Schütt *et al.*, J. Chem. Phys. **148**, 241722 (2018)

Other MPNNs:
O. T. Unke and M. Meuwly, J. Chem. Theor. Comput. **15**, 3678 (2019)
N. Lubbers, J.S. Smith, and K. Barros, J. Chem. Phys. **148**, 241715 (2018)

Equivariant MPNNs:
J. Klicpera, J. Groß, and S. Günnemann, in *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethiopia, April 26–May 1* (2020)
S. Batzner *et al.,* arXiv 2101.03164 (2021)
K. T. Schütt, O.T. Unke, and M. Gastegger, arXiv 2102.03150 (2021)

Automatic Differentiation ("autodiff")

- Derivatives without symbolic differentiation or finite differences
- Essentially: systematically apply the chain rule

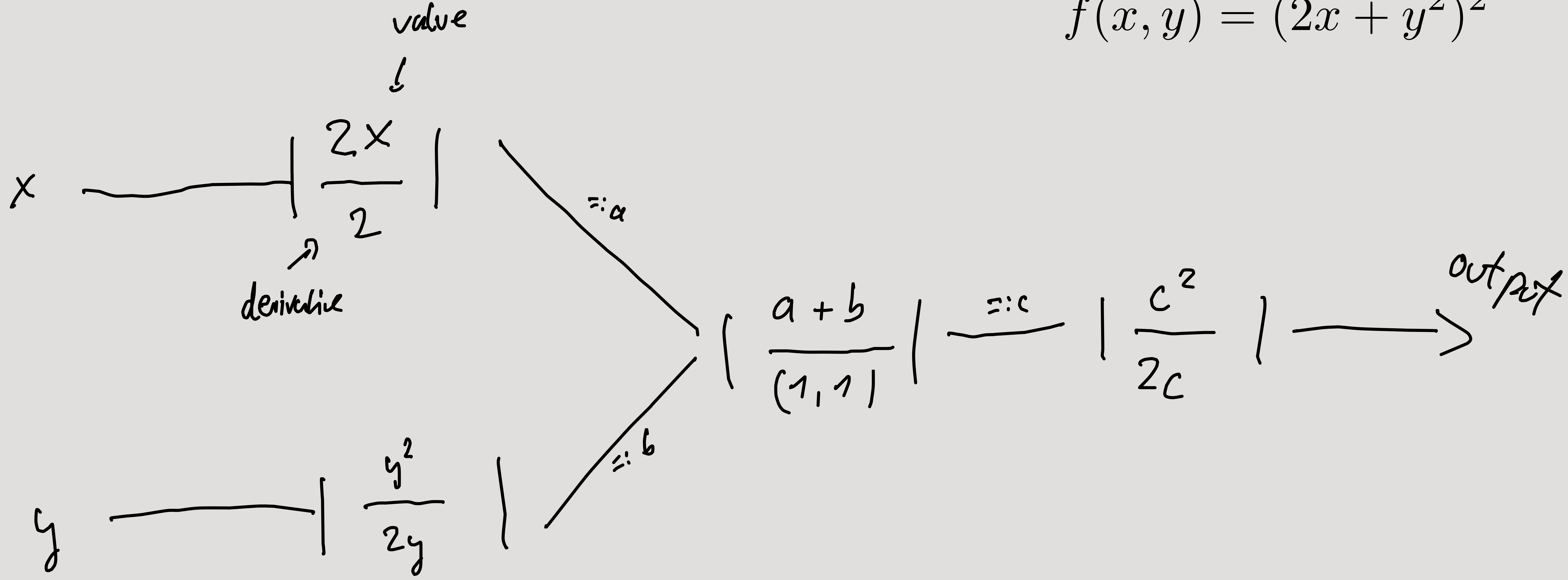
Example

$$f(x, y) = (2x + y^2)^2$$

$$\frac{\partial f}{\partial x} = 2(2x + y^2)2$$

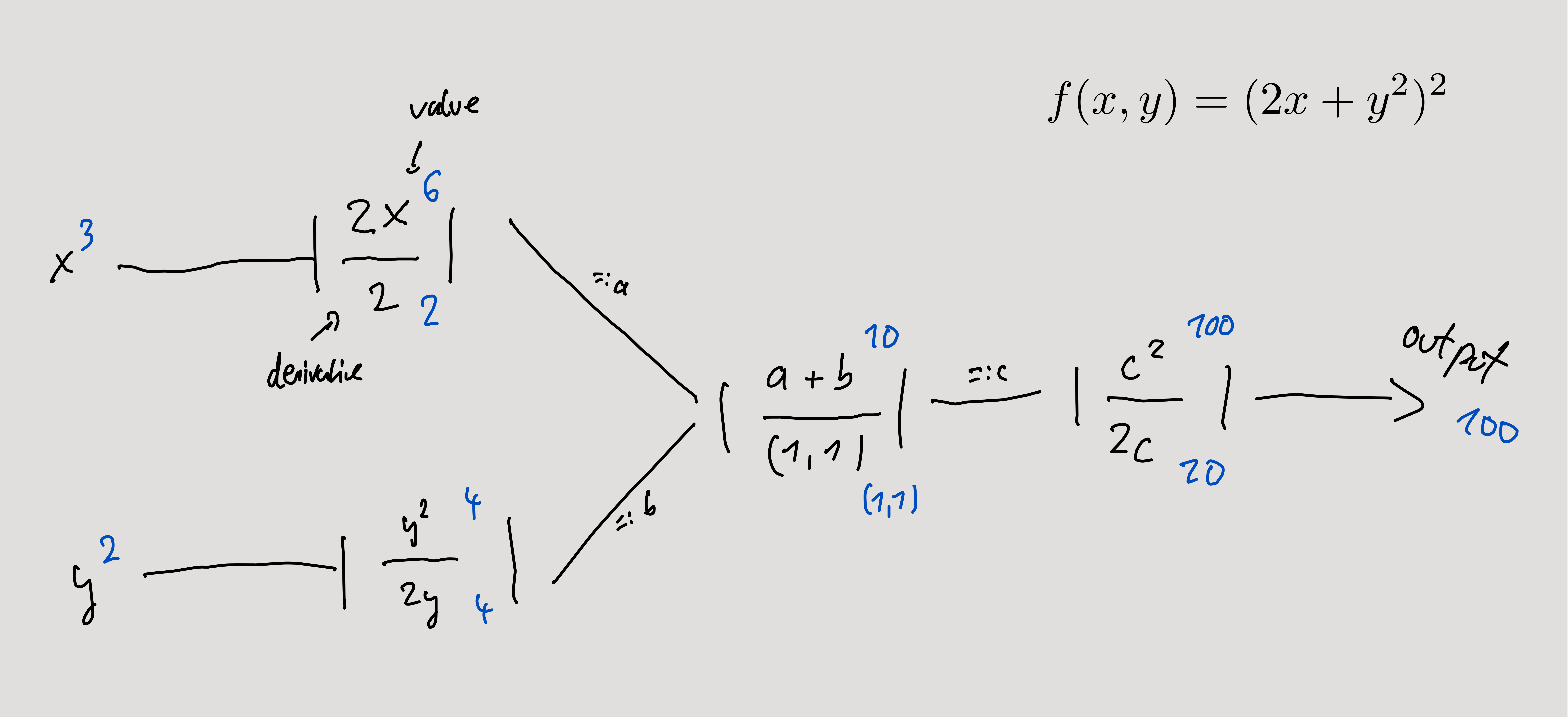
$$\frac{\partial f}{\partial y} = 2(2x + y^2)2y$$

Automatic Differentiation

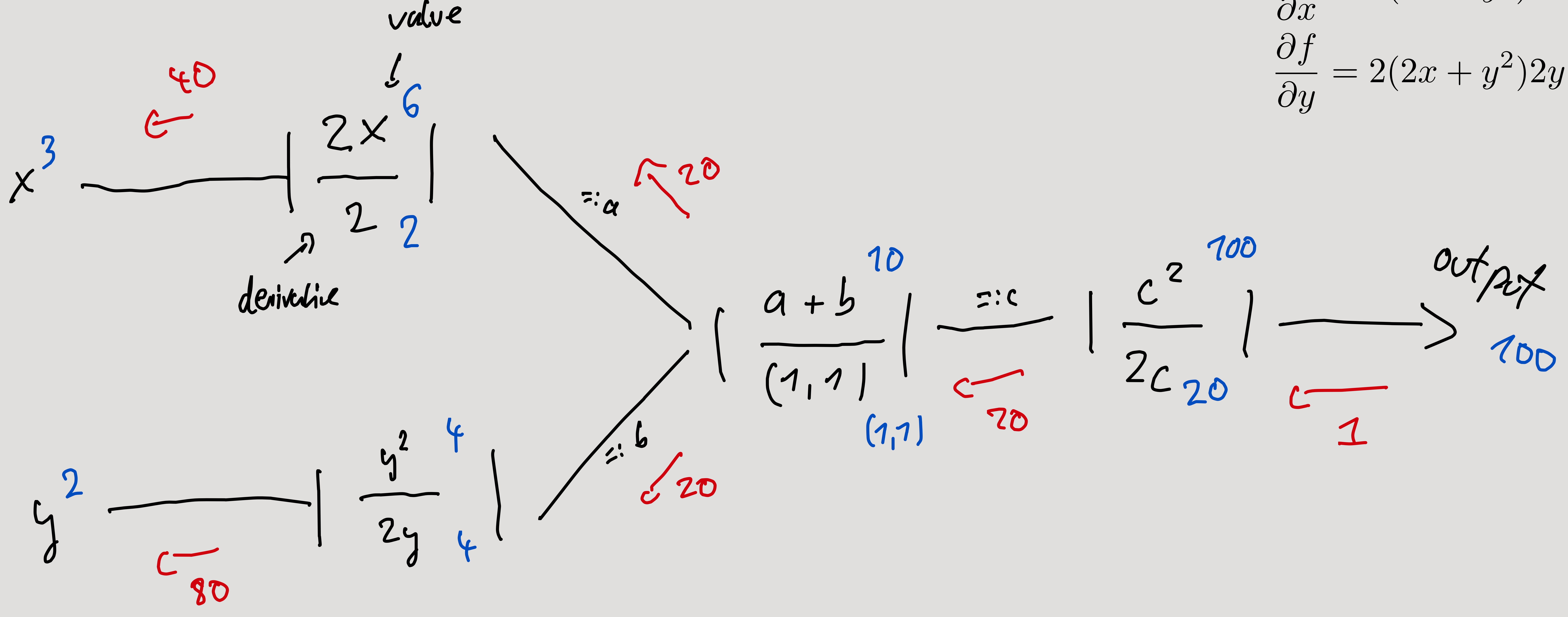


$$f(x, y) = (2x + y^2)^2$$

Automatic Differentiation: Computation Graph



Automatic Differentiation: Backpropagation/reverse mode

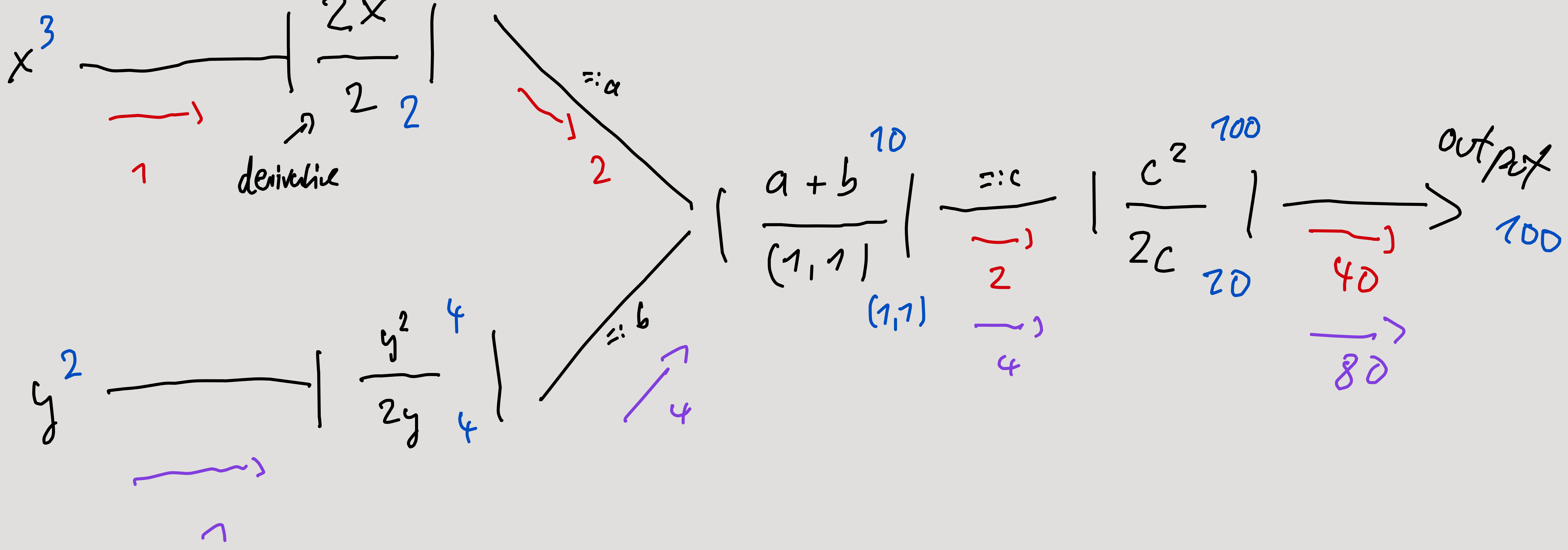


$$\frac{\partial f}{\partial x} = 2(2x + y^2)2$$

$$\frac{\partial f}{\partial y} = 2(2x + y^2)2y$$

Automatic Differentiation: forward mode

$$\frac{\partial f}{\partial x} = 2(2x + y^2)2$$
$$\frac{\partial f}{\partial y} = 2(2x + y^2)2y$$



Automatic Differentiation (“autodiff”)

Takeaways

- Backward mode: derivative of **one** output wrt **all** inputs in **one** traversal
- Forward mode: derivatives of **all** outputs wrt **one** input in **one** traversal

➡ Efficient for different computations

➡ **Cheap** one column/one row Jacobians

- Obtain derivatives of any node in the graph with respect to earlier steps

Challenge 1: Heat Flux Formulation

$$\kappa(T, p) \propto \frac{1}{T^2 V} \lim_{t \to \infty} \int_0^t \mathrm{d}\tau \, \langle J(\tau) \otimes J(0) \rangle_{T,p}$$

Heat Flux

$$E(t) = \int d^3\vec{r}\, e(r, t)$$

$$\vec{J}(t) = \frac{\mathrm{d}}{\mathrm{d}t} \int \mathrm{d}^3\vec{r}\; \vec{r} e(\vec{r}, t)$$

Marcolongo, Umari, Baroni, *Nat. Phys.* **12** 80 (2015)

Heat Flux: Hardy Formula

$$\vec{J}(t) = \frac{\mathrm{d}}{\mathrm{d}t} \int \mathrm{d}^3\vec{r}\; \vec{r} e(\vec{r}, t)$$

$$e(\vec{r}) = \sum_i \delta(\vec{r} - \vec{r}_i)\,(U_i + T_i) \qquad \Rightarrow \vec{J} = \sum_i \vec{r}_i \left(\dot{T}_i + \dot{U}_i\right) + \sum_i \dot{\vec{r}}\,(T_i + U_i)$$

$$U_i = U_i(\{r_i\}_{i=1,\ldots,n}, \{b_1, b_2, b_3\}) \qquad \Rightarrow \vec{J} = \sum_{ji} \vec{r}_{ij} \left(\frac{\partial U_i}{\partial \vec{r}_j} \cdot \vec{v}_j\right)$$

Marcolongo, Umari, Baroni, *Nat. Phys.* **12** 80 (2015)

Fan et al., *Phys. Rev. B* **92** 094301 (2015)

Heat Flux: Virial Formula

$$\vec{J}(t) = \frac{\mathrm{d}}{\mathrm{d}t} \int \mathrm{d}^3\vec{r}\; \vec{r} e(\vec{r}, t)$$

$$e(\vec{r}) = \sum_i \delta(\vec{r} - \vec{r}_i)\,(U_i + T_i) \qquad \Rightarrow \vec{J} = \underbrace{\sum_i \vec{r}_i \left(\dot{T}_i + \dot{U}_i\right)} + \sum_i \dot{\vec{r}}\,(T_i + U_i)$$

$$U_i = U_i(\{\vec{r}_{ij} | j \neq i\}) \qquad \Rightarrow \vec{J} = \sum_{ij} \vec{r}_{ji} \left(\frac{\partial U}{\partial \vec{r}_{ij}} \cdot \vec{v}_j \right)$$

C. Carbogno, R. Ramprasad, and M. Scheffler,
Phys. Rev. Lett, **118**, 175901 (2017)

Heat Flux: Autodiff

$$\vec{J} = \sum_{ij} \vec{r}_{ji} \left(\frac{\partial U}{\partial \vec{r}_{ij}} \cdot \vec{v}_j \right)$$

Virial formula

Gradient of **only** *U*
☞ *0.08s/step for 2500 atoms*

$$\vec{J} = \sum_{ji} \vec{r}_{ij} \left(\frac{\partial U_i}{\partial \vec{r}_j} \cdot \vec{v}_j \right)$$

Hardy formula

Gradient of **all** U_i
☞ *60s/step for 2500 atoms*

Are they equivalent…?

Challenge 2: Infrastructure

… a *small* problem

```
$ la -h */md/*.son
-rw-r--r-- 1 mlang mfh 1.6T May 16 11:53 00/md/trajectory.son
-rw-r--r-- 1 mlang mfh 1.6T May 16 12:14 01/md/trajectory.son
-rw-r--r-- 1 mlang mfh 1.6T May 16 11:57 02/md/trajectory.son
-rw-r--r-- 1 mlang mfh 1.6T May 16 12:27 03/md/trajectory.son
-rw-r--r-- 1 mlang mfh 1.6T May 16 14:17 04/md/trajectory.son
-rw-r--r-- 1 mlang mfh 1.6T May 16 12:21 05/md/trajectory.son
-rw-r--r-- 1 mlang mfh 590G May 12 13:36 06/md/trajectory.son
-rw-r--r-- 1 mlang mfh 1.6T May 16 14:12 07/md/trajectory.son
-rw-r--r-- 1 mlang mfh 1.6T May 16 12:58 08/md/trajectory.son
-rw-r--r-- 1 mlang mfh 1.6T May 16 14:17 09/md/trajectory.son
-rw-r--r-- 1 mlang mfh 1.6T May 16 14:43 10/md/trajectory.son
-rw-r--r-- 1 mlang mfh 131G May  4 06:56 therm/md/trajectory.son
```

This does *not* fit into memory!

Infrastructure

`stepson` | towards `vibes` 2.0

✓ Instant MD restarts

➡ Run MD with large systems

✓ Stream-based and batched parsing of trajectories

✓ Stream-based and batched GK post-processing

➡ Work with trajectories of any length and number of atoms

✓ `dask` backend for post-processing

➡ Potential for parallel/multinode/exascale processing

✓ ✨ Nice console output ✨

➡ Fun to work with 😉

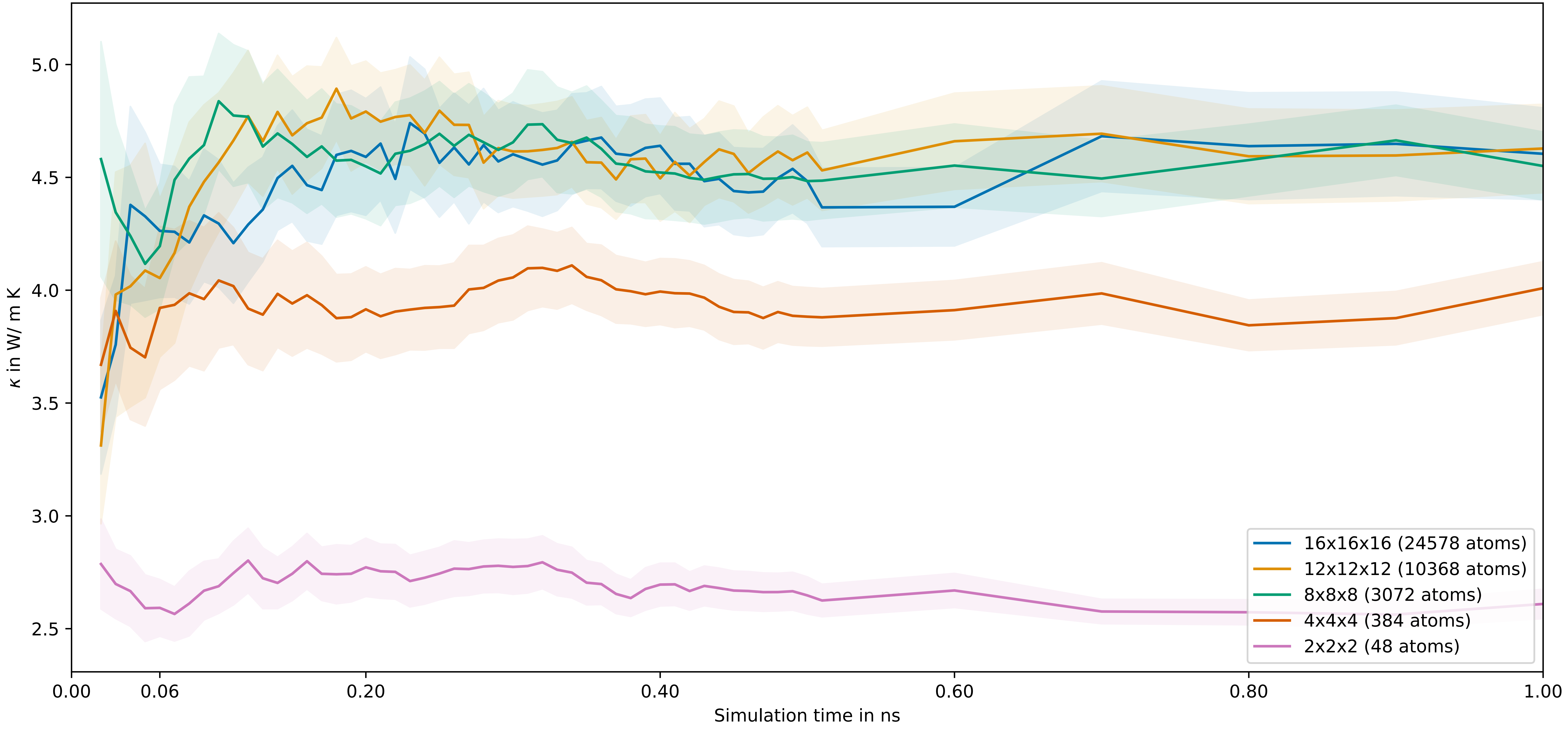
gknet /Users/marcel/base/desk/phd/projects/gknet/playground/stepson

```
See you, space cowboy... ✨

marcel@vpn-64-154:~
➤ z stepson
direnv: loading ~/base/desk/phd/projects/gknet/.envrc
direnv: export +VIRTUAL_ENV ~PATH ~XPC_SERVICE_NAME
marcel@vpn-64-154:~/b/d/p/p/g/p/stepson
➤ gknet out md zro_schnet.son
```

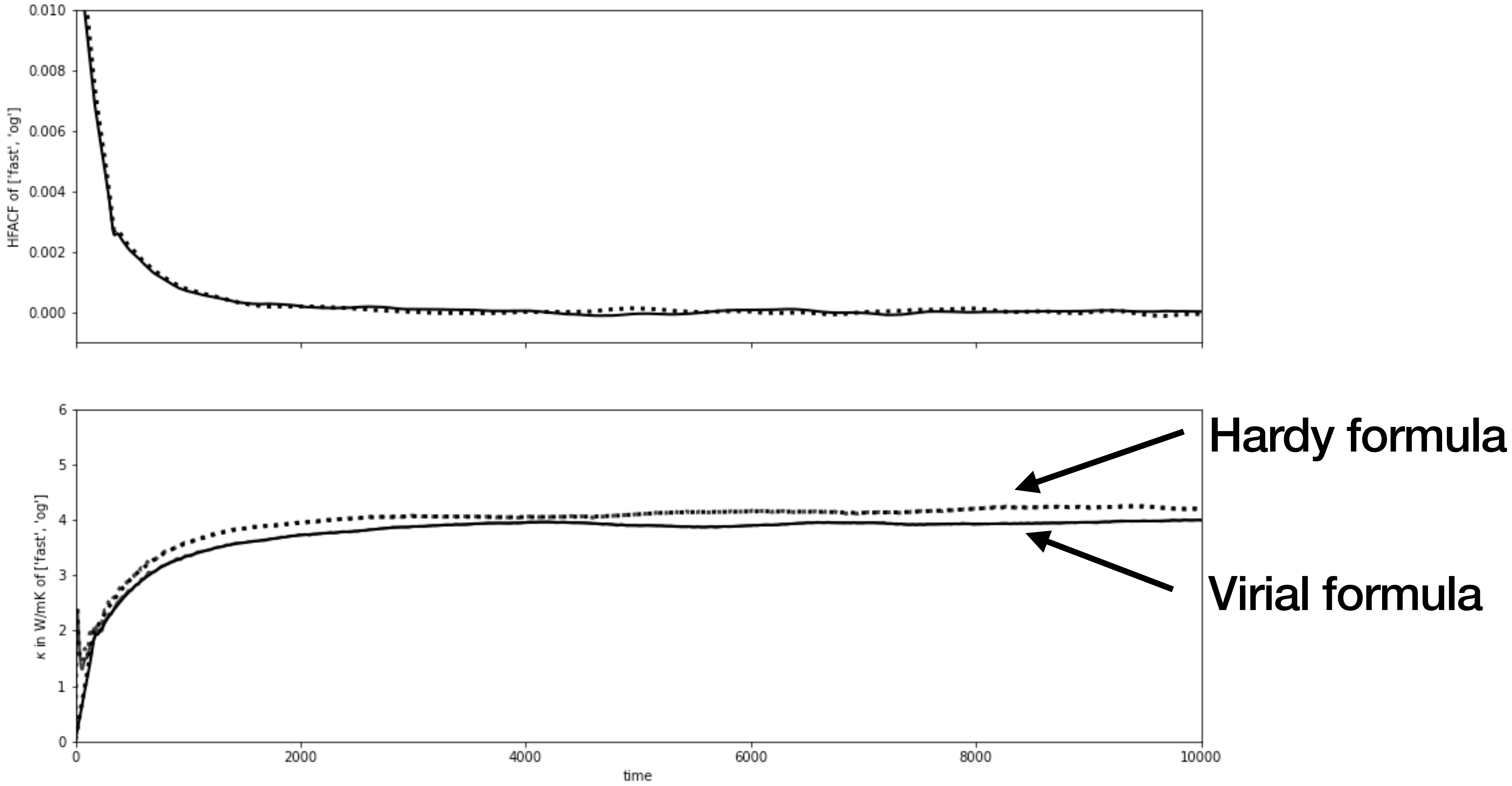
Results!

ZrO_2 @ 300K with SchNet (11 runs each) (virials heat flux)



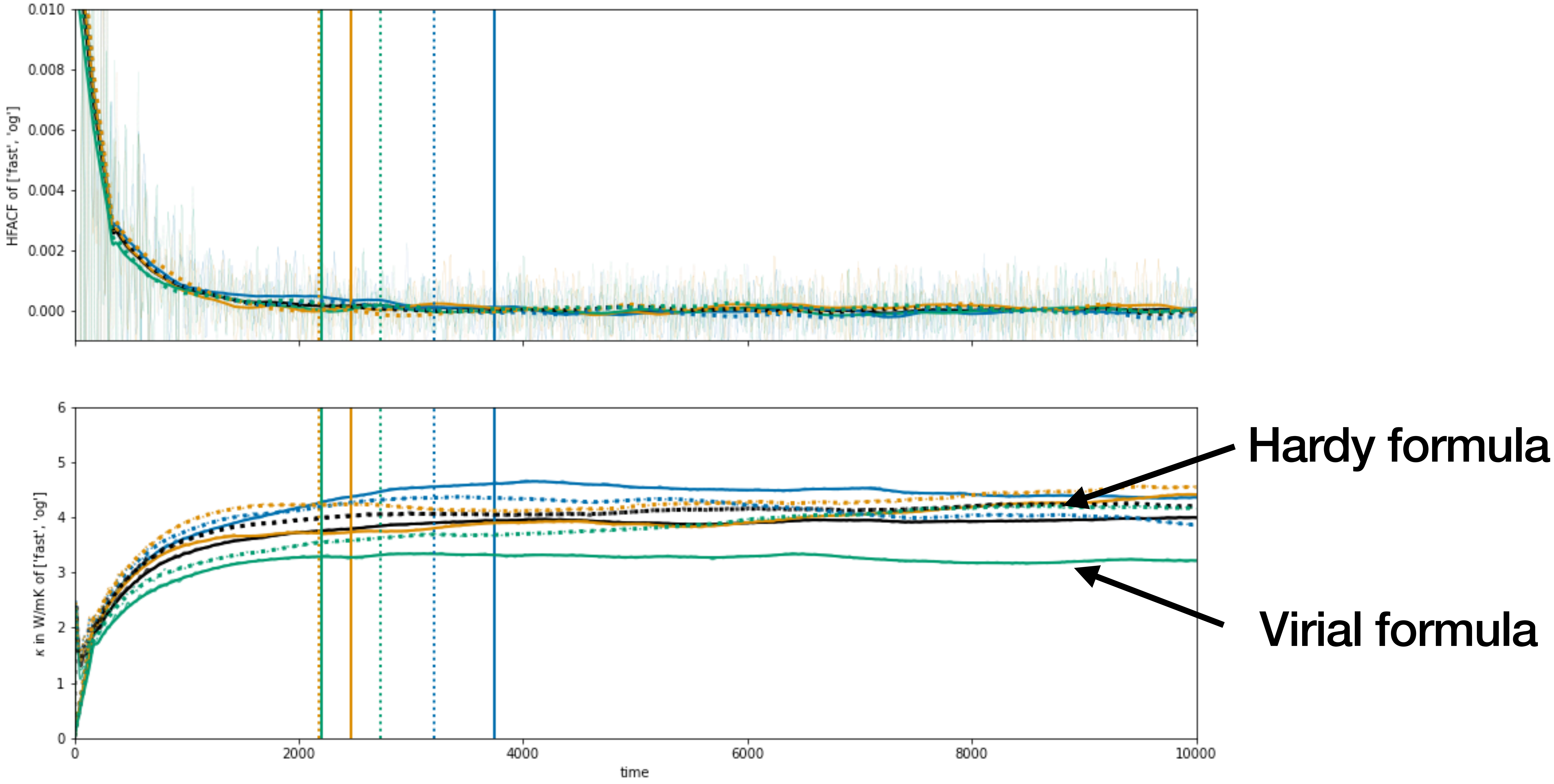
Results!

ZrO_2 @ 300K with SchNet (3 runs each, 384 atoms)



Results!

ZrO_2 @ 300K with SchNet (3 runs each, 384 atoms) with <u>components</u> of *κ*



Next

- Heat flux formulation: further testing for equivalence
- Converged runs at higher temperatures
- Lattice expansion with i-pi

Questions? Updates?

langer@fhi.mpg.de

@marceldotsci

Thanks to:

Klaus-Robert Müller

Davide Donadio

Michael Gastegger

Kristof Schütt

Yair Litman

European Research Council

Established by the European Commission

TEC1p: Big-Data Analytics for the Thermal and Electrical Conductivity of Materials from First Principles

Co-authors:

Florian Knoop, Christian Carbogno, Matthias Scheffler, Matthias Rupp