Reducing the computational cost of solving ODE systems via machine learning

Tommaso Grassi (LMU)

F.Nauman, J.P.Ramsey, S.Bovino, G.Picogna, B.Ercolano

Reducing the computational cost of solving ODE systems via machine learning

Astronomy & Astrophysics manuscript no. main
April 27, 2021

Reducing the complexity of chemical networks via interpretable autoencoders

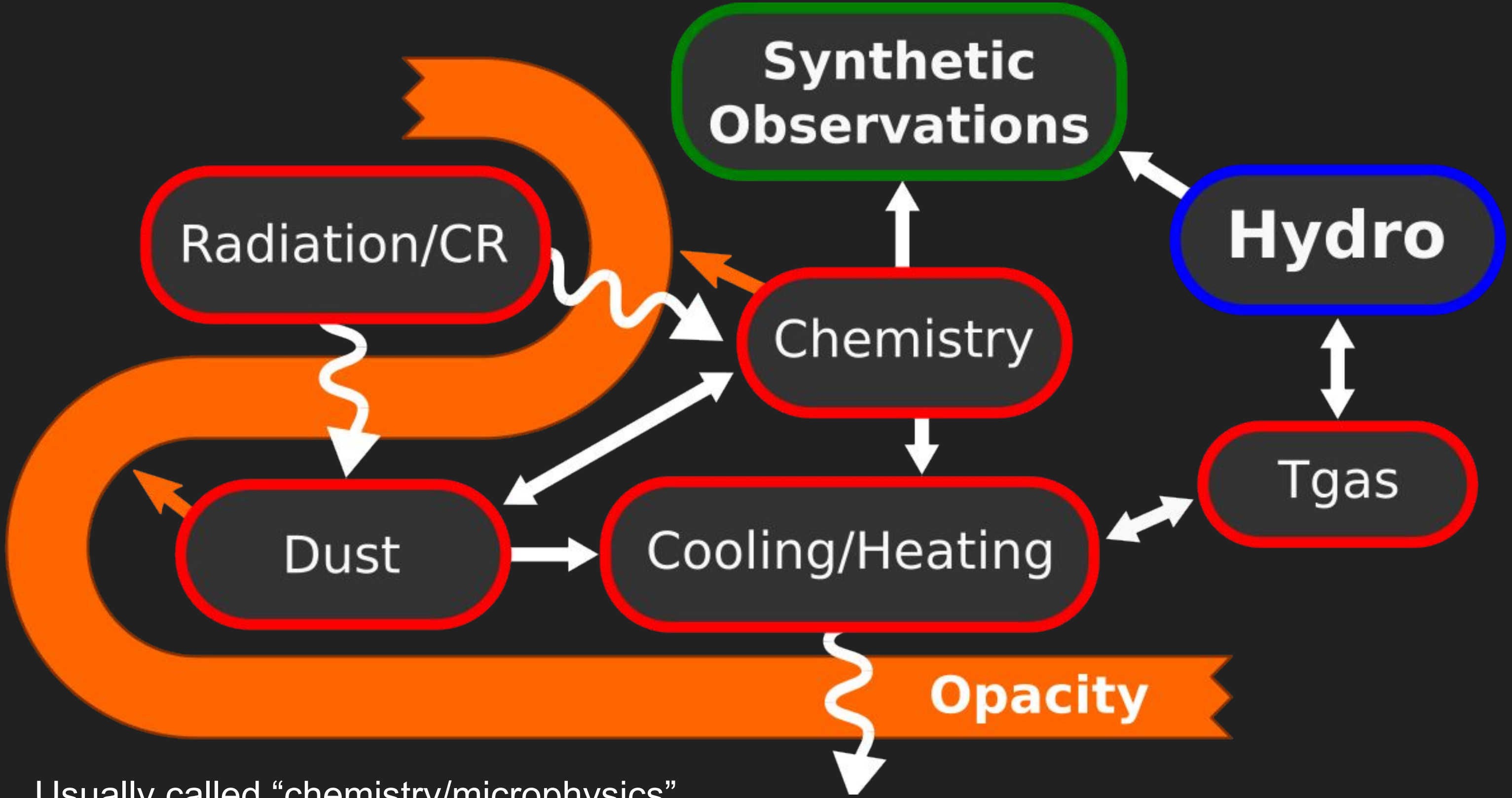
T. Grassi[1, 2,★], F. Nauman[3], J. P. Ramsey[4], S. Bovino[5], G. Picogna[1, 2], and B. Ercolano[1, 2]

arxiv: 2104.09516

Outline

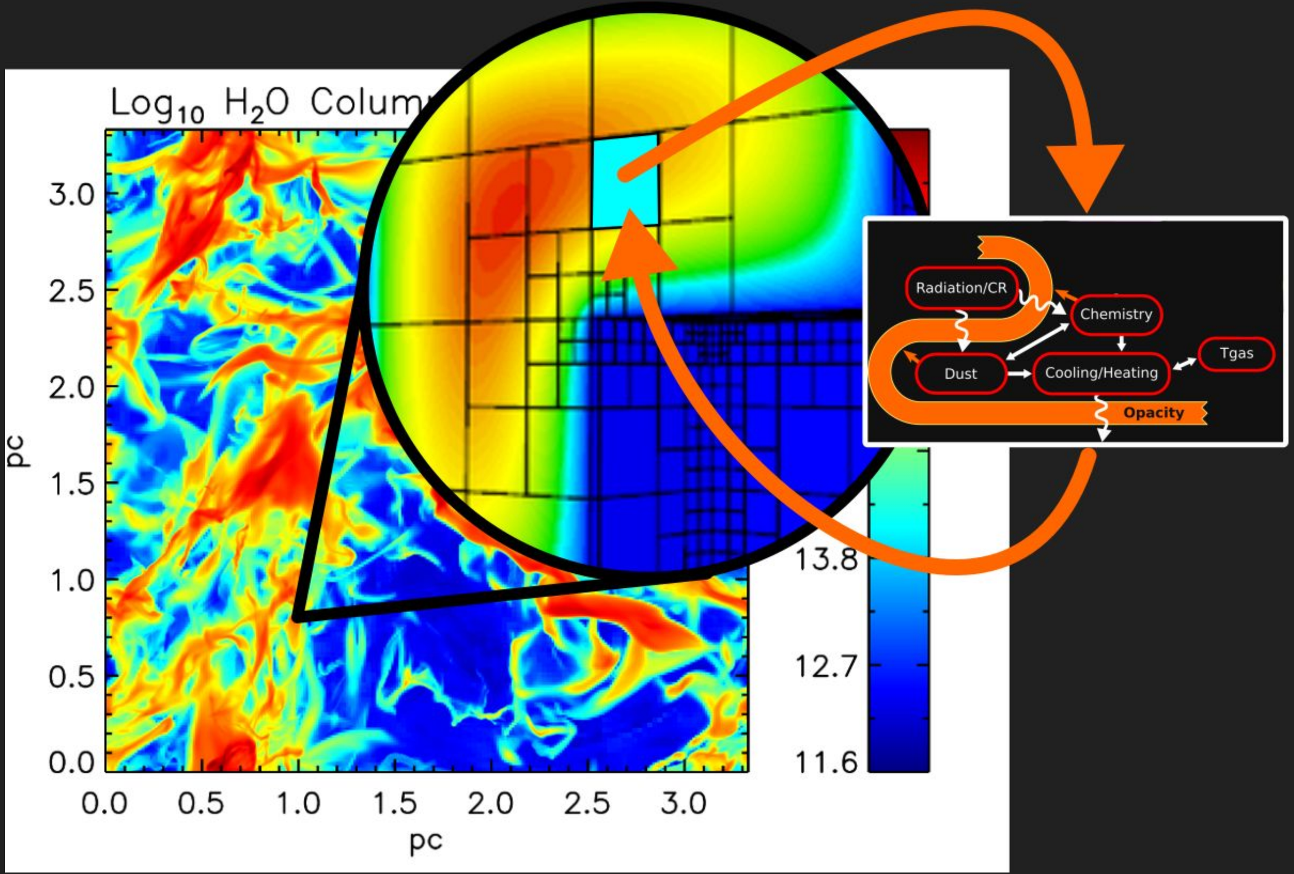
1. Astrophysical Context
2. System of Ordinary Differential Equations (ODEs)
3. ODE systems in (astro)chemistry
4. Deep Neural Networks (DNNs)
5. DNN Autoencoders
6. Our Method: ODEs + Autoencoders

Context /1



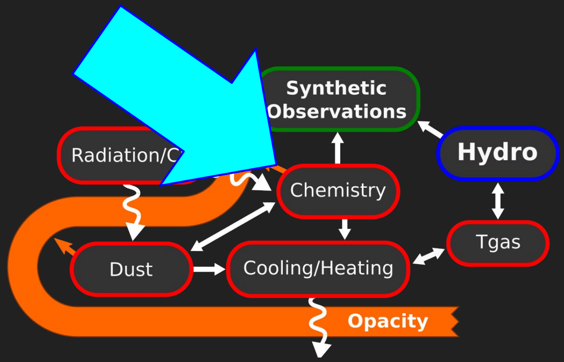
Usually called “chemistry/microphysics”

Context /2



Computational Overhead

Chemistry



$$\boxed{\mathrm{C} + \mathrm{O}} \leftrightharpoons \boxed{\mathrm{CO}}$$

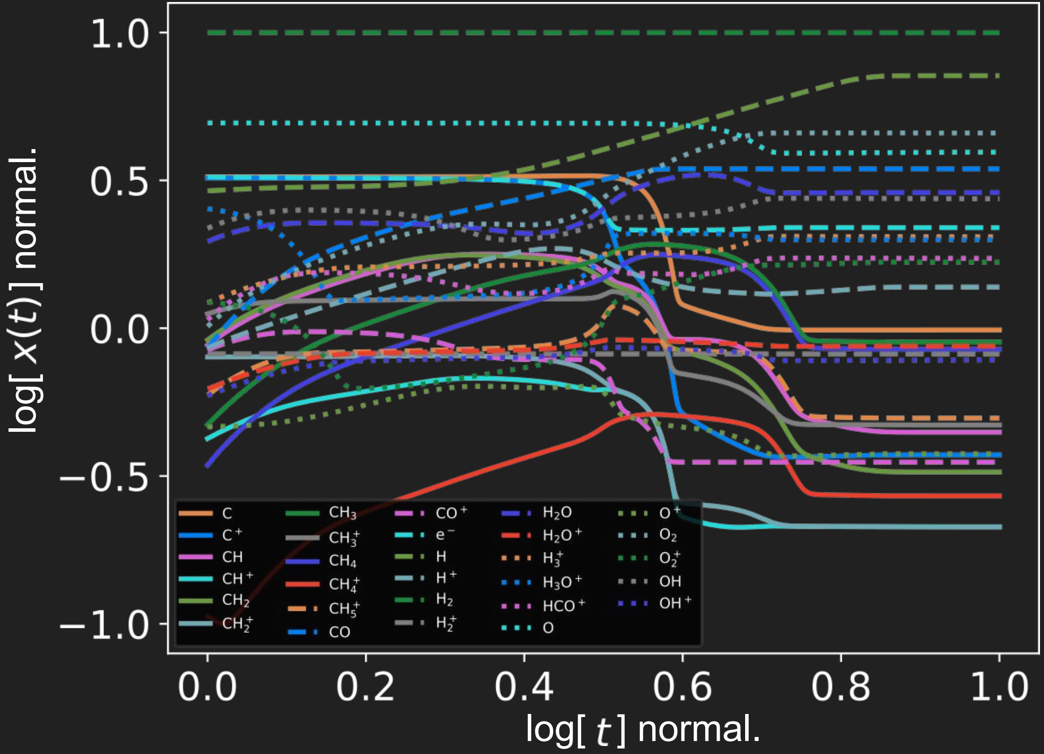
$$\frac{\mathrm{d}x_{\mathrm{C}}}{\mathrm{d}t} = \frac{\mathrm{d}x_{\mathrm{O}}}{\mathrm{d}t} = -\boxed{k_1\, x_{\mathrm{C}}\, x_{\mathrm{O}}} + \boxed{k_2\, x_{\mathrm{CO}}}$$

$$\frac{\mathrm{d}x_{\mathrm{CO}}}{\mathrm{d}t} = +\boxed{k_1\, x_{\mathrm{C}}\, x_{\mathrm{O}}} - \boxed{k_2\, x_{\mathrm{CO}}}$$

$$\dot{\bar{x}} = f(\bar{x}; k)$$

Solving ODEs

Advance x(t) in time with BDF solving $\dot{x} = f(\bar{x}; k)$



Typical networks

Species → Reactions

Primordial: 10 → 30

MC (hydro): 40 → 300

MC (0D): 450 → 4500+

Disk: 450+ → 5000+

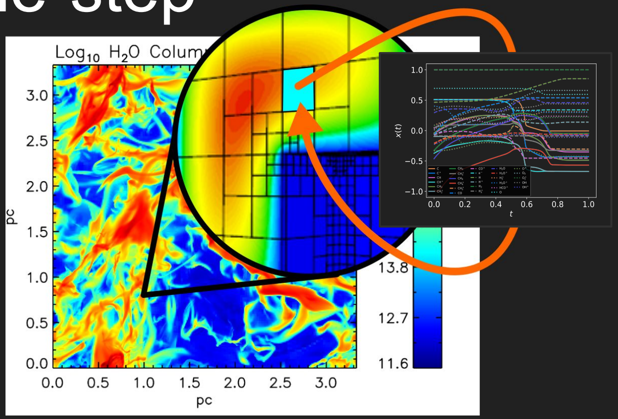
This Example: 29 → 224

PROBLEM

Integrating for a (hydro) time-step

$$\dot{x} = f(\bar{x}; k)$$

for each cell



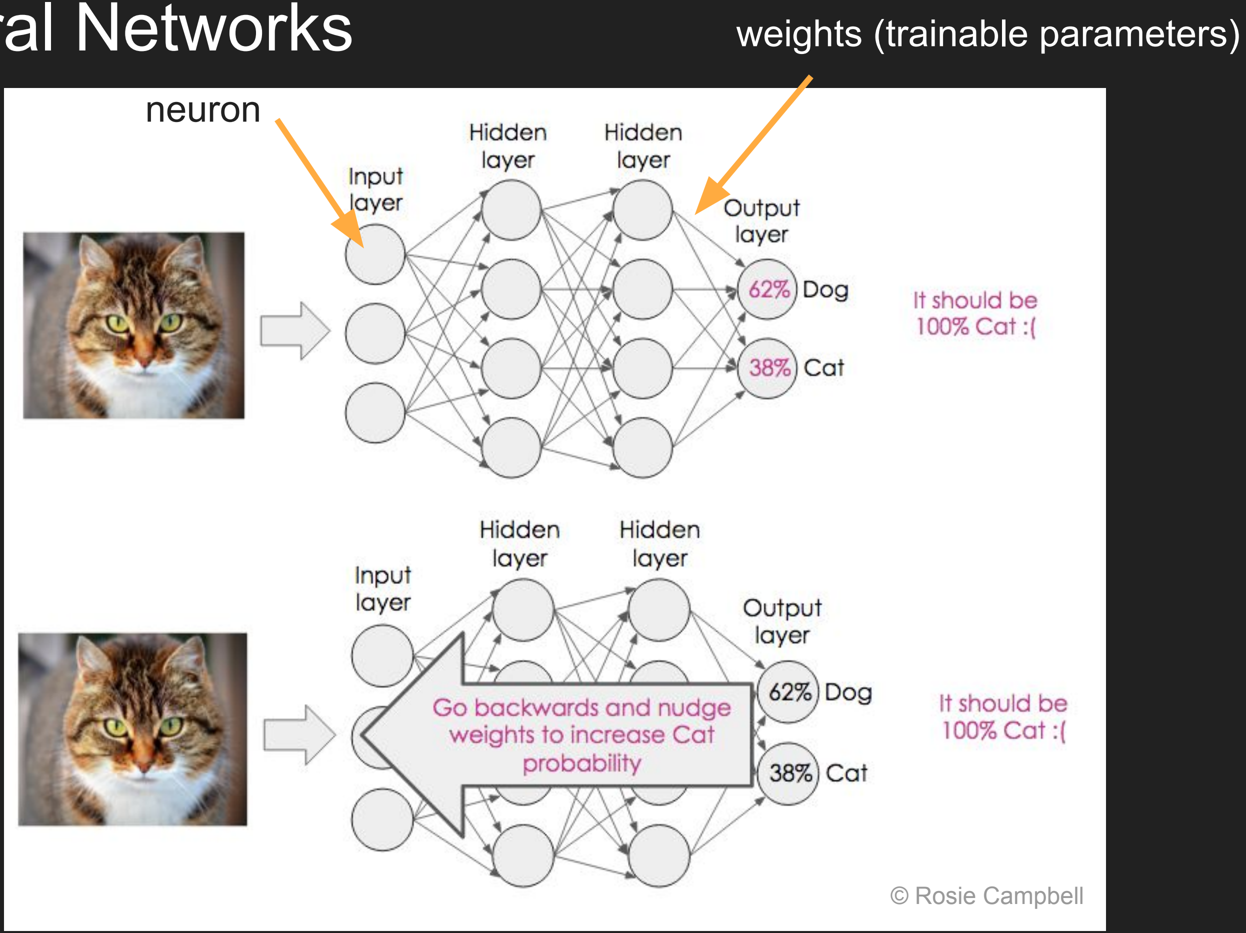
AIM

reduce to zero the numerical impact of integrating

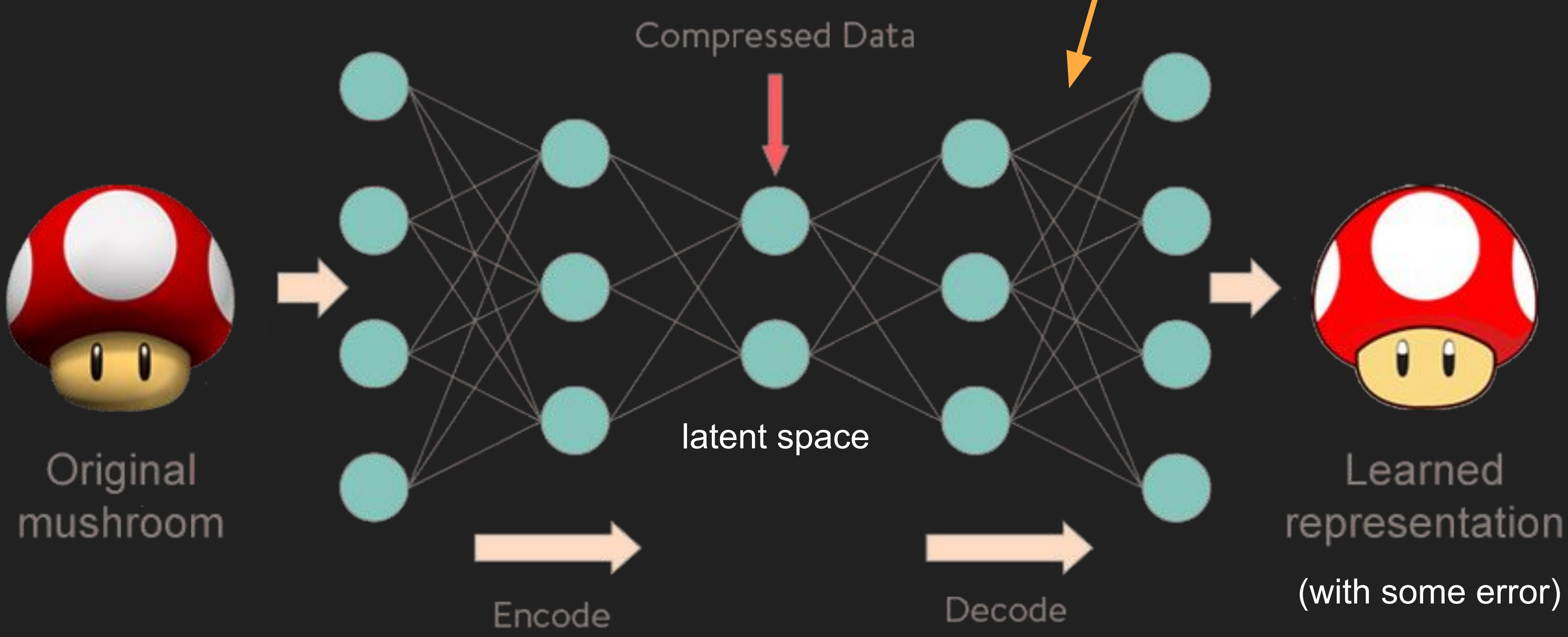
$$\dot{x} = f(\bar{x}; k)$$

with standard ODE solvers (e.g. BDF)

Deep Neural Networks



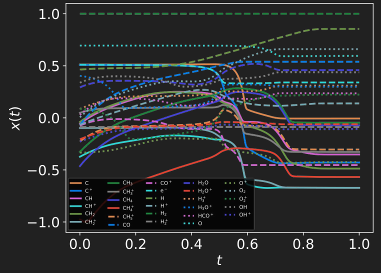
Autoencoders

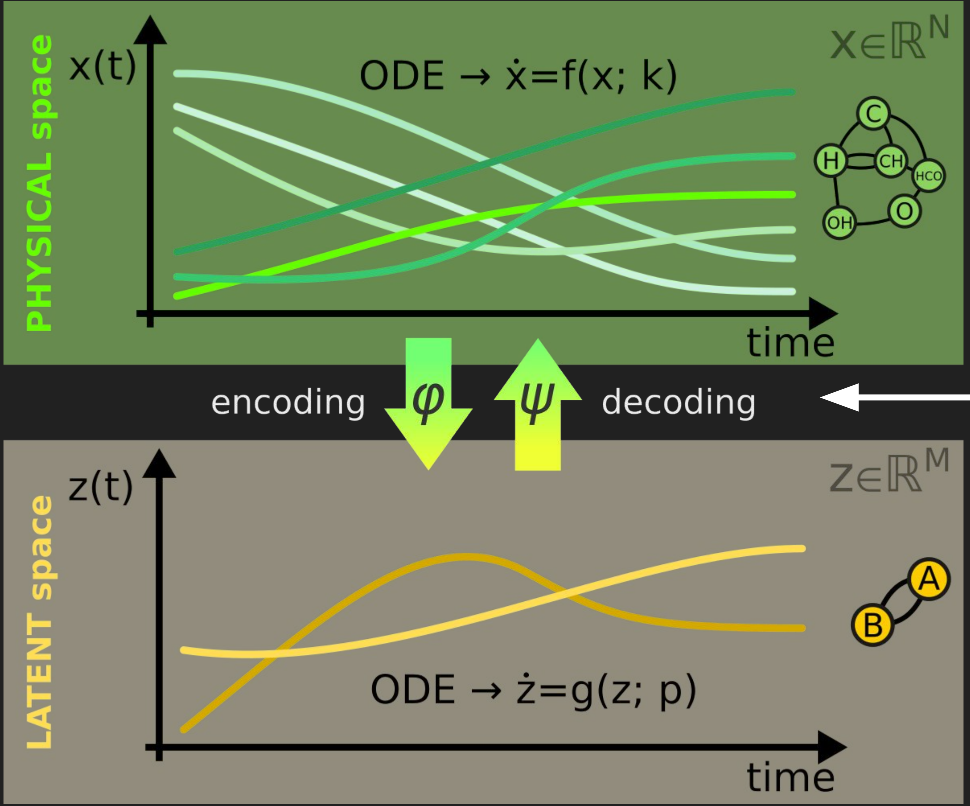


$\mathbb{R}^N$ $\mathbb{R}^{M<N}$ $\mathbb{R}^N$

AUTOENCODERS
Reduce the dimensionality of the data

Solve compressed ODE



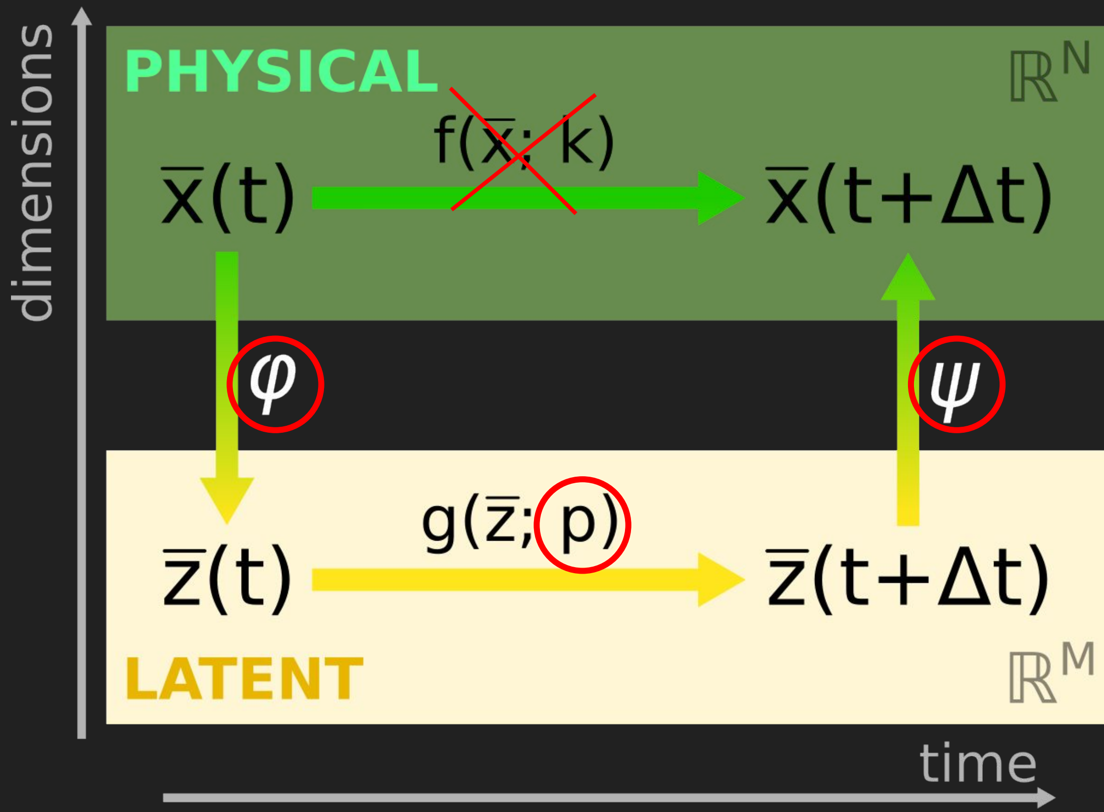


STANDARD PROBLEM

AUTOENCODER

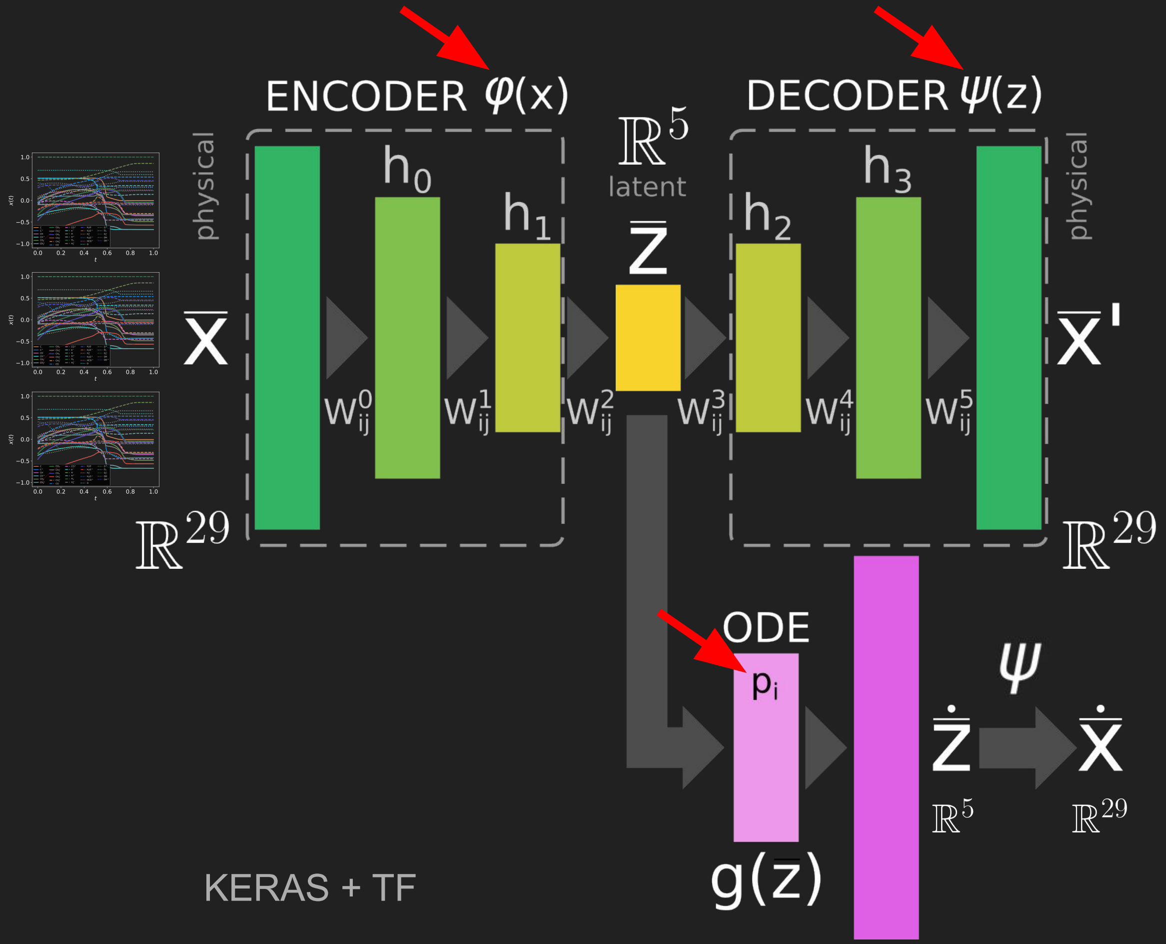
REDUCED PROBLEM

Operators



PHYSICAL
$\mathbb{R}^N$
$f(\bar{x}; k)$
$\bar{x}(t)$
$\bar{x}(t+\Delta t)$
φ
ψ
$g(\bar{z}; p)$
$\bar{z}(t)$
$\bar{z}(t+\Delta t)$
LATENT
$\mathbb{R}^M$
dimensions
time

Implementation



KERAS + TF

Latent chemical network
5 species + 12 reactions

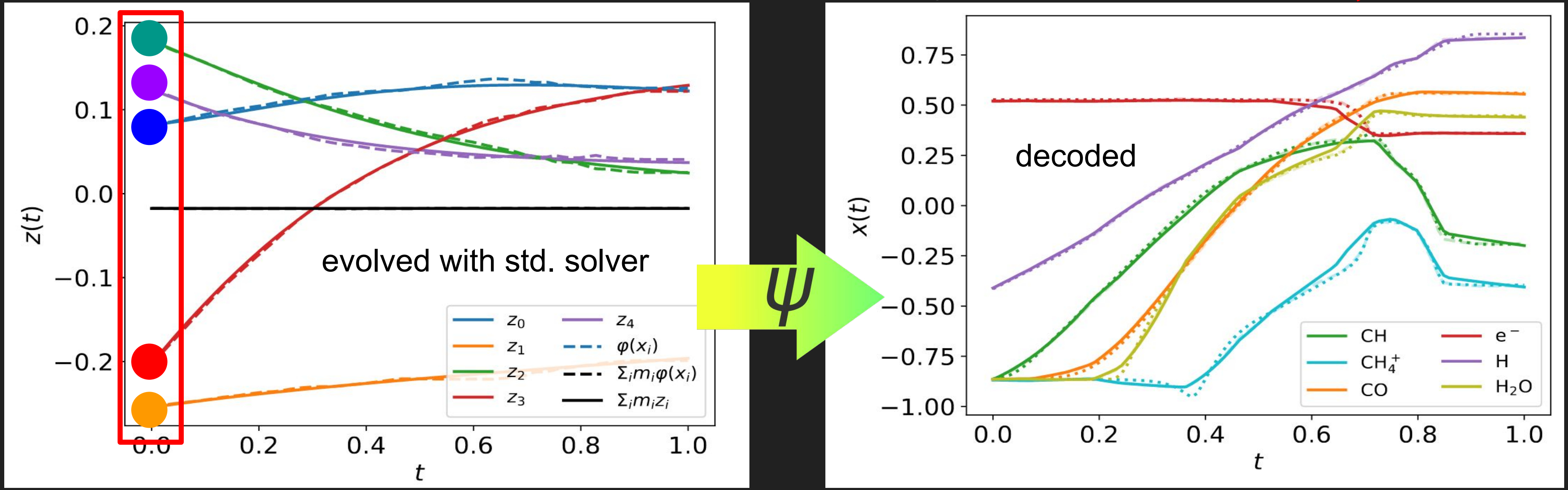
$$A + A \underset{}{\overset{p1,\, p2}{\rightleftharpoons}} AA$$
$$A + B \overset{p3,\, p4}{\rightleftharpoons} AB$$
$$B + B \overset{\dots}{\rightleftharpoons} BB$$
$$AA + BB \rightleftharpoons AB + AB$$
$$AA + B \rightleftharpoons AB + A$$
$$BB + A \rightleftharpoons AB + B$$

Original chemical network
29 species + 224 reactions

Example Results

Latent space $g(z;p)$

Physical space ~~$f(x;k)$~~



5 out of 5 “species” (A, B, AA, BB, AB)

6 out of 29 species (H, C, O, CO, CH, …)

Conclusions

Advantages:

- x65 speed-up (but in practice: integration time $\longrightarrow 0$)
- Interpretability of the latent space (it's a chemical network!)
- New perspective on the problem
- Promising (we also might learn something from latent space)

Limitations:

- $g(z; p)$ needs to be designed beforehand (but…)
- It needs training data
- This case-study uses constant temperature, CRs, radiation, ...
- Training needs some fine-tuning
- Is the compression always guaranteed? We don't know