Surrogate models in computational astrochemistry

COST NanoSpace – AI in Astrochemistry Training School 2025

Lorenzo Branca

7/August/2025





UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386



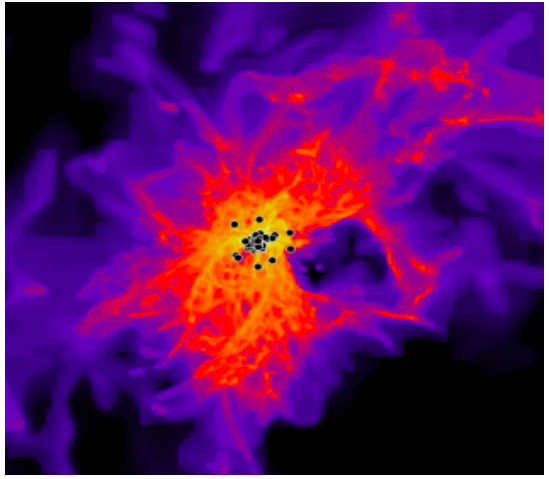
Overview

- Astrochemical evolution as ODEs system: properties, resolution and coupling with hydro simulations.
- Surrogate Modelling, how deep learning can replace ODEs solvers: methods, state of the art and some results
- Surrogate models and simulations: are we ready?

Introduction

InterStellar Medium physics

Decataldo et al;2020

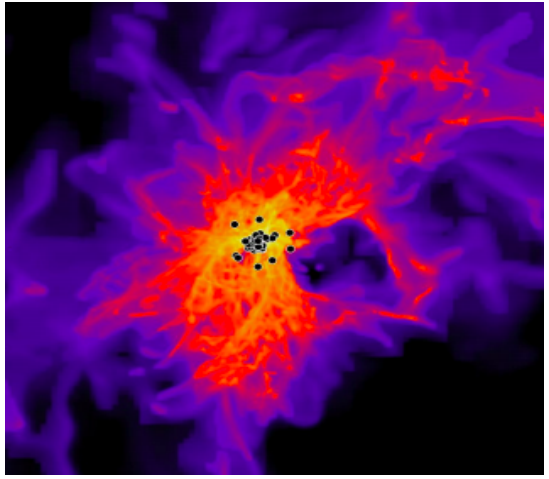


Molecular Cloud (MC)
simulation
about 10^8 finite elements

total of ~350 kCPUhr

InterStellar Medium physics

Decataldo et al;2020



processes

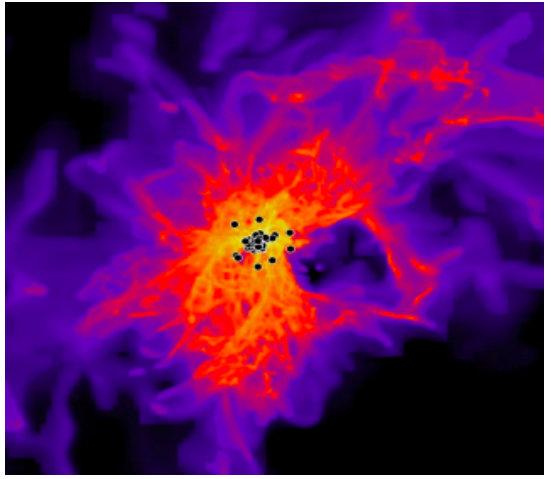
approximate CPU cost

Molecular Cloud (MC) simulation
about 10^8 finite elements

total of ~350 kCPUhr

InterStellar Medium physics

Decataldo et al;2020



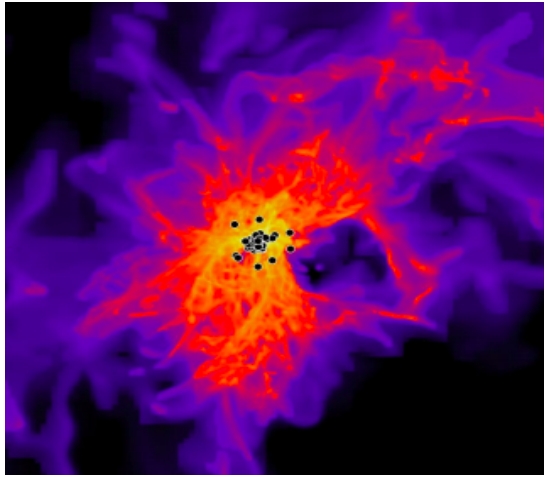
Molecular Cloud (MC) simulation
about 10^8 finite elements

total of ~350 kCPUhr

processes		approximate CPU cost
Fluid dynamics	$\partial_t \mathcal{U} + \nabla \cdot \mathcal{F} = \mathcal{S}$	~20%

InterStellar Medium physics

Decataldo et al;2020



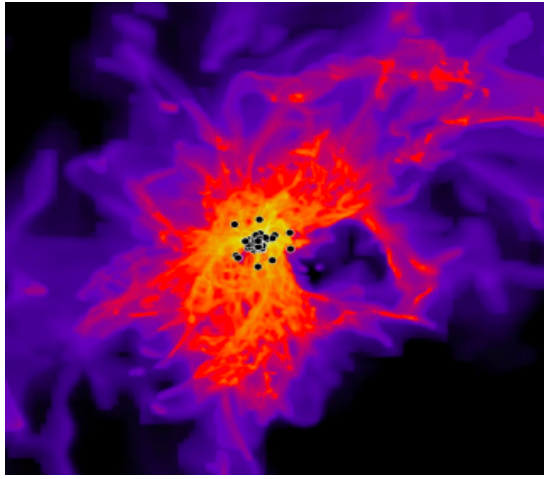
Molecular Cloud (MC) simulation
about 10^8 finite elements

total of ~350 kCPUhr

processes		approximate CPU cost
Fluid dynamics	$\partial_t \mathcal{U} + \nabla \cdot \mathcal{F} = \mathcal{S}$	~20%
(self)gravity	$\nabla^2 \Phi = 4\pi G \rho$	~10%

InterStellar Medium physics

Decataldo et al;2020



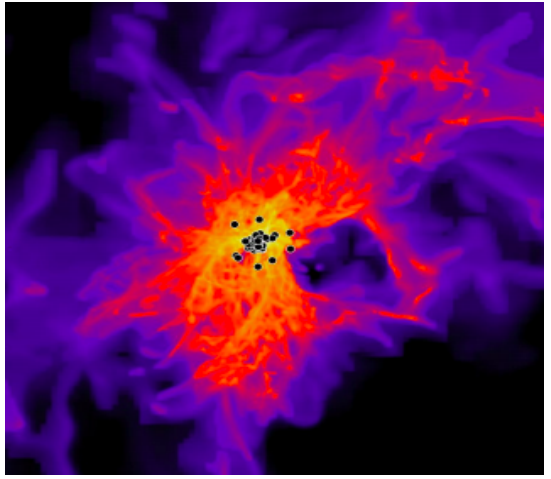
Molecular Cloud (MC) simulation
about 10^8 finite elements

total of ~350 kCPUhr

processes		approximate CPU cost
Fluid dynamics	$\partial_t \mathcal{U} + \nabla \cdot \mathcal{F} = \mathcal{S}$	~20%
(self)gravity	$\nabla^2 \Phi = 4\pi G \rho$	~10%
Radiative transfer	$\partial_t I_\nu + \hat{n} \nabla \cdot I_\nu = j_\nu - k_\nu I_\nu$	~20%

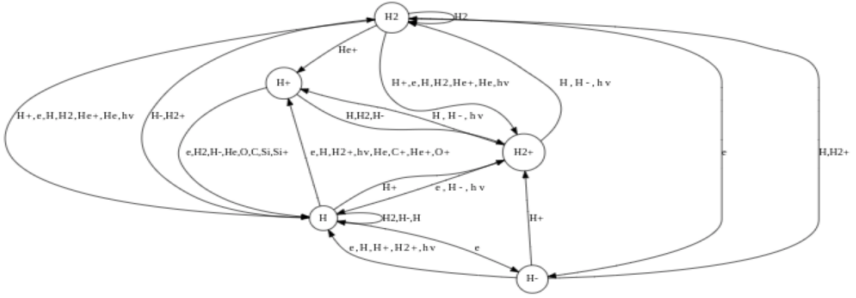
InterStellar Medium physics

Decataldo et al;2020



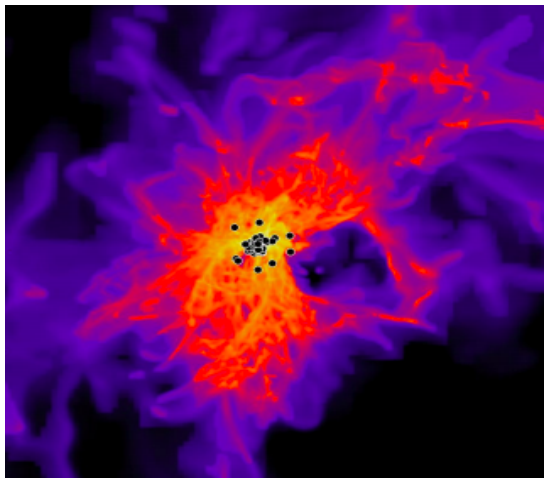
Molecular Cloud (MC) simulation
about 10^8 finite elements

total of ~350 kCPUhr

processes		approximate CPU cost
Fluid dynamics	$\partial_t \mathcal{U} + \nabla \cdot \mathcal{F} = \mathcal{S}$	~20%
(self)gravity	$\nabla^2 \Phi = 4\pi G \rho$	~10%
Radiative transfer	$\partial_t I_\nu + \hat{n} \nabla \cdot I_\nu = j_\nu - k_\nu I_\nu$	~20%
chemistry		~50%

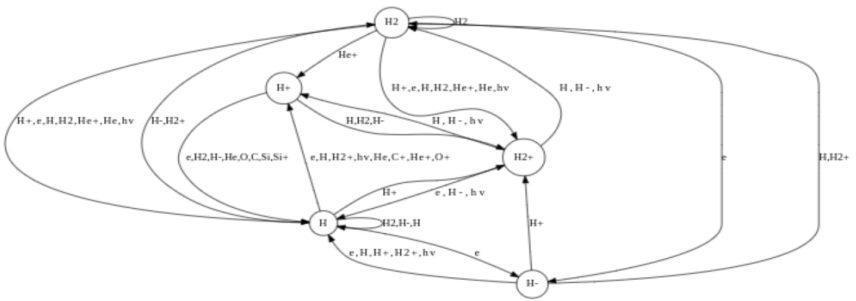
InterStellar Medium physics

Decataldo et al;2020



Molecular Cloud (MC) simulation
about 10^8 finite elements

total of ~350 kCPUhr

processes		approximate CPU cost
Fluid dynamics	$\partial_t \mathcal{U} + \nabla \cdot \mathcal{F} = \mathcal{S}$	~20%
(self)gravity	$\nabla^2 \Phi = 4\pi G \rho$	~10%
Radiative transfer	$\partial_t I_\nu + \hat{n} \nabla \cdot I_\nu = j_\nu - k_\nu I_\nu$	~20%
chemistry		~50%

non-equilibrium chemistry evolution plays a crucial role in cosmological and astrophysical phenomena, especially in the study of InterStellar Medium (ISM)

ISM (photo-thermo)-chemistry

ISM (photo-thermo)-chemistry

$$\dot{n}_k = A_k^{ij} n_i n_j + B_k^i n_i$$

$$\dot{T} = \Gamma - \Lambda$$

ISM (photo-thermo)-chemistry

$$\dot{n}_k = \overbrace{A_k^{ij}}^{(T)} n_i n_j + \overbrace{B_k^i}^{(\mathbf{F},\, CR)} n_i$$

$$\dot{T} = \overbrace{\Gamma - \Lambda}^{(\mathbf{n},\, T,\, \mathbf{F},\, CR)}$$

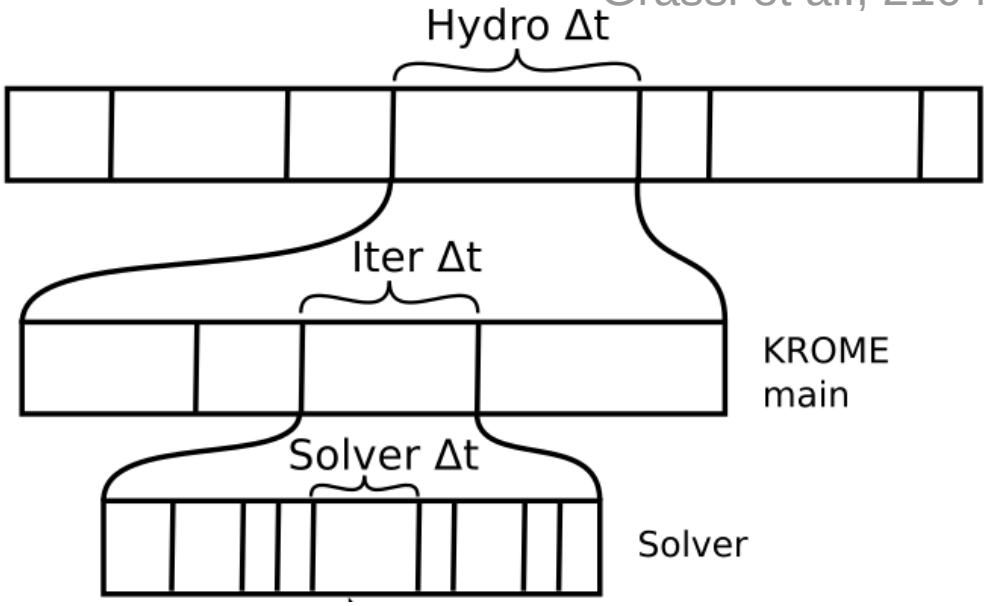
ISM (photo-thermo)-chemistry

(*T*) (**F**, *CR*)

$$\dot{n}_k = A_k^{ij} n_i n_j + B_k^i n_i$$

$$\dot{T} = \Gamma - \Lambda$$ (**n**, *T*, **F**, *CR*)

Grassi et al., 2104



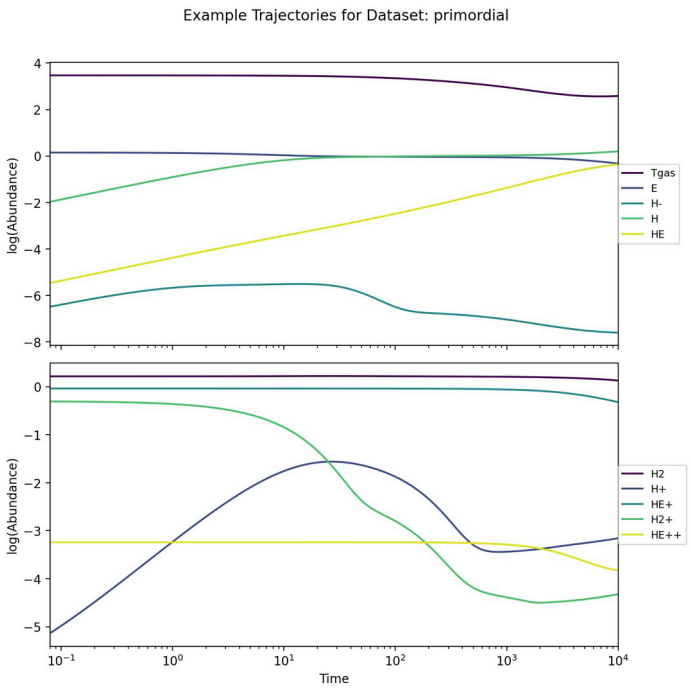
ISM (photo-thermo)-chemistry

$$\dot{n}_k = A_k^{ij} n_i n_j + B_k^i n_i$$

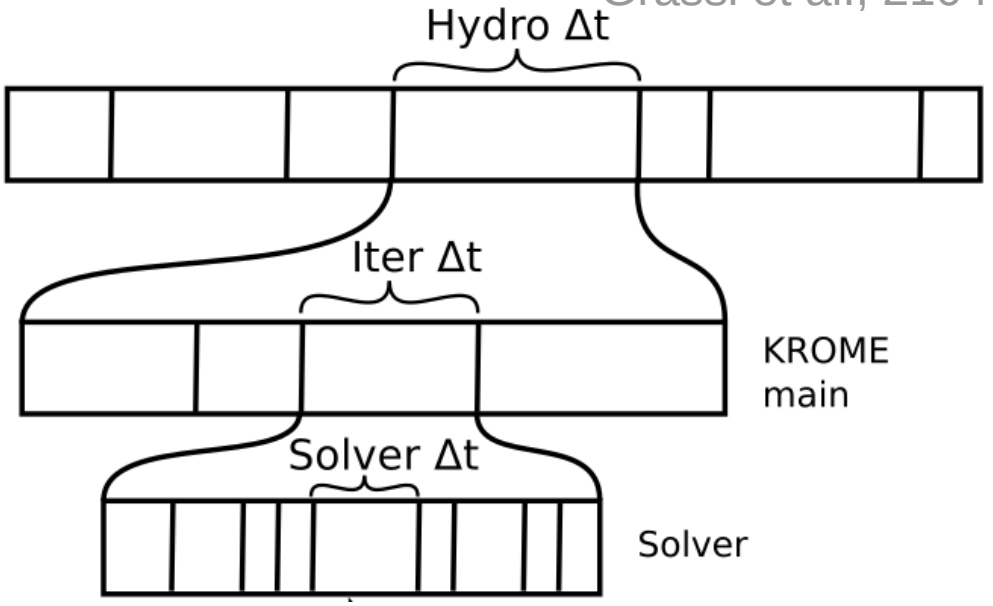
(T) ($\mathbf{F}$, CR)

$$\dot{T} = \Gamma - \Lambda$$

($\mathbf{n}$, T, $\mathbf{F}$, CR)



Grassi et al., 2104



ISM (photo-thermo)-chemistry

ISM (photo-thermo)-chemistry

$\tau_{chem}/\tau_{HD} < 10^{-4}$ ⟶ Higher number of time step

ISM (photo-thermo)-chemistry

$\tau_{chem}/\tau_{HD} < 10^{-4}$ ⟶ Higher number of time step

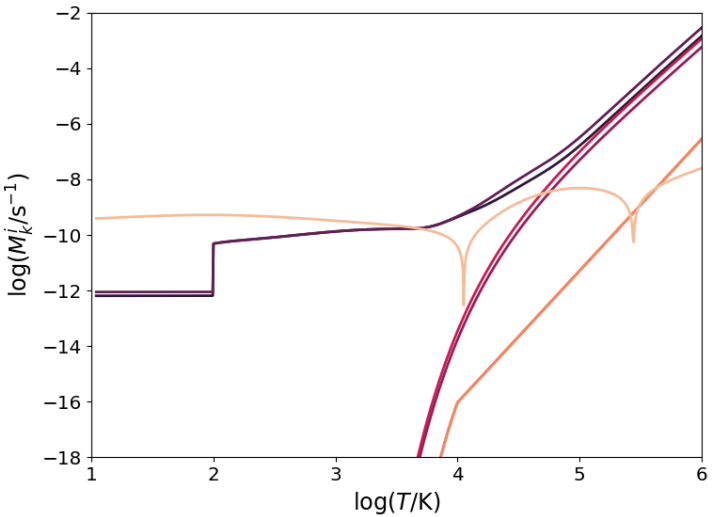
$N_{rea} \propto N_s^{\alpha}, \alpha > 1$ ⟶ Higher floating point operations

ISM (photo-thermo)-chemistry

$\tau_{chem}/\tau_{HD} < 10^{-4}$ → Higher number of time step

$N_{rea} \propto N_s^{\alpha}, \alpha > 1$ → Higher floating point operations

Stifness $M_k^i = A_k^{ij} n_j + B_k^i$



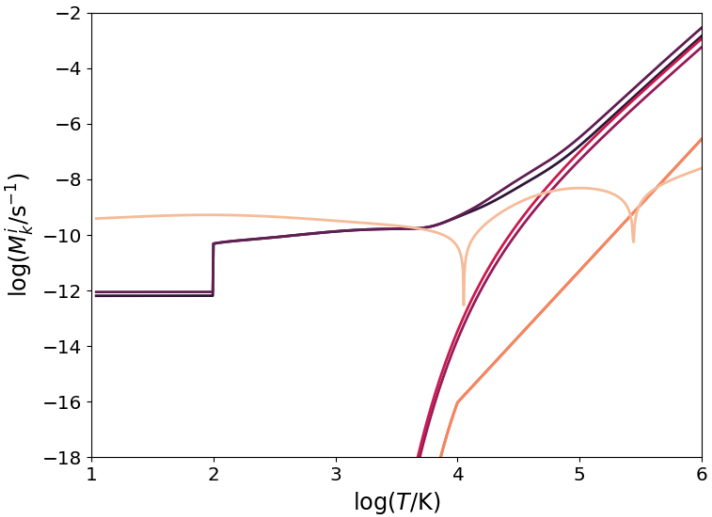
→ Requires implicit solvers

ISM (photo-thermo)-chemistry

$\tau_{chem}/\tau_{HD} < 10^{-4}$ → Higher number of time step

$N_{rea} \propto N_s^{\alpha}, \alpha > 1$ → Higher floating point operations

Stifness $M_k^i = A_k^{ij} n_j + B_k^i$



→ Requires implicit solvers

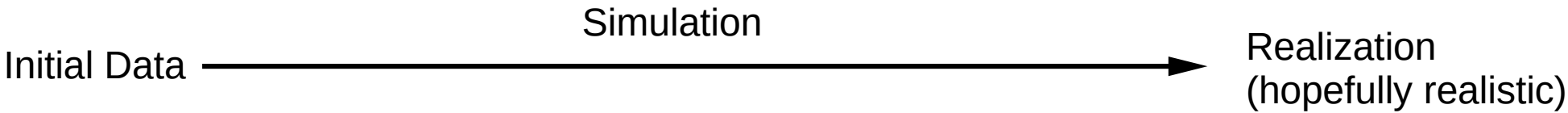
Load balancing → Decrease efficiency

Emulators/surrogate models

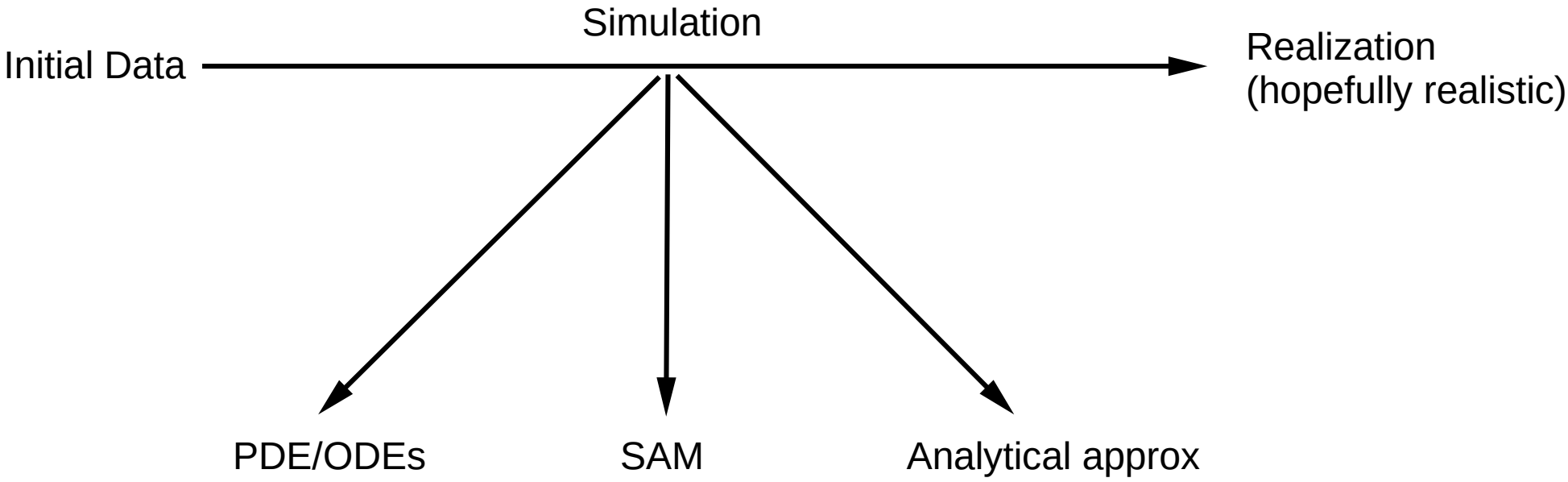
Emulators/surrogate models

Initial Data

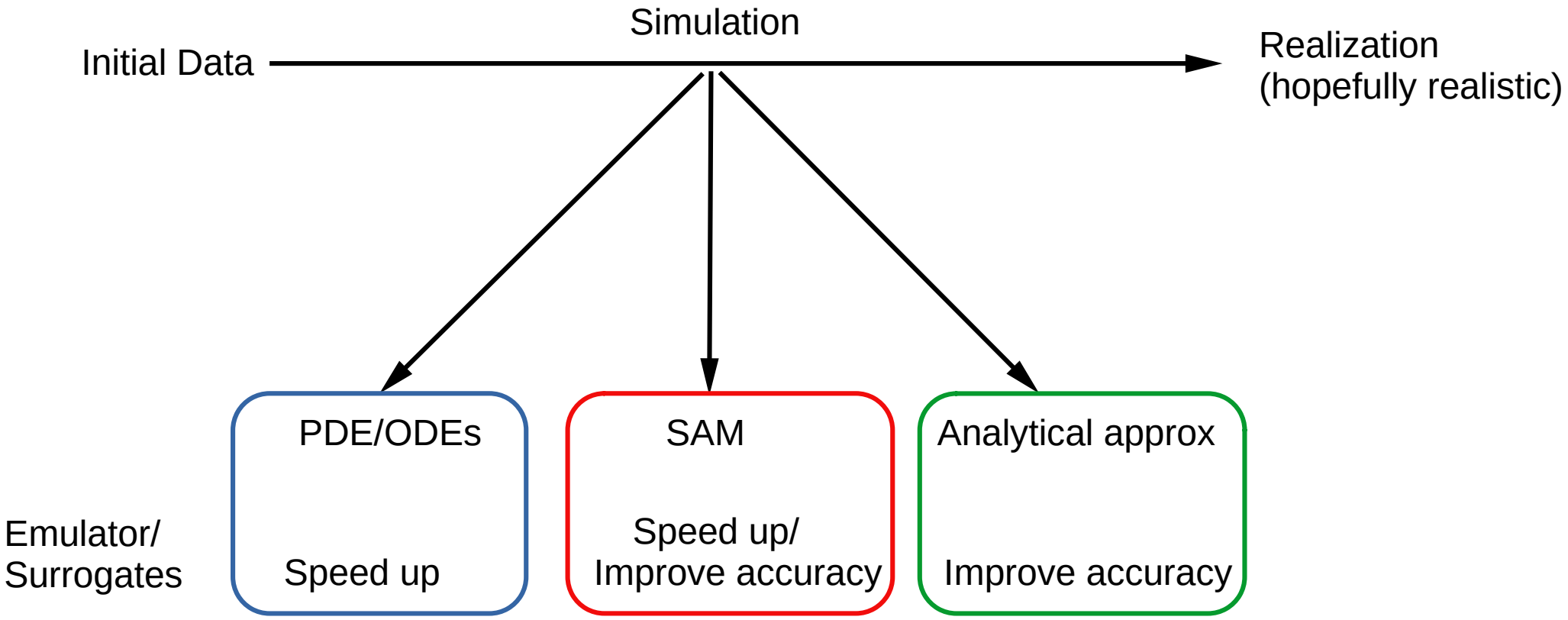
Emulators/surrogate models



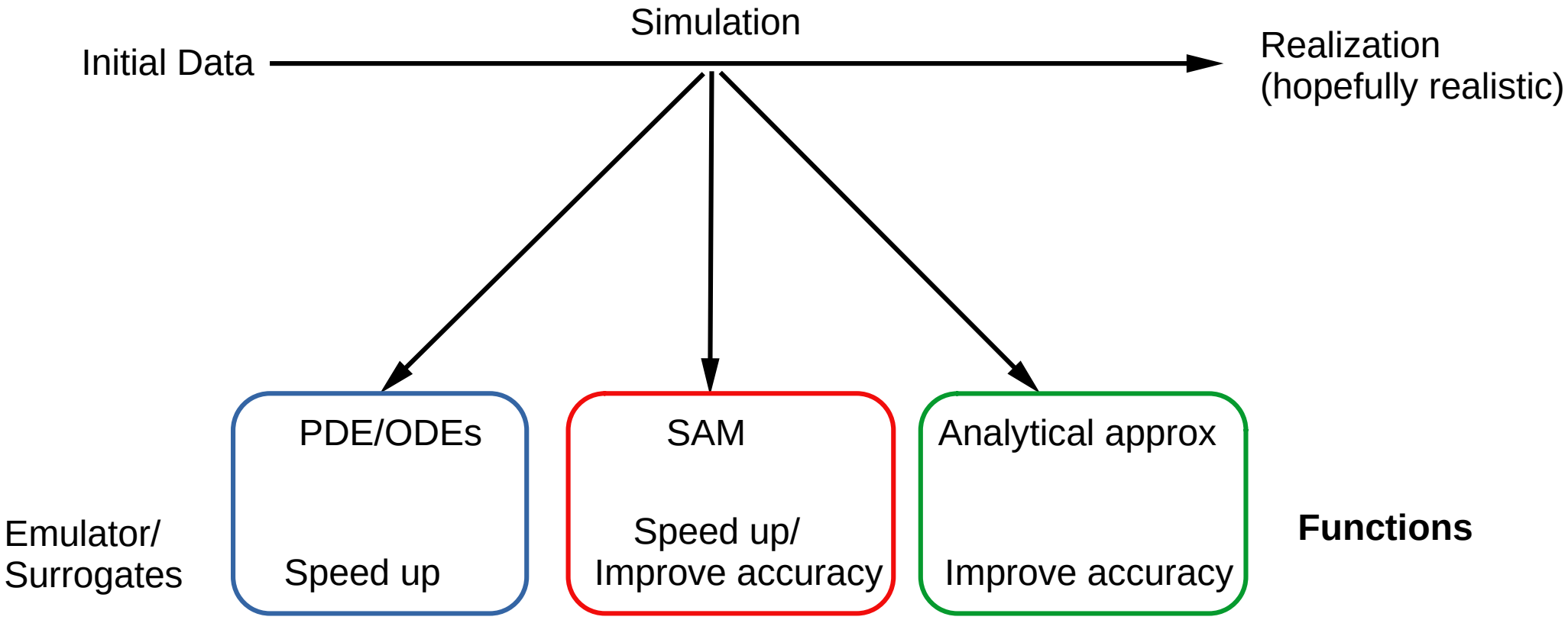
Emulators/surrogate models

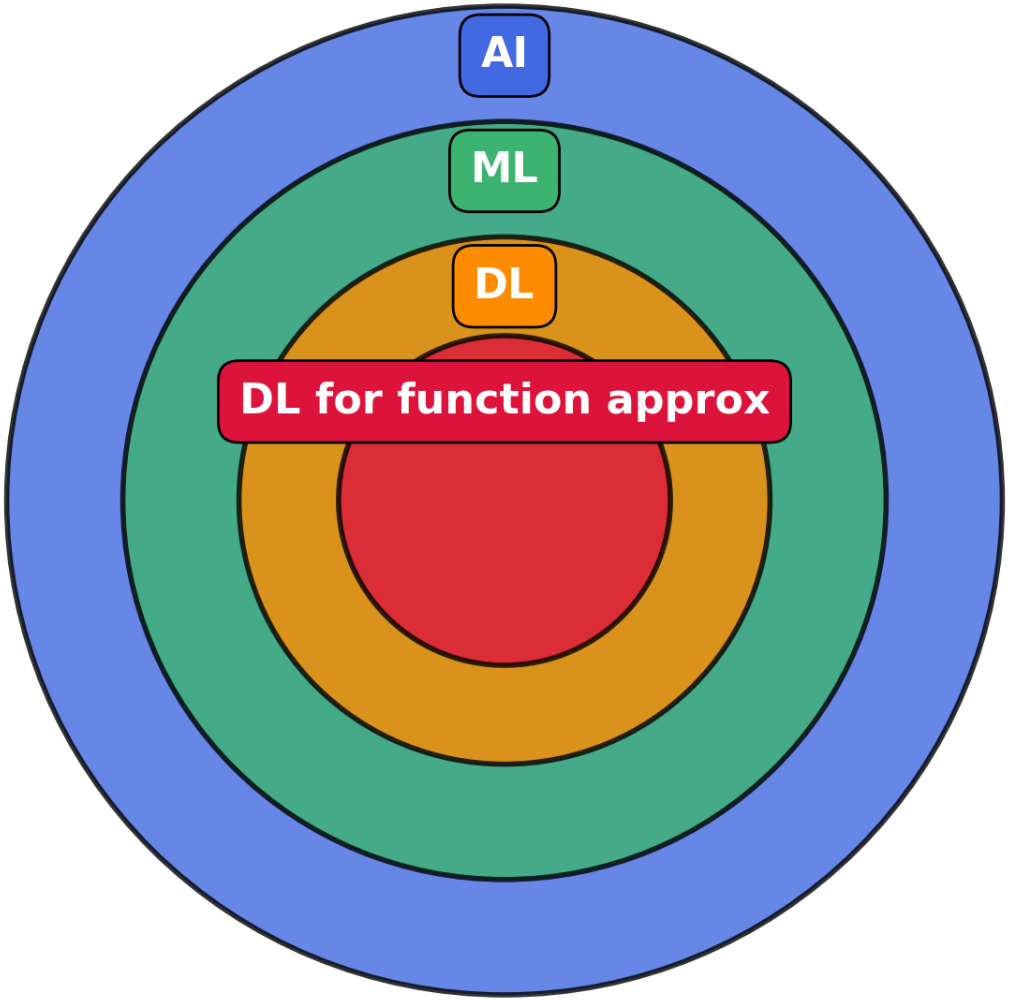


Emulators/surrogate models



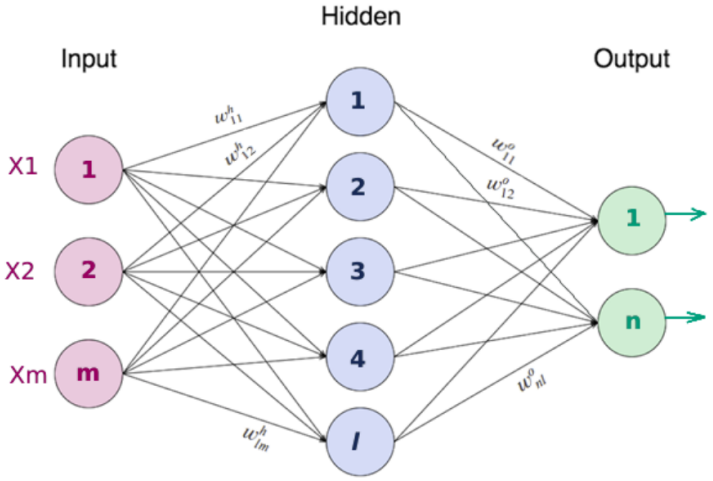
Emulators/surrogate models




AI
ML
DL
DL for function approx

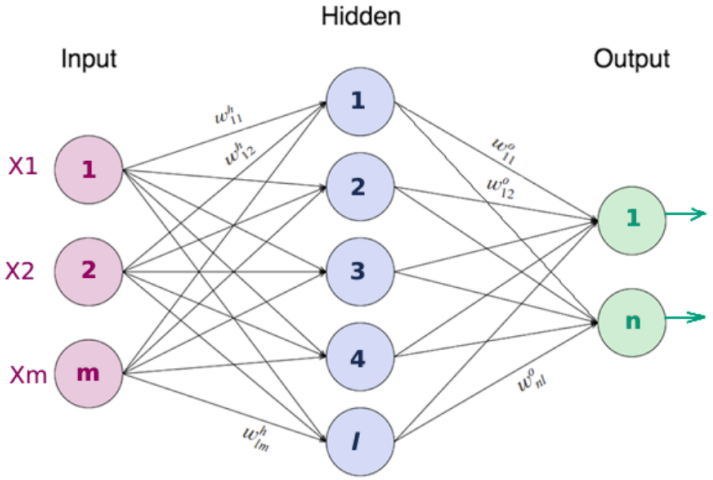
Neural networks as functions approximators

Neural networks as functions approximators



$$y_k = \sigma_k(\theta_{ik}, x_j) \circ \ldots \circ \sigma_1(\theta_{i1}, x_1)$$

Neural networks as functions approximators



$$y_k = \sigma_k(\theta_{ik}, x_j) \circ \ldots \circ \sigma_1(\theta_{i1}, x_1)$$

Universal approximation theorem — Let $C(X, \mathbb{R}^m)$ denote the set of continuous functions from a subset X of a Euclidean $\mathbb{R}^n$ space to a Euclidean space $\mathbb{R}^m$. Let $\sigma \in C(\mathbb{R}, \mathbb{R})$. Note that $(\sigma \circ x)_i = \sigma(x_i)$, so $\sigma \circ x$ denotes σ applied to each component of x.

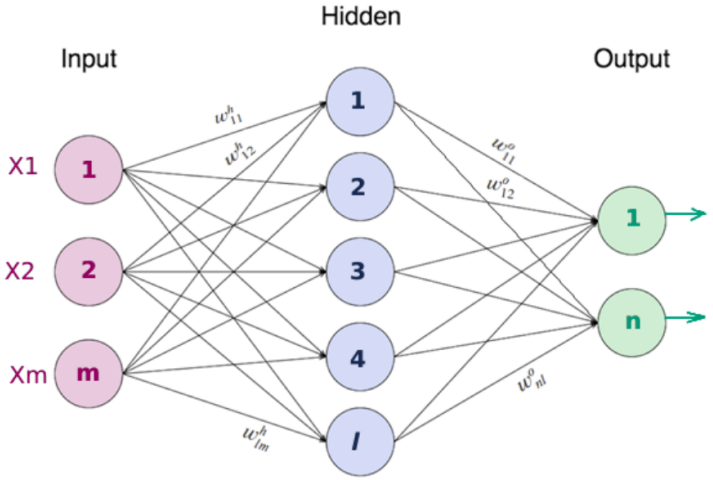
Then σ is not polynomial if and only if for every $n \in \mathbb{N}$, $m \in \mathbb{N}$, compact $K \subseteq \mathbb{R}^n$, $f \in C(K, \mathbb{R}^m)$, $\varepsilon > 0$ there exist $k \in \mathbb{N}$, $A \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, $C \in \mathbb{R}^{m \times k}$ such that

$$\sup_{x \in K} \|f(x) - g(x)\| < \varepsilon$$

where $g(x) = C \cdot (\sigma \circ (A \cdot x + b))$

Cybenko, 1989

Neural networks as functions approximators



$$y_k = \sigma_k(\theta_{ik}, x_j) \circ \ldots \circ \sigma_1(\theta_{i1}, x_1)$$

$$\mathcal{L}_\theta = D(f(x), y_\theta(x)) : \mathbb{R}^n \to \mathbb{R}, .$$

Universal approximation theorem — Let $C(X, \mathbb{R}^m)$ denote the set of continuous functions from a subset X of a Euclidean $\mathbb{R}^n$ space to a Euclidean space $\mathbb{R}^m$. Let $\sigma \in C(\mathbb{R}, \mathbb{R})$. Note that $(\sigma \circ x)_i = \sigma(x_i)$, so $\sigma \circ x$ denotes σ applied to each component of x.

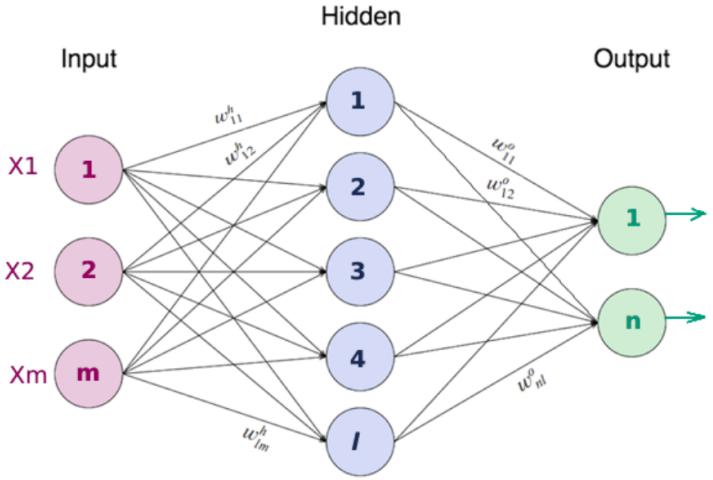
Then σ is not polynomial if and only if for every $n \in \mathbb{N}$, $m \in \mathbb{N}$, compact $K \subseteq \mathbb{R}^n$, $f \in C(K, \mathbb{R}^m)$, $\varepsilon > 0$ there exist $k \in \mathbb{N}$, $A \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, $C \in \mathbb{R}^{m \times k}$ such that

$$\sup_{x \in K} \|f(x) - g(x)\| < \varepsilon$$

where $g(x) = C \cdot (\sigma \circ (A \cdot x + b))$

Cybenko, 1989

Neural networks as functions approximators



$$y_k = \sigma_k(\theta_{ik}, x_j) \circ \ldots \circ \sigma_1(\theta_{i1}, x_1)$$

$$\mathcal{L}_\theta = D(f(x), y_\theta(x)) : \mathbb{R}^n \rightarrow \mathbb{R}, .$$

$$\partial_{\theta_{ij}} \mathcal{L}_{\theta_{ij}}$$

Universal approximation theorem — Let $C(X, \mathbb{R}^m)$ denote the set of continuous functions from a subset X of a Euclidean $\mathbb{R}^n$ space to a Euclidean space $\mathbb{R}^m$. Let $\sigma \in C(\mathbb{R}, \mathbb{R})$. Note that $(\sigma \circ x)_i = \sigma(x_i)$, so $\sigma \circ x$ denotes σ applied to each component of x.

Then σ is not polynomial if and only if for every $n \in \mathbb{N}$, $m \in \mathbb{N}$, compact $K \subseteq \mathbb{R}^n$, $f \in C(K, \mathbb{R}^m)$, $\varepsilon > 0$ there exist $k \in \mathbb{N}$, $A \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, $C \in \mathbb{R}^{m \times k}$ such that

$$\sup_{x \in K} \|f(x) - g(x)\| < \varepsilon$$

where $g(x) = C \cdot (\sigma \circ (A \cdot x + b))$

Cybenko, 1989

What is needed?

What is needed?

$$(t, \mathbf{IC}, \mathbf{p}) \rightarrow \mathrm{model}_{\theta}(t, \mathbf{IC}, \mathbf{p}) = (\mathbf{n}(t), T(t))$$

What is needed?

$$(t, \mathbf{IC}, \mathbf{p}) \rightarrow \mathrm{model}_\theta(t, \mathbf{IC}, \mathbf{p}) = (\mathbf{n}(t), T(t))$$

$$\mathrm{min}_\theta ||\mathrm{data} - \mathrm{model}_\theta(t, \mathbf{IC}, \mathbf{p})||$$

$$\mathcal{L}_\theta = ||\mathrm{data} - NN_\theta(t, \mathrm{IC})(\mathbf{p})||$$

What is needed?

$$(t, \mathbf{IC}, \mathbf{p}) \rightarrow \mathrm{model}_\theta(t, \mathbf{IC}, \mathbf{p}) = (\mathbf{n}(t), T(t))$$

$$\min_\theta ||\mathrm{data} - \mathrm{model}_\theta(t, \mathbf{IC}, \mathbf{p})||$$

~~$$\mathcal{L}_\theta = ||\mathrm{data} - NN_\theta(t, \mathrm{IC})(\mathbf{p})||$$~~

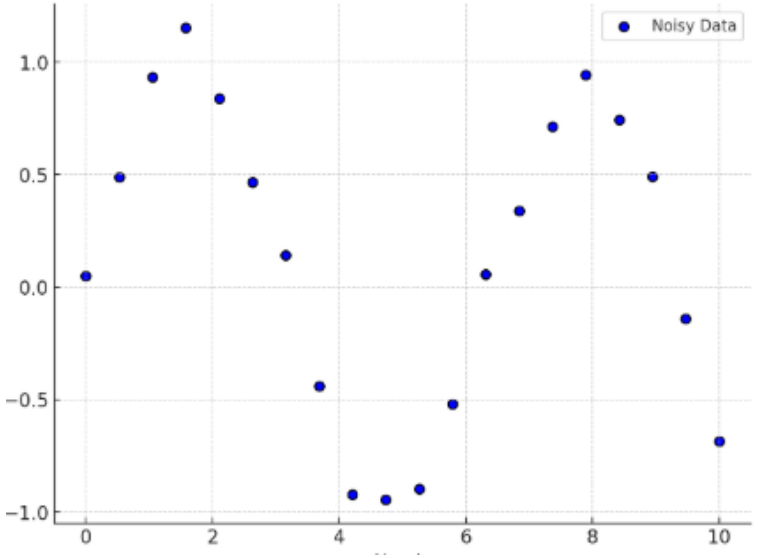
Some works on the topic

- Grassi et al., 2022
- Branca & Pallottini, 2023 and 2024
- Sulzer & Buck, 2023
- Jensen & Buck, 2024
- Holdship et al., 2021
- Asensio Ramos et al., 2024
- Vermariën et al., 2024
- Maes et al., 2024
- And more...

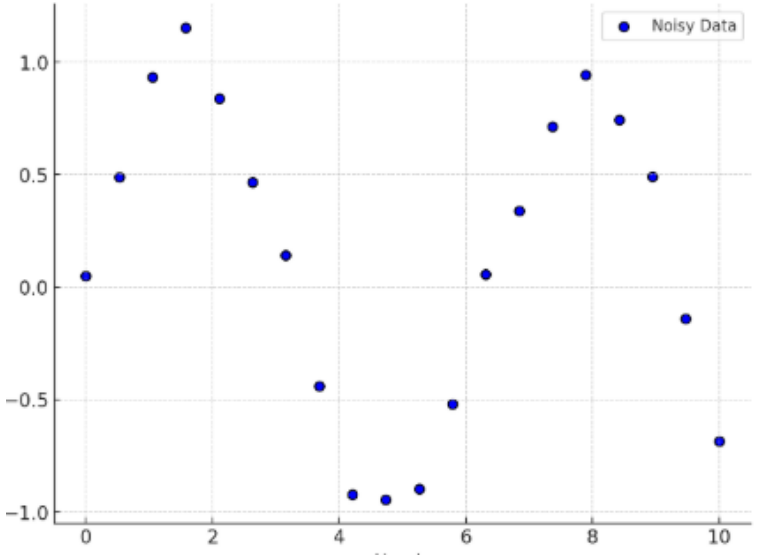
Neural Ordinary differential Equations (and coupling with autoencoders)

NODEs theory

NODEs theory

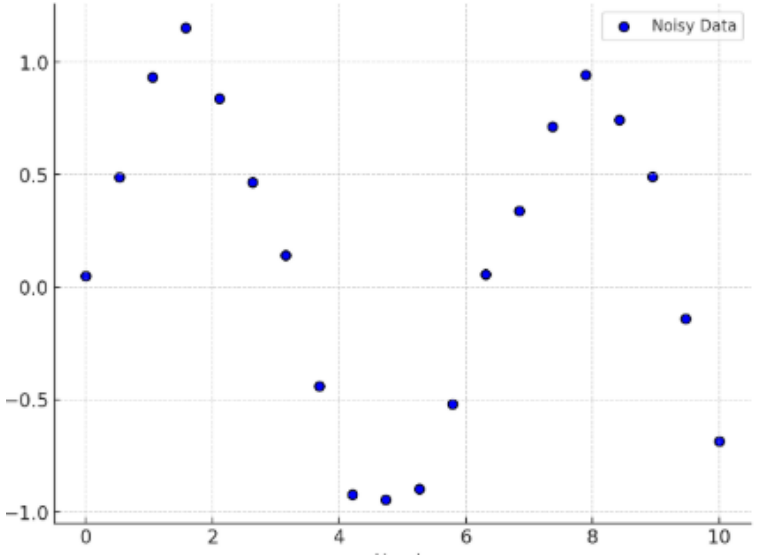


NODEs theory



$$\dot{y} = f(x, y)$$

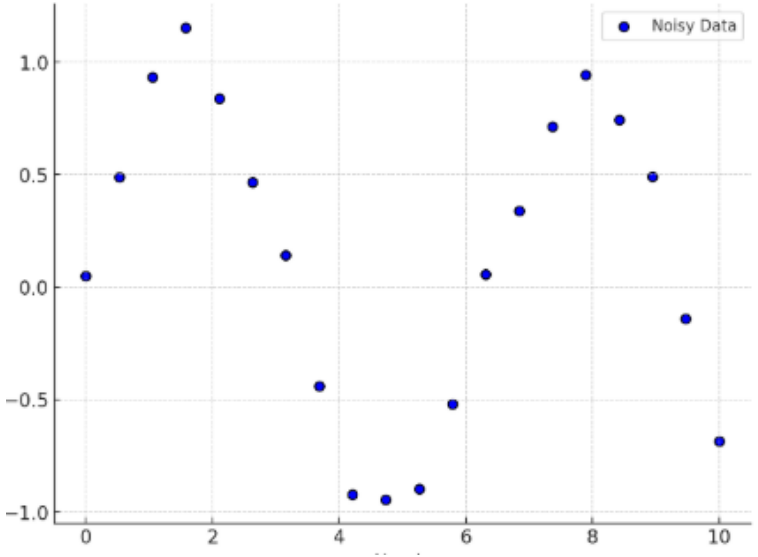
NODEs theory



$$\dot{y} = f(x, y)$$

$$\dot{y} = f_\theta(x, y)$$

NODEs theory

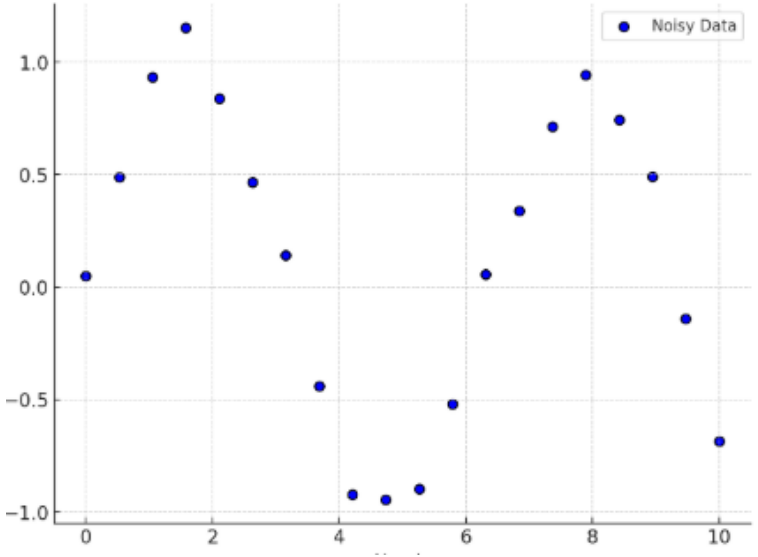


$$\dot{y} = f(x, y)$$

$$\dot{y} = f_\theta(x, y)$$

$$y = \mathrm{ODEsolve}(f_\theta(x, y))$$

NODEs theory



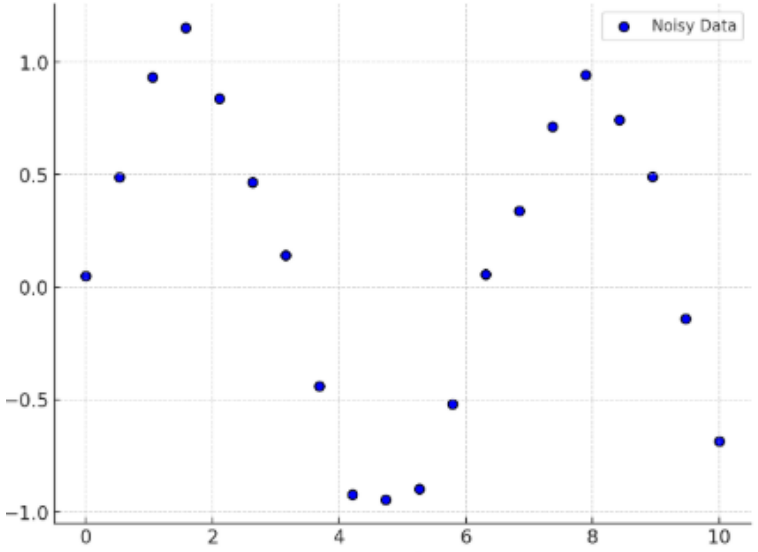
$$\dot{y} = f(x, y)$$

$$\dot{y} = f_\theta(x, y)$$

$$y = \text{ODEsolve}(f_\theta(x, y))$$

$$f \simeq argmin_\theta ||y_{data} - \text{ODEsolve}(f_\theta(x, y))||$$

NODEs theory

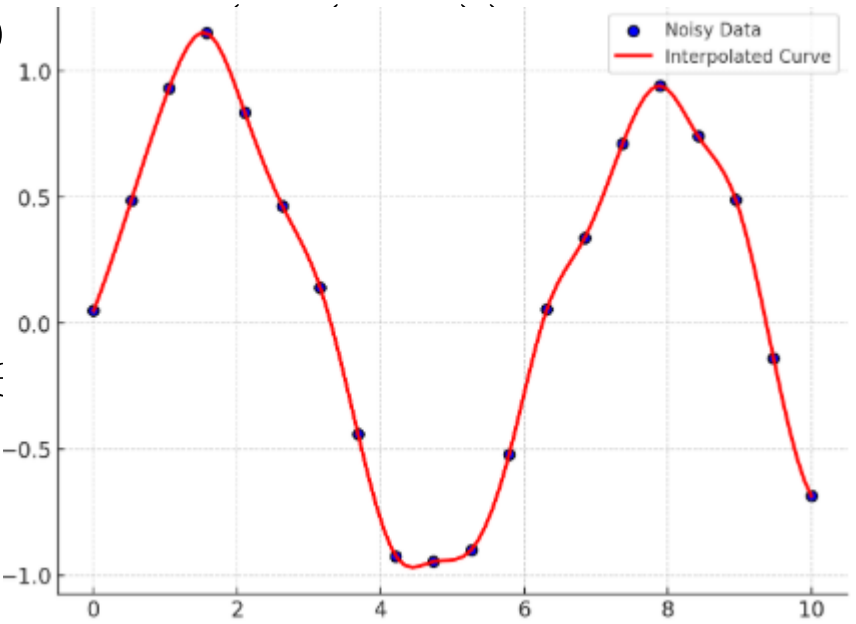


$$\dot{y} = f(x, y)$$

$$\dot{y} = f_\theta(x, y)$$

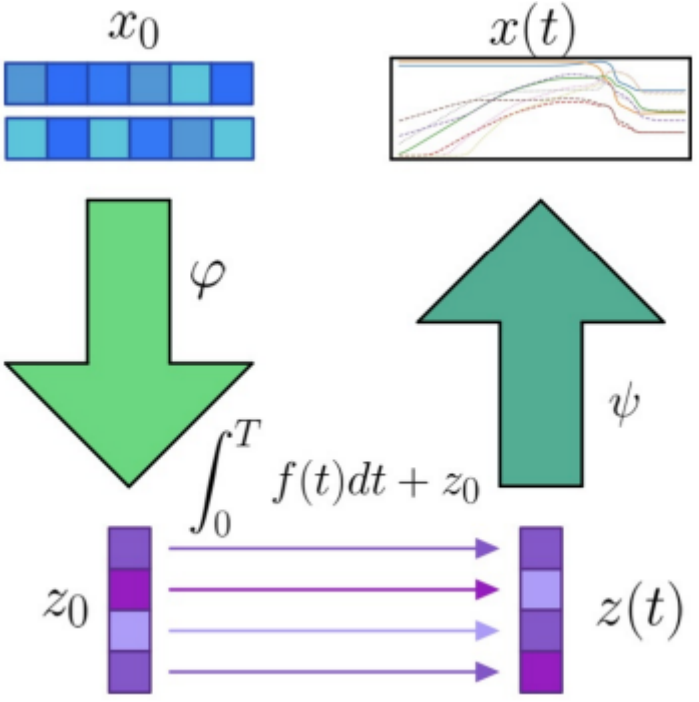
$$y = \mathrm{OD}$$

$$f \simeq argmin_\theta ||y_{data} - \mathrm{ODEsolve}$$



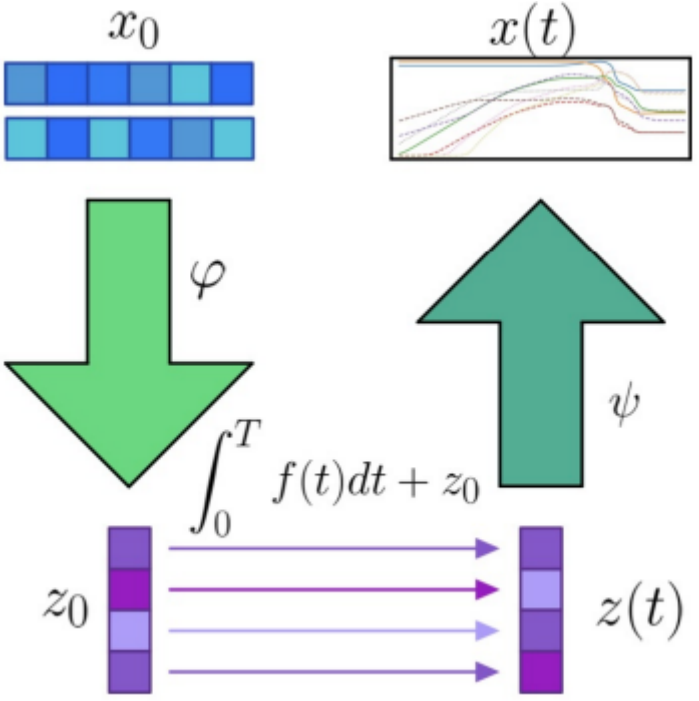
NODEs + autoencoders

NODEs + autoencoders

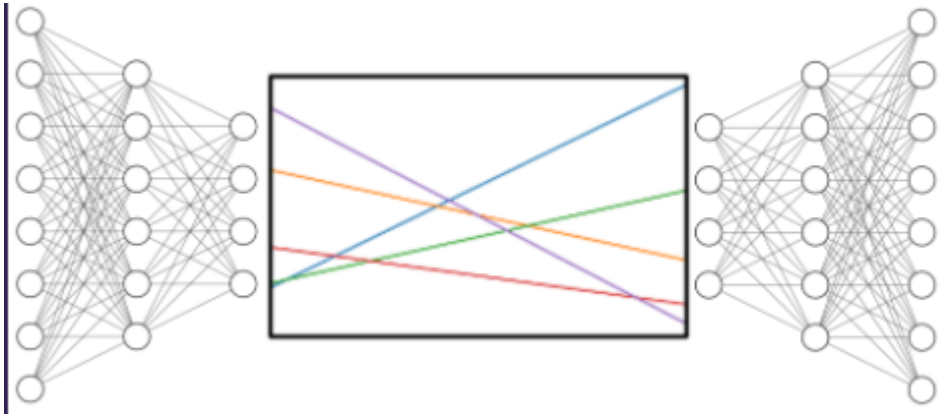


Sulzer & Buck, 2023

NODEs + autoencoders

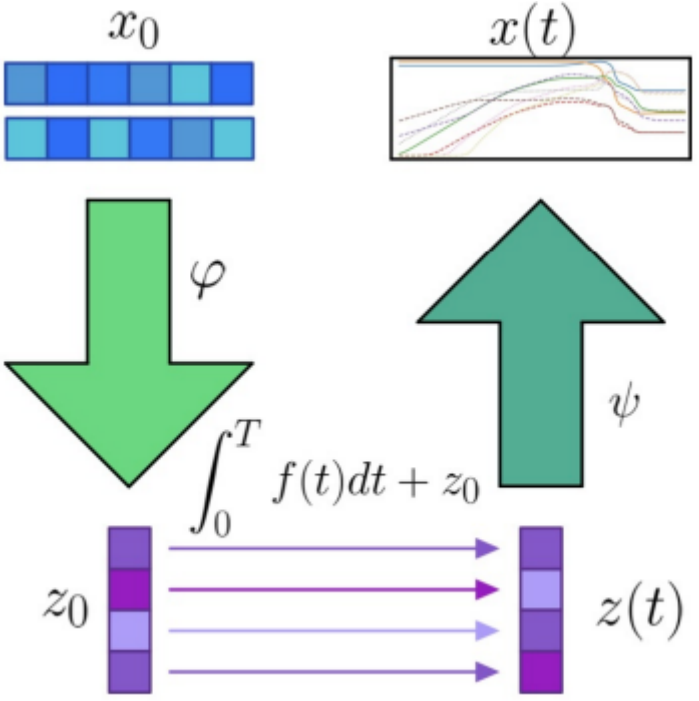


Sulzer & Buck, 2023

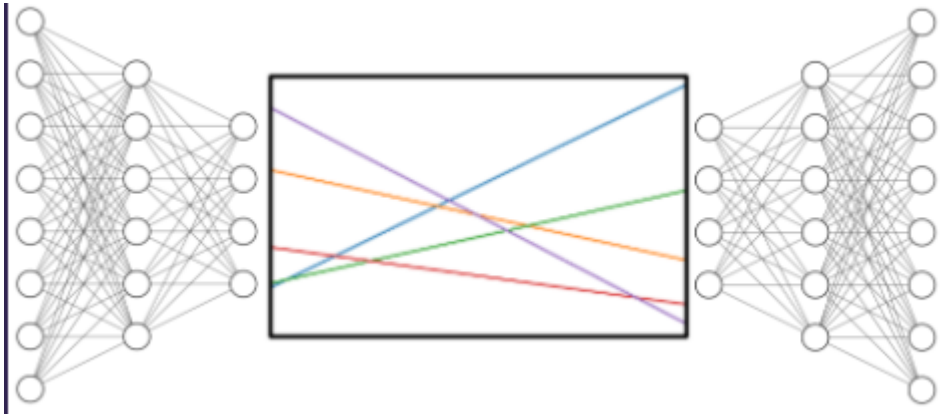


$$f_{\theta*} \simeq poly$$

NODEs + autoencoders



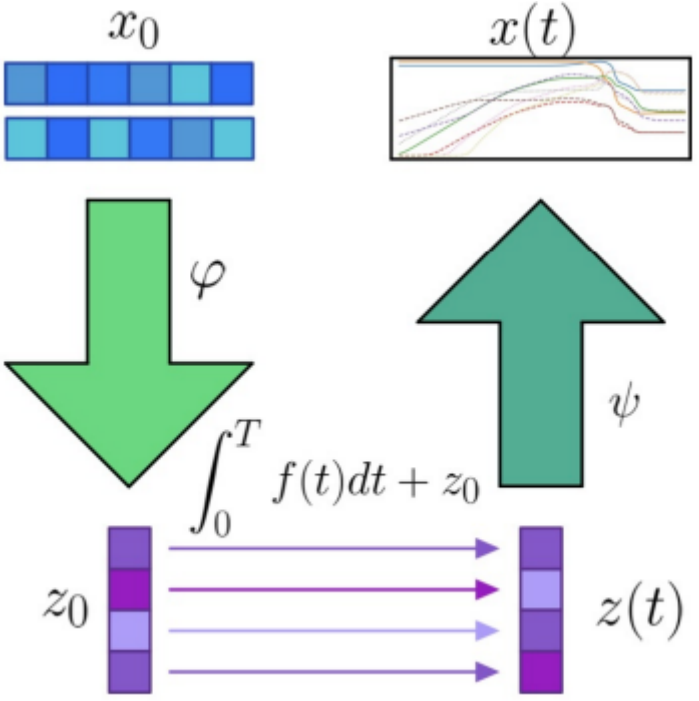
Sulzer & Buck, 2023



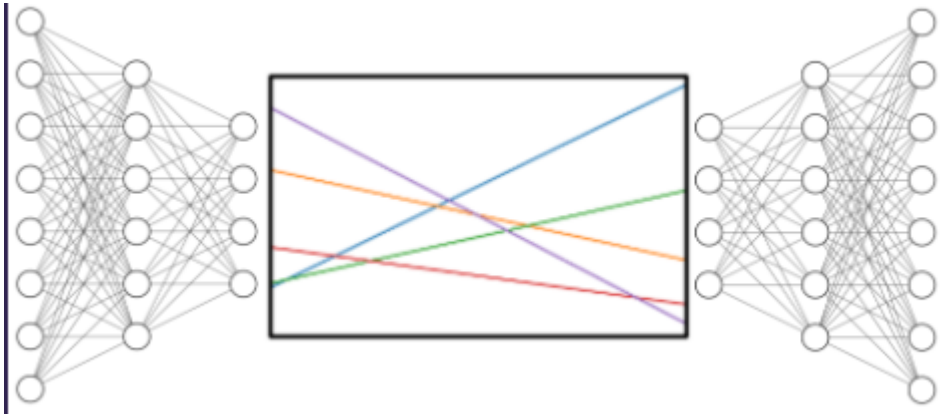
$$f_{\theta*} \simeq poly$$

Done via symbolic regression

NODEs + autoencoders



Sulzer & Buck, 2023

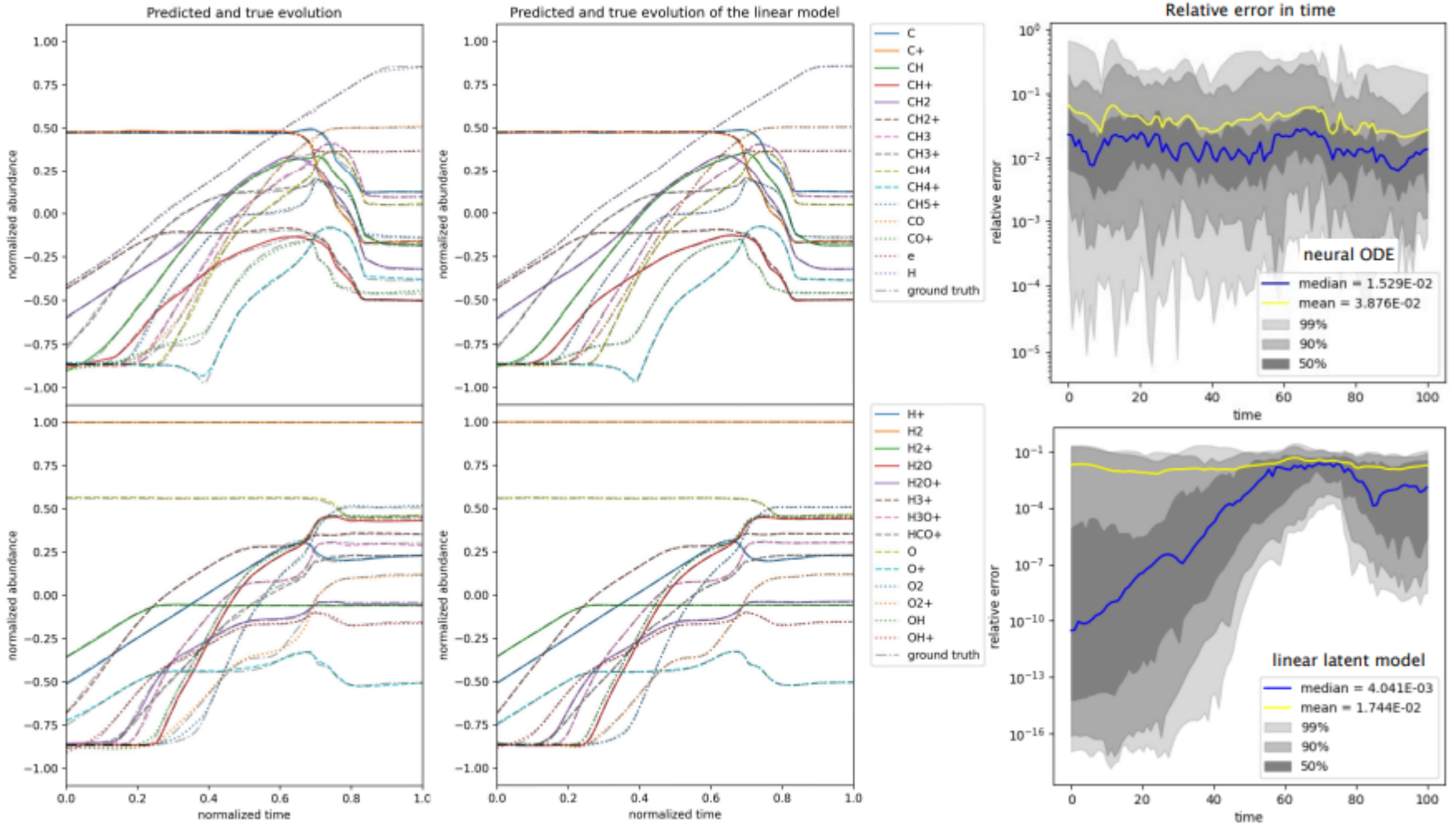


$$f_{\theta*} \simeq poly$$

Done via symbolic regression

See PySR

Some results



Physics Informed Neural Networks (PINNs)

PINN Theory Raissi, 2019

PINN Theory Raissi, 2019

Always remember, NN are nothing but function

PINN Theory Raissi, 2019

Always remember, NN are nothing but function

L. Branca

PINN Theory Raissi, 2019

Always remember, NN are nothing but function

L. Branca

$$\mathcal{D}_x f = g(f, x)$$

PINN Theory Raissi, 2019

Always remember, NN are nothing but function

L. Branca

$$\mathcal{D}_x f = g(f, x) \qquad NN(\theta, x) \simeq f(x)$$

PINN Theory Raissi, 2019

Always remember, NN are nothing but function

L. Branca

$$\mathcal{D}_x f = g(f, x) \qquad NN(\theta, x) \simeq f(x)$$

$$\mathcal{D}_x NN(\theta) = g(NN(\theta), x)$$

PINN Theory Raissi, 2019

Always remember, NN are nothing but function

L. Branca

$$\mathcal{D}_x f = g(f, x) \qquad NN(\theta, x) \simeq f(x)$$

$$\mathcal{D}_x NN(\theta) = g(NN(\theta), x) \quad \mathcal{L}(\theta) = ||\mathcal{D}_x NN(\theta) - g(NN(\theta), x)||_x$$

PINN Theory Raissi, 2019

Always remember, NN are nothing but function

L. Branca

$$\mathcal{D}_x f = g(f, x) \qquad NN(\theta, x) \simeq f(x)$$

$$\mathcal{D}_x NN(\theta) = g(NN(\theta), x) \quad \mathcal{L}(\theta) = ||\mathcal{D}_x NN(\theta) - g(NN(\theta), x)||_x$$

$$NN(\theta^*) = argmin_\theta \mathcal{L}(\theta)$$

PINN Theory

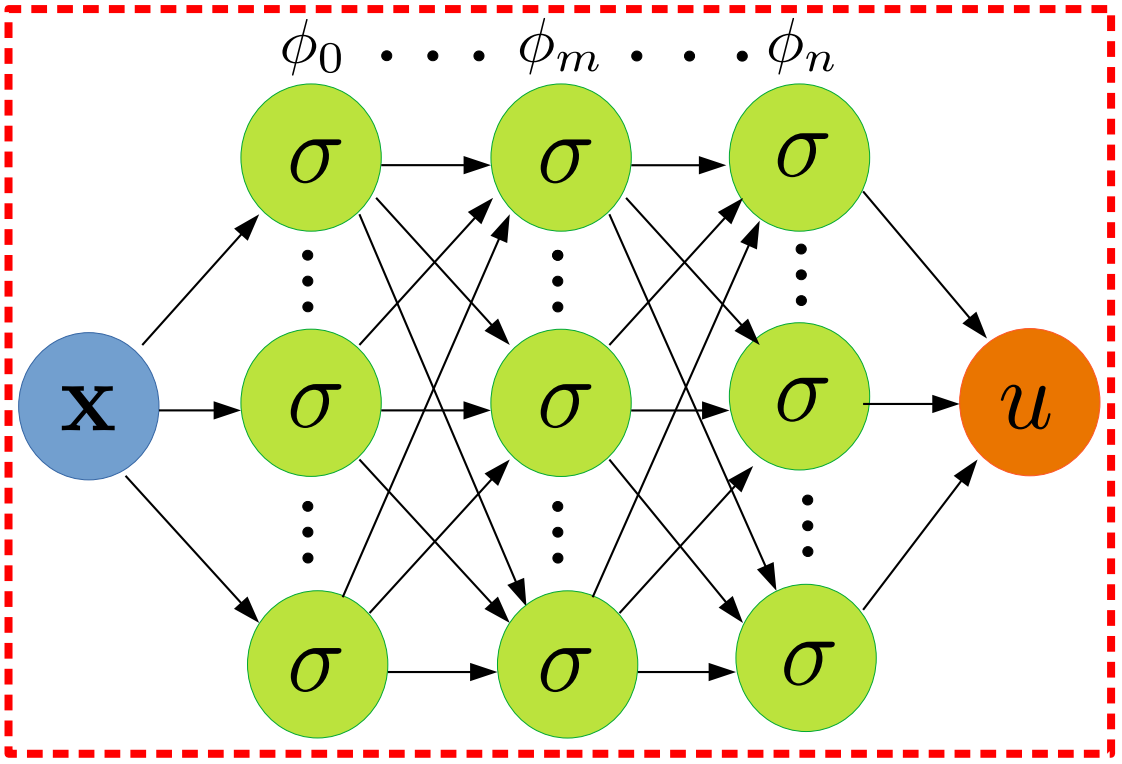
PINN Theory

$$\mathcal{D}(u) = f$$

$$\mathcal{B}(u) = g$$

PINN Theory

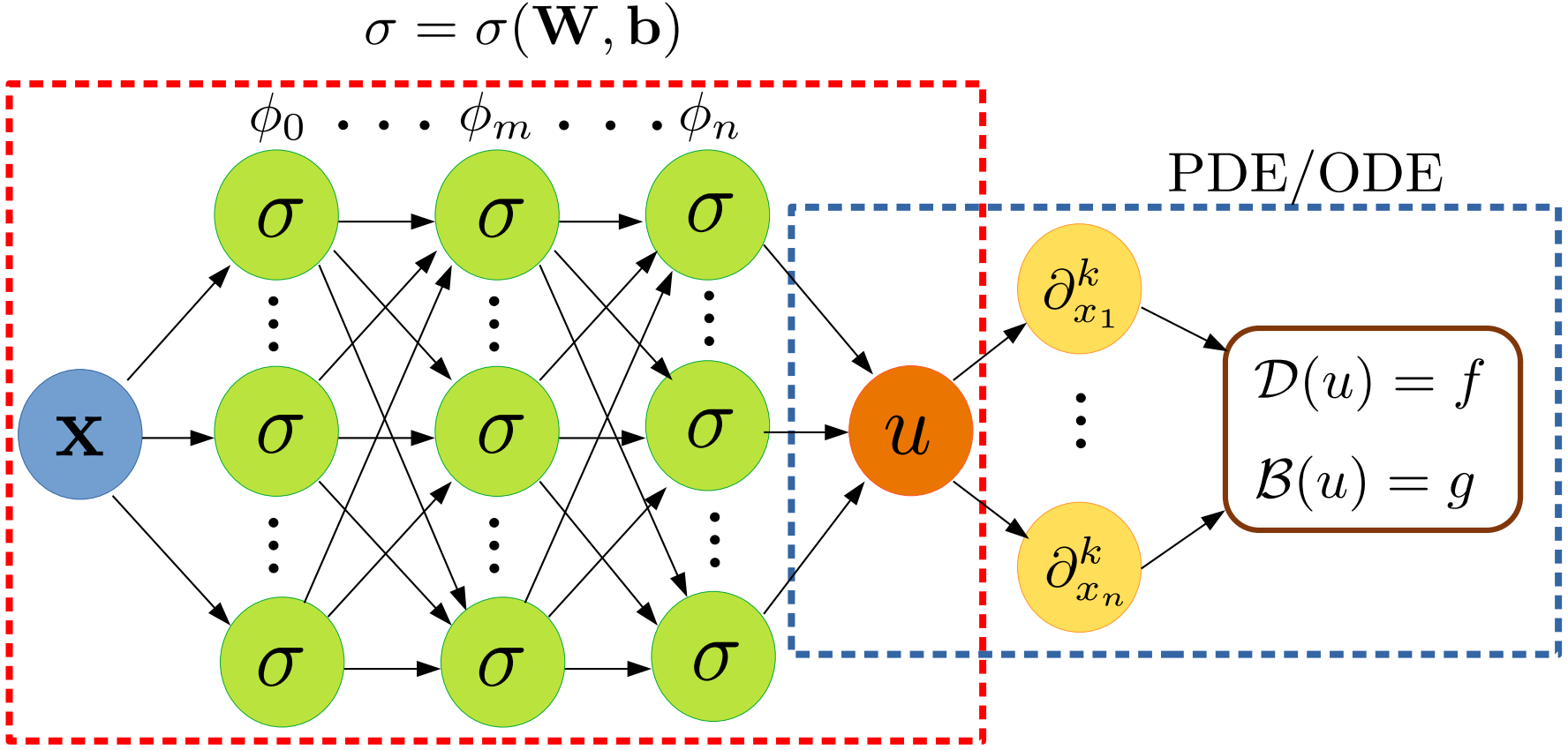
$$\sigma = \sigma(\mathbf{W}, \mathbf{b})$$



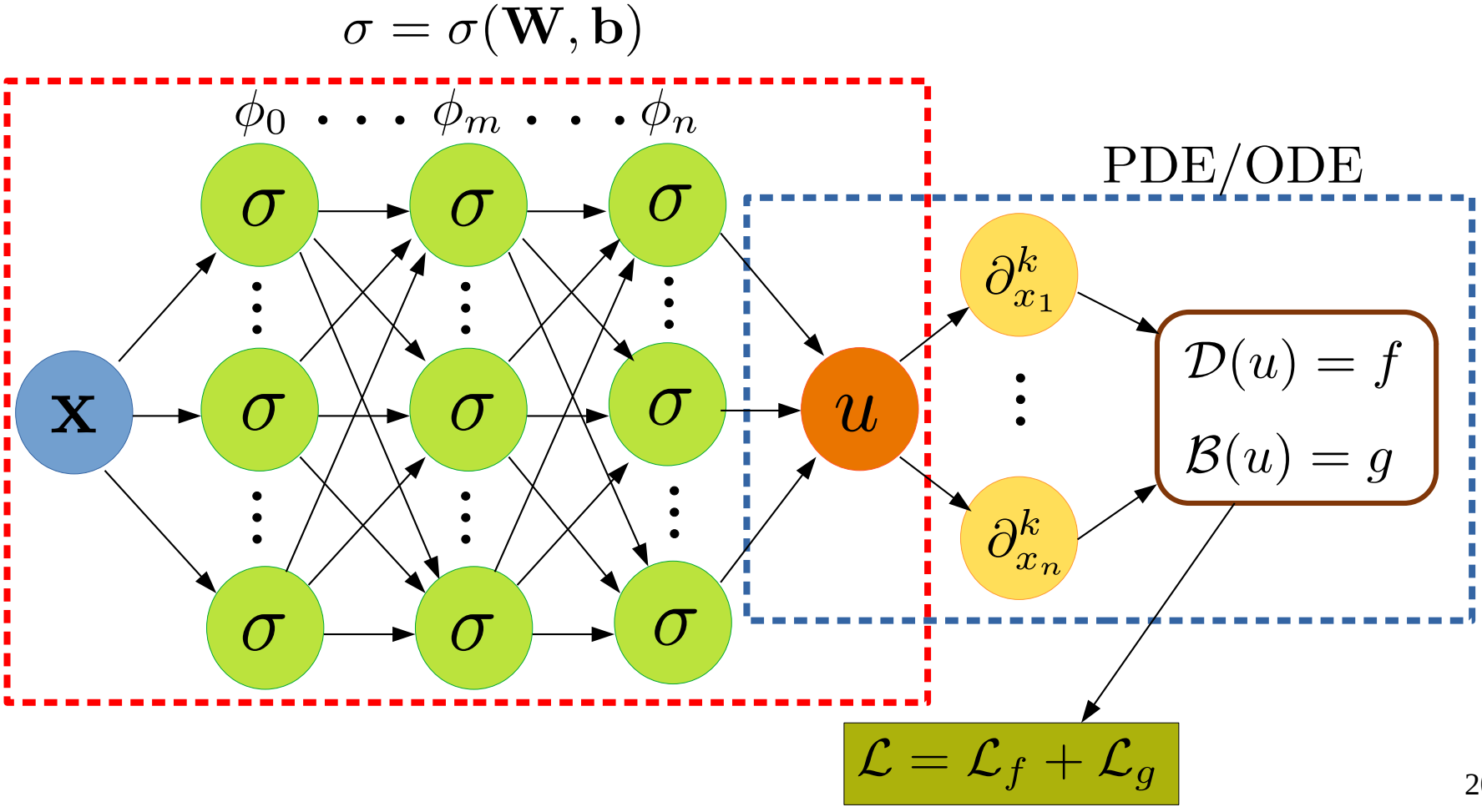
$$\mathcal{D}(u) = f$$

$$\mathcal{B}(u) = g$$

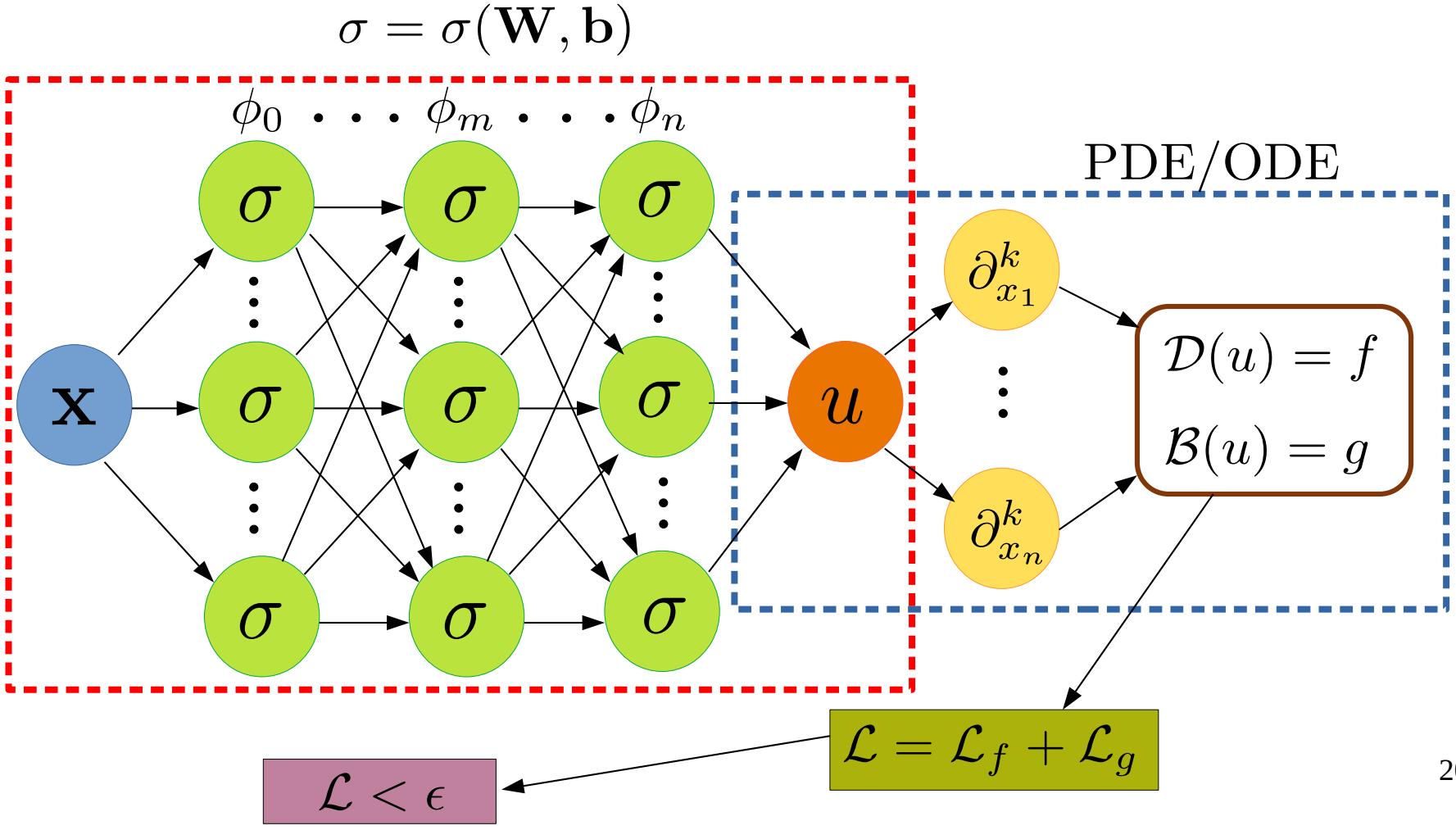
PINN Theory


$\sigma = \sigma(\mathbf{W}, \mathbf{b})$
$\phi_0 \quad \cdots \quad \phi_m \quad \cdots \quad \phi_n$
$\mathbf{x}$
σ
u
PDE/ODE
$\partial^k_{x_1}$
$\partial^k_{x_n}$
$\mathcal{D}(u) = f$
$\mathcal{B}(u) = g$

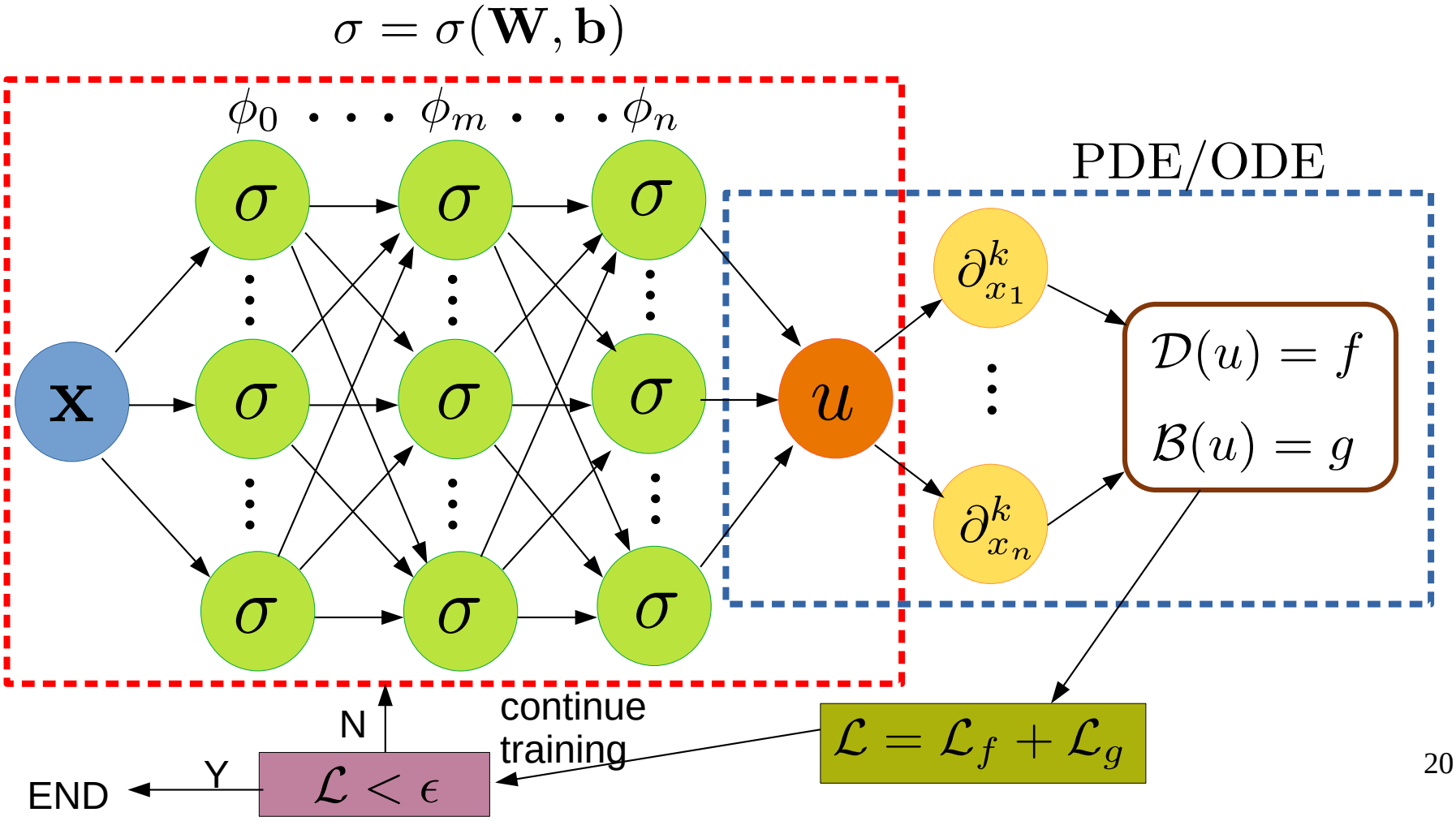
PINN Theory


$\sigma = \sigma(\mathbf{W}, \mathbf{b})$
$\phi_0 \quad \cdots \quad \phi_m \quad \cdots \quad \phi_n$
$\mathbf{x}$
σ
u
PDE/ODE
$\partial^k_{x_1}$
$\partial^k_{x_n}$
$\mathcal{D}(u) = f$
$\mathcal{B}(u) = g$
$\mathcal{L} = \mathcal{L}_f + \mathcal{L}_g$

PINN Theory


$\sigma = \sigma(\mathbf{W}, \mathbf{b})$
$\phi_0 \quad \cdots \quad \phi_m \quad \cdots \quad \phi_n$
$\mathbf{x}$
σ
u
PDE/ODE
$\partial^k_{x_1}$
$\partial^k_{x_n}$
$\mathcal{D}(u) = f$
$\mathcal{B}(u) = g$
$\mathcal{L} = \mathcal{L}_f + \mathcal{L}_g$
$\mathcal{L} < \epsilon$

PINN Theory


$\sigma = \sigma(\mathbf{W}, \mathbf{b})$
$\phi_0 \quad \cdots \quad \phi_m \quad \cdots \quad \phi_n$
$\mathbf{x}$
σ
u
PDE/ODE
$\partial^k_{x_1}$
$\partial^k_{x_n}$
$\mathcal{D}(u) = f$
$\mathcal{B}(u) = g$
$\mathcal{L} = \mathcal{L}_f + \mathcal{L}_g$
$\mathcal{L} < \epsilon$
N
Y
continue training
END

Specific case of study

- **2 different chemical network: *atomic* (*molecular*)**
  - 7(9) species: e^- $\mathrm{H}^-, \mathrm{H}, \mathrm{H}^+, \mathrm{He}, \mathrm{He}^+, \mathrm{He}^{++}, (\mathrm{H}_2, \mathrm{H}_2^+)$
  - 25(46) chemical reactions, photo-heating, CR-heating, eso/endo-thermic heating/cooling, metal cooling, (H_2 cooling)
  - HM ionizing photon flux
- **S*ingle* and *general* Initial Conditions**

Single

$$\mathrm{T}_{in} = 10^3 \mathrm{K}$$

$$\mathrm{n}_{in} = 10^2 \mathrm{cm}^{-3}$$

n_k/n fixed

General

$$\mathrm{T}_{in} \in [20, 10^6] \mathrm{K}$$

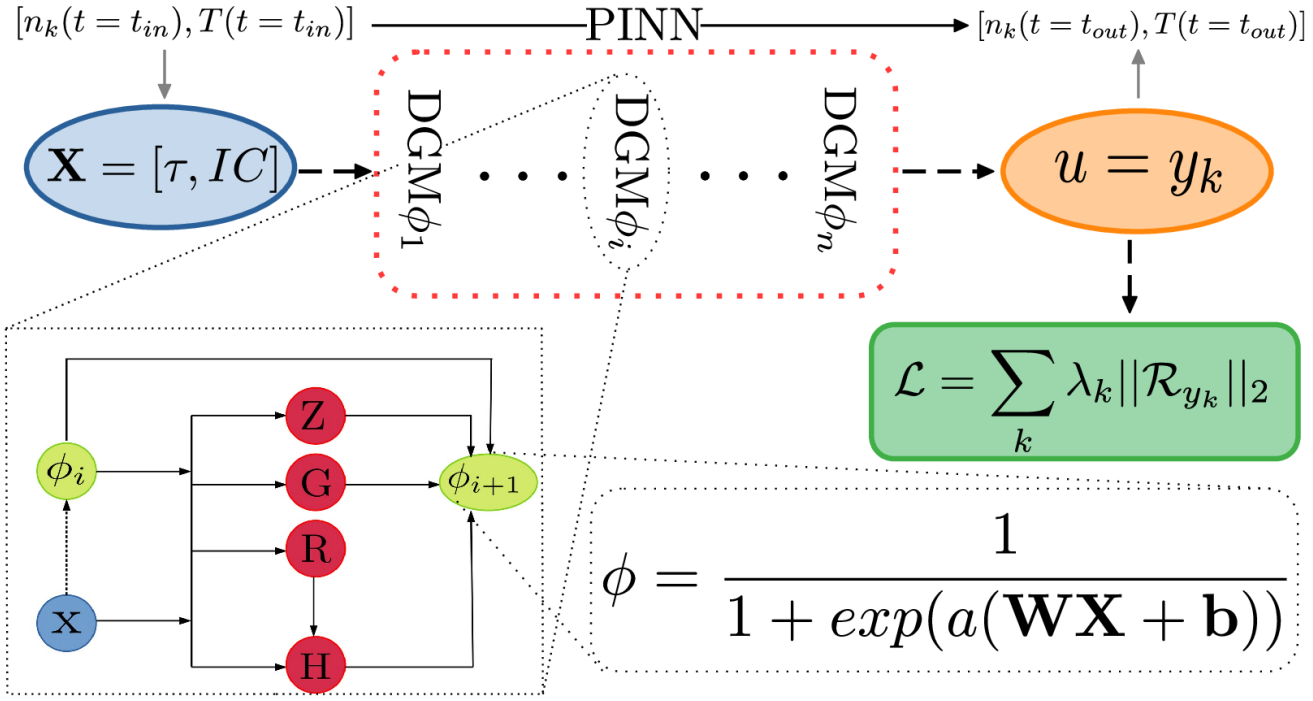
$$\mathrm{n}_{in} \in [10^{-2}, 10^3] \mathrm{cm}^{-3}$$

$$\mathrm{n}_k/\mathrm{n} \in [10^{-6}, 0.8]$$

PINN architecture and training

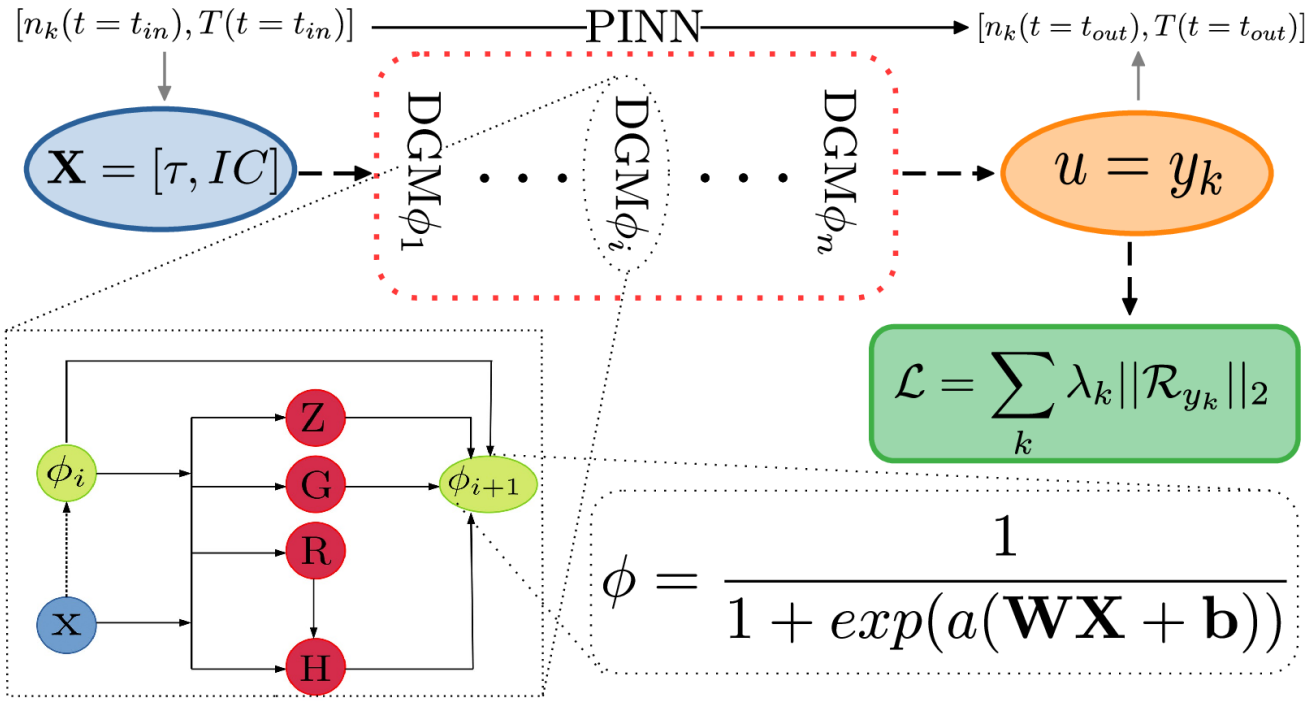
Branca & Pallottini 2023

PINN architecture and training

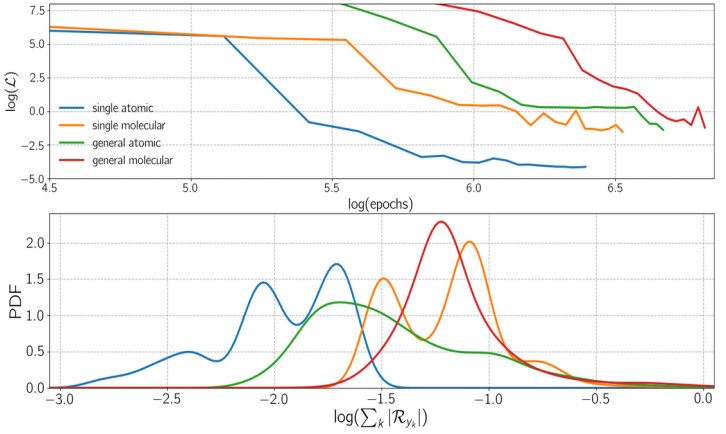


Branca & Pallottini 2023

PINN architecture and training



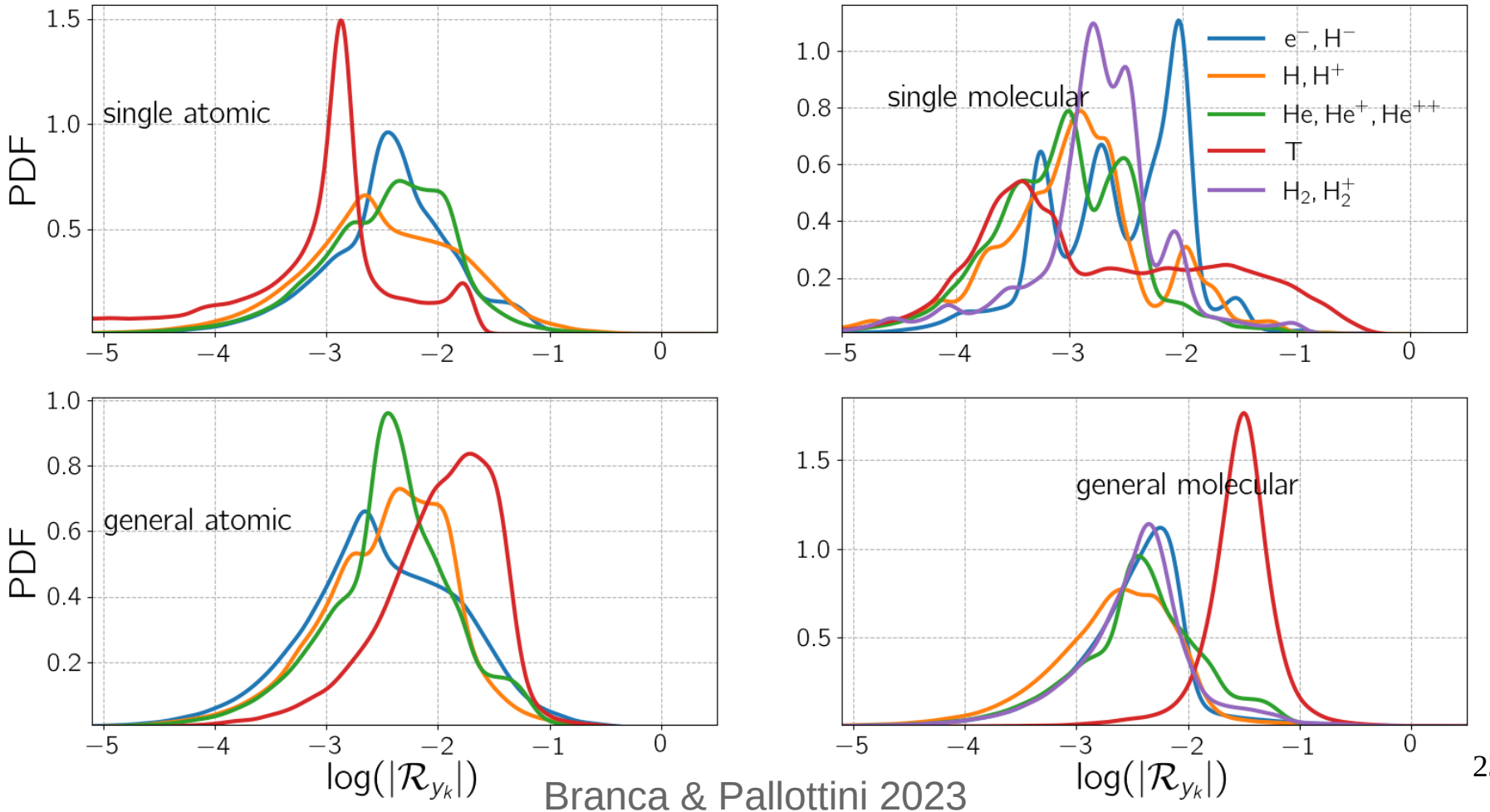
Branca & Pallottini 2023



Training time (GPUhrs):

- ~120
- ~820
- ~1224
- ~1920

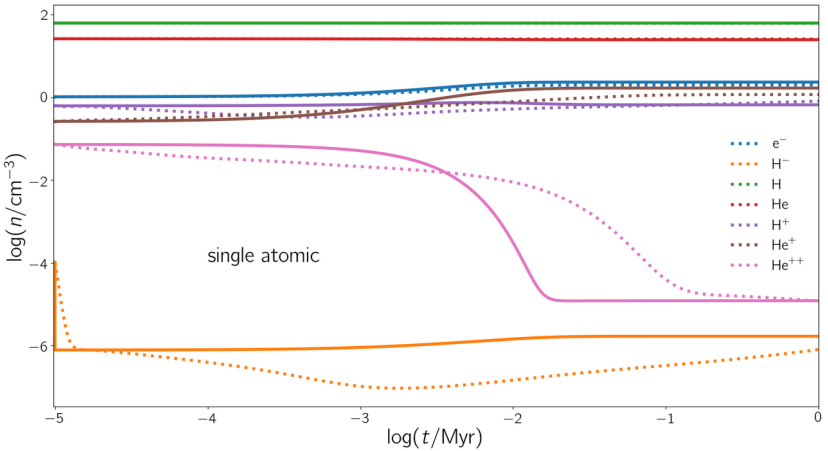
Results: residuals

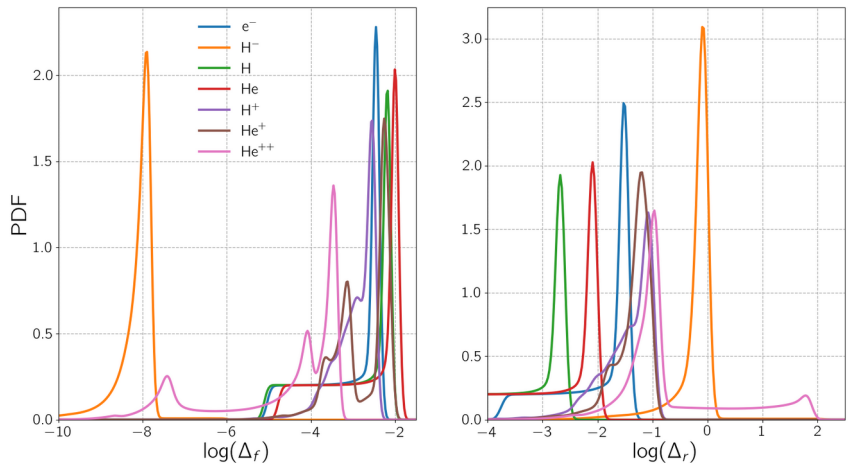


Branca & Pallottini 2023

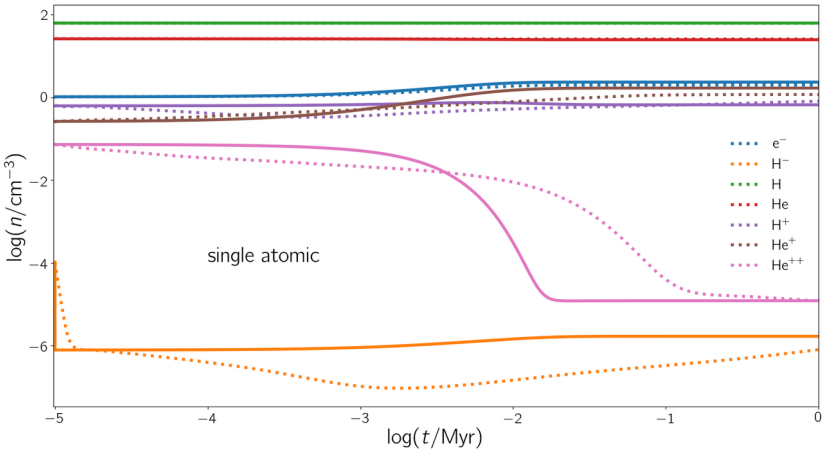
Results: comparison

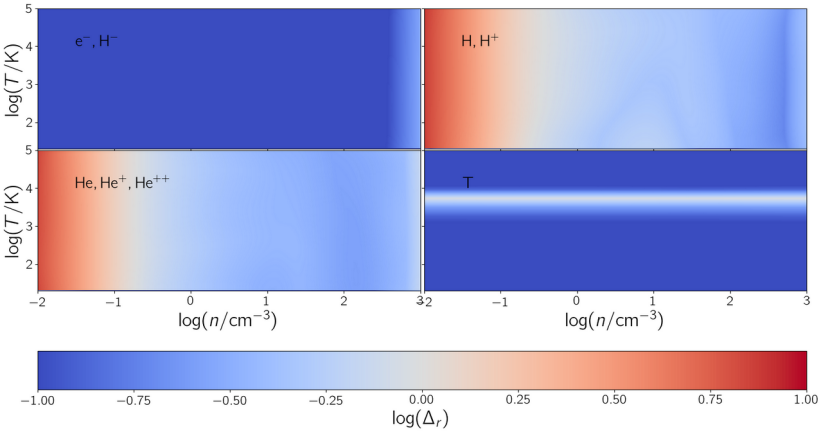
Results: comparison

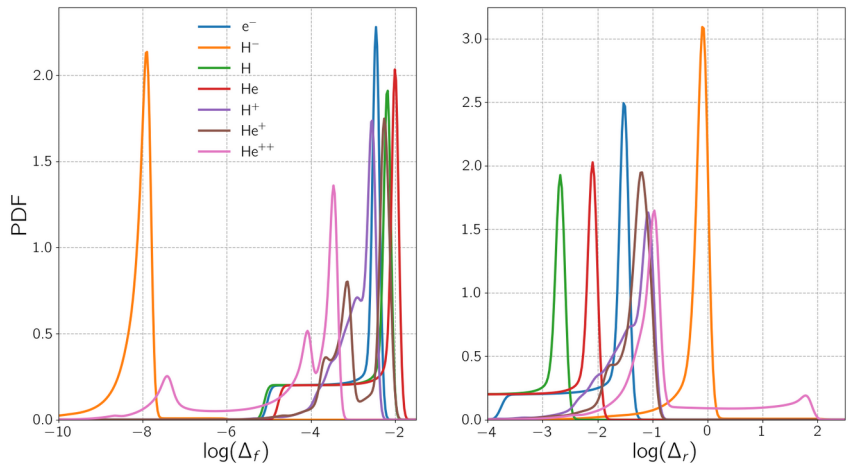


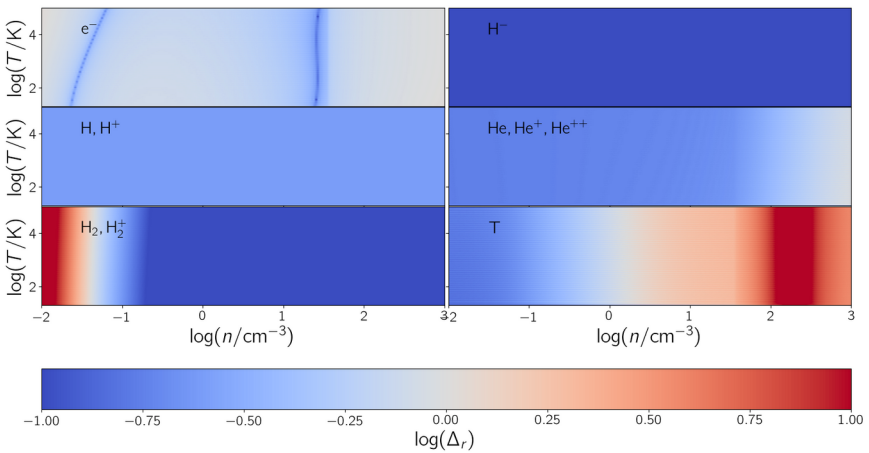


Results: comparison



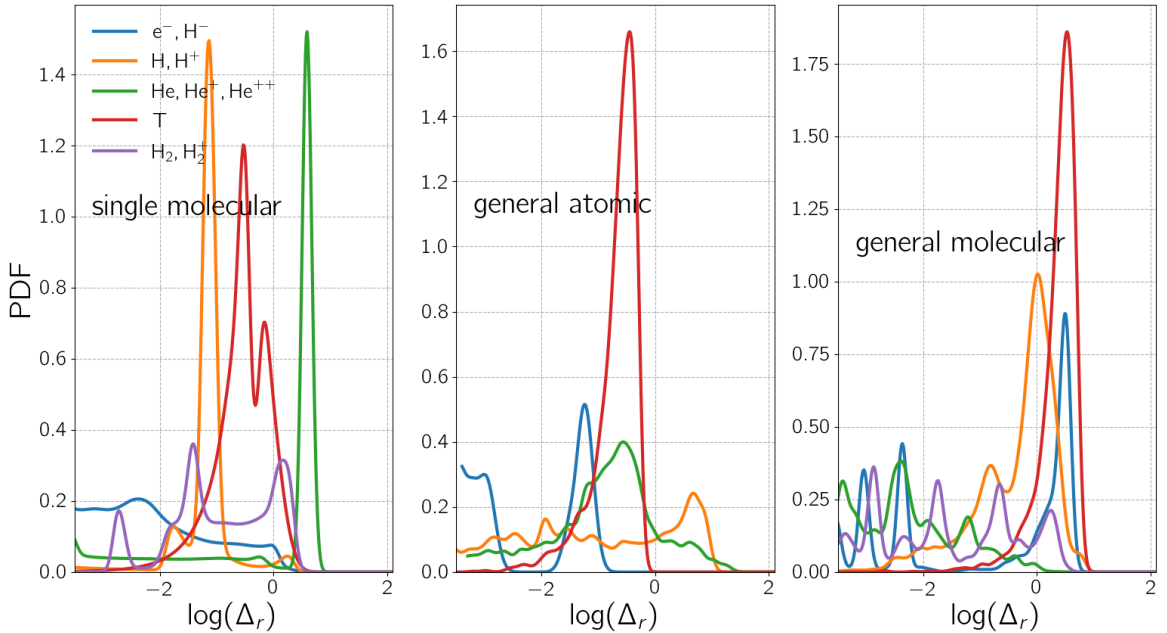






Results: comparison

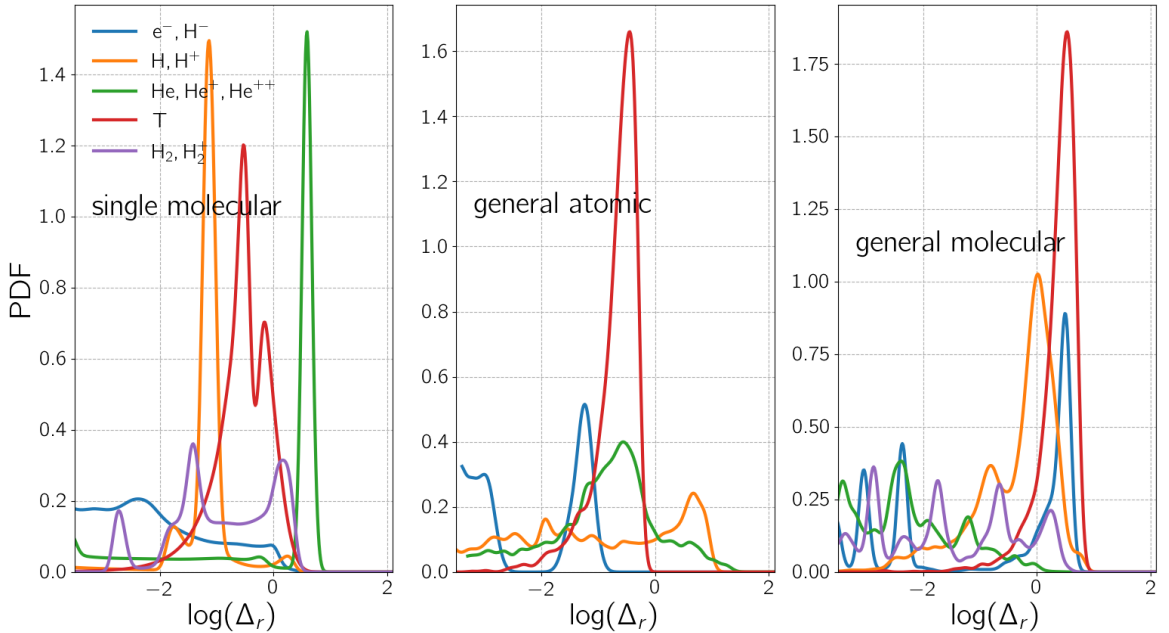
Results: comparison



Quantile	atomic single	molecular single	atomic general	molecular general
50%	-1.51	-2.91	-3.87	-1.5
75%	-1.05	-0.84	-0.89	-0.15
90%	-0.77	0.16	-0.15	0.38

Branca & Pallottini 2023

Results: comparison

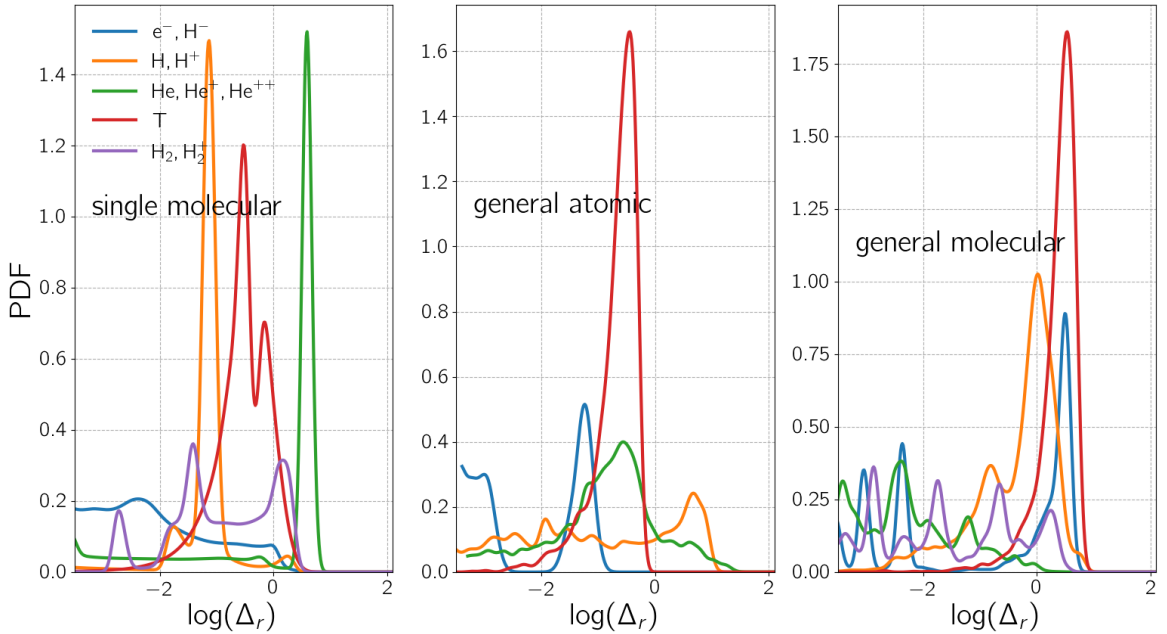


Quantile	atomic single	molecular single	atomic general	molecular general
50%	-1.51	-2.91	-3.87	-1.5
75%	-1.05	-0.84	-0.89	-0.15
90%	-0.77	0.16	-0.15	0.38

Accuracy: ~10%

Branca & Pallottini 2023

Results: comparison



Quantile	atomic single	molecular single	atomic general	molecular general
50%	-1.51	-2.91	-3.87	-1.5
75%	-1.05	-0.84	-0.89	-0.15
90%	-0.77	0.16	-0.15	0.38

Accuracy: ~10%
Speed up: ~90X

Branca & Pallottini 2023

PINNs summary

PINNs summary

PINNs summary

- No need of training data

PINNs summary

- No need of training data
- Training expansive (~1000 GPUhr)

PINNs summary

- No need of training data
- Training expansive (~1000 GPUhr)
- Low accuracy (~10%), not competitive

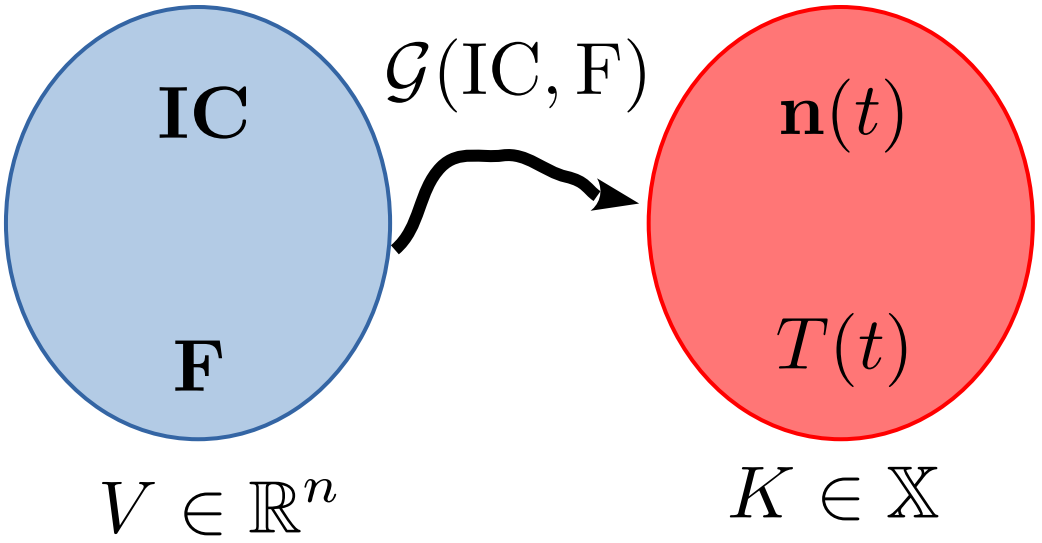
PINNs summary

- No need of training data
- Training expansive (~1000 GPUhr)
- Low accuracy (~10%), not competitive
- More sophisticated models needed

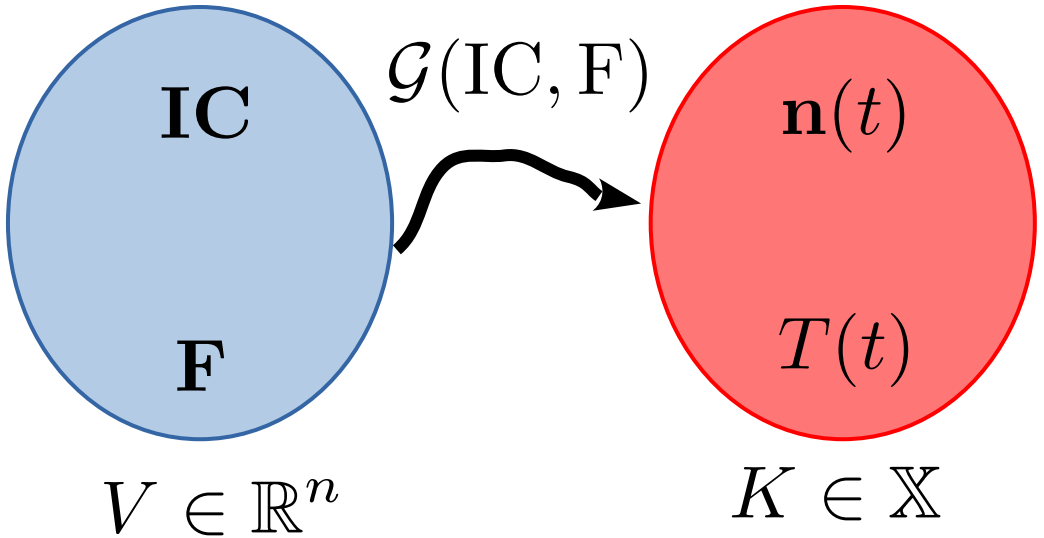
Deep Operator Network (DeepONet) as emulators for ISM chemistry

From functions to operators

From functions to operators



From functions to operators



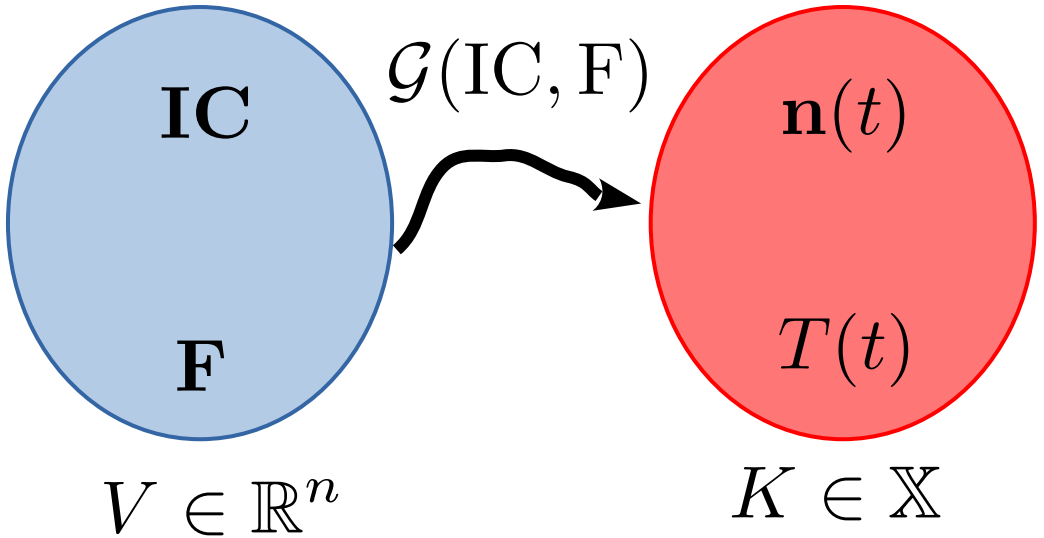
Theorem 1 (Universal Approximation Theorem for Operator). *Suppose that σ is a continuous non-polynomial function, X is a Banach space, $K_1 \subset X$, $K_2 \subset \mathbb{R}^d$ are two compact sets in X and $\mathbb{R}^d$, respectively, V is a compact set in $C(K_1)$, G is a nonlinear continuous operator, which maps V into $C(K_2)$. Then for any $\epsilon > 0$, there are positive integers n, p and m, constants c_i^k, ξ_{ij}^k, θ_i^k, $\zeta_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$, $x_j \in K_1$, $i = 1, \ldots, n$, $k = 1, \ldots, p$ and $j = 1, \ldots, m$, such that*

$$\left| G(u)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right)}_{\text{branch}} \underbrace{\sigma(w_k \cdot y + \zeta_k)}_{\text{trunk}} \right| < \epsilon \tag{1}$$

holds for all $u \in V$ and $y \in K_2$. Here, $C(K)$ is the Banach space of all continuous functions defined on K with norm $\| f \|_{C(K)} = \max_{x \in K} |f(x)|$.

Chen, 1995

From functions to operators



Theorem 1 (Universal Approximation Theorem for Operator). *Suppose that σ is a continuous non-polynomial function, X is a Banach space, $K_1 \subset X$, $K_2 \subset \mathbb{R}^d$ are two compact sets in X and $\mathbb{R}^d$, respectively, V is a compact set in $C(K_1)$, G is a nonlinear continuous operator, which maps V into $C(K_2)$. Then for any $\epsilon > 0$, there are positive integers n, p and m, constants c_i^k, ξ_{ij}^k, θ_i^k, $\zeta_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$, $x_j \in K_1$, $i = 1, \ldots, n$, $k = 1, \ldots, p$ and $j = 1, \ldots, m$, such that*

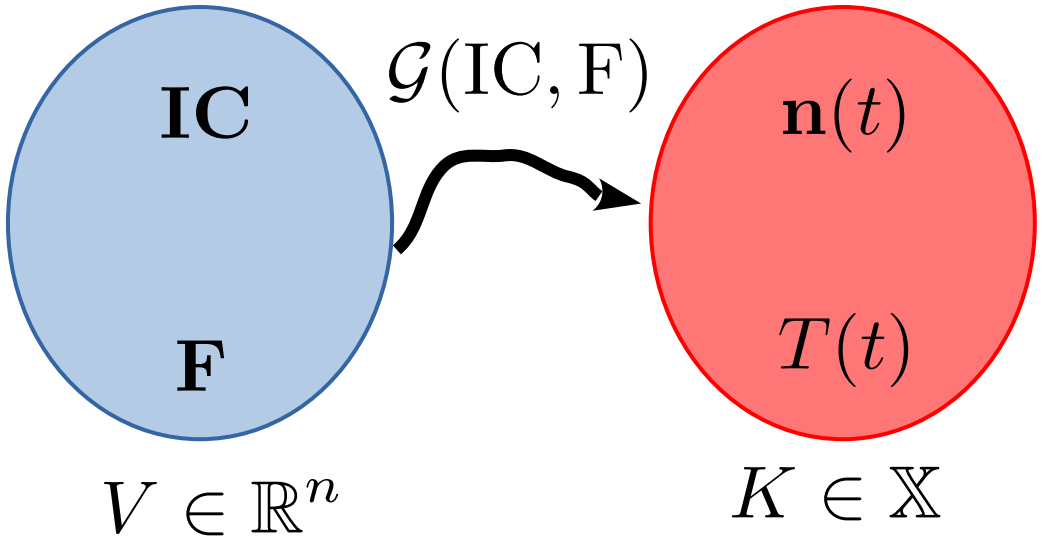
$$\left| G(u)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right)}_{\text{branch}} \underbrace{\sigma(w_k \cdot y + \zeta_k)}_{\text{trunk}} \right| < \epsilon \tag{1}$$

holds for all $u \in V$ and $y \in K_2$. Here, $C(K)$ is the Banach space of all continuous functions defined on K with norm $\| f \|_{C(K)} = \max_{x \in K} |f(x)|$.

Chen, 1995

$$||\mathcal{G}(u)(y) - f(u) \otimes g(y)|| < \epsilon$$

From functions to operators



Theorem 1 (Universal Approximation Theorem for Operator). *Suppose that σ is a continuous non-polynomial function, X is a Banach space, $K_1 \subset X$, $K_2 \subset \mathbb{R}^d$ are two compact sets in X and $\mathbb{R}^d$, respectively, V is a compact set in $C(K_1)$, G is a nonlinear continuous operator, which maps V into $C(K_2)$. Then for any $\epsilon > 0$, there are positive integers n, p and m, constants c_i^k, ξ_{ij}^k, θ_i^k, $\zeta_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$, $x_j \in K_1$, $i = 1, \ldots, n$, $k = 1, \ldots, p$ and $j = 1, \ldots, m$, such that*

$$\left| G(u)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right)}_{\text{branch}} \underbrace{\sigma(w_k \cdot y + \zeta_k)}_{\text{trunk}} \right| < \epsilon \tag{1}$$

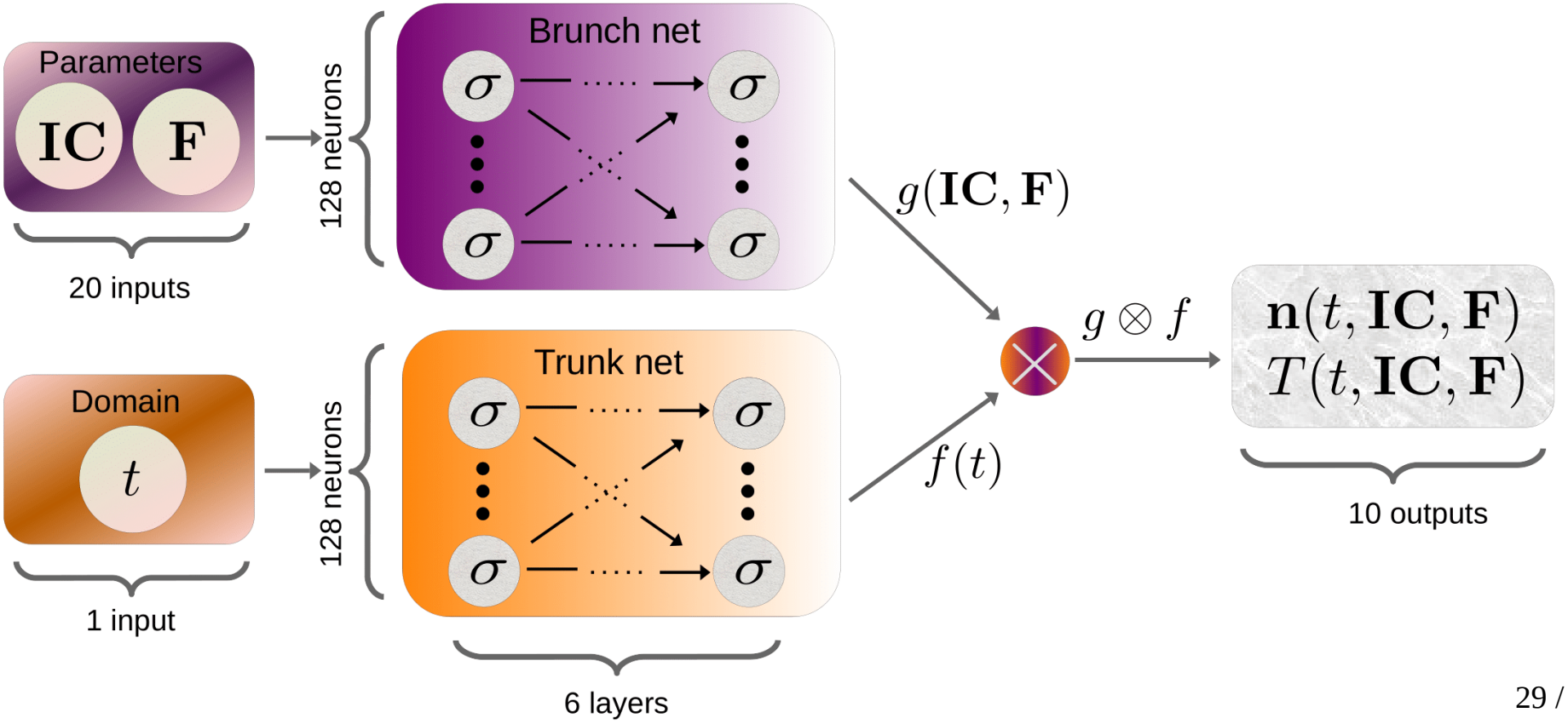
holds for all $u \in V$ and $y \in K_2$. Here, $C(K)$ is the Banach space of all continuous functions defined on K with norm $\| f \|_{C(K)} = \max_{x \in K} |f(x)|$.

Chen, 1995

$$||\mathcal{G}(u)(y) - f(u) \otimes g(y)|| < \epsilon$$

Operator Net1 Net2

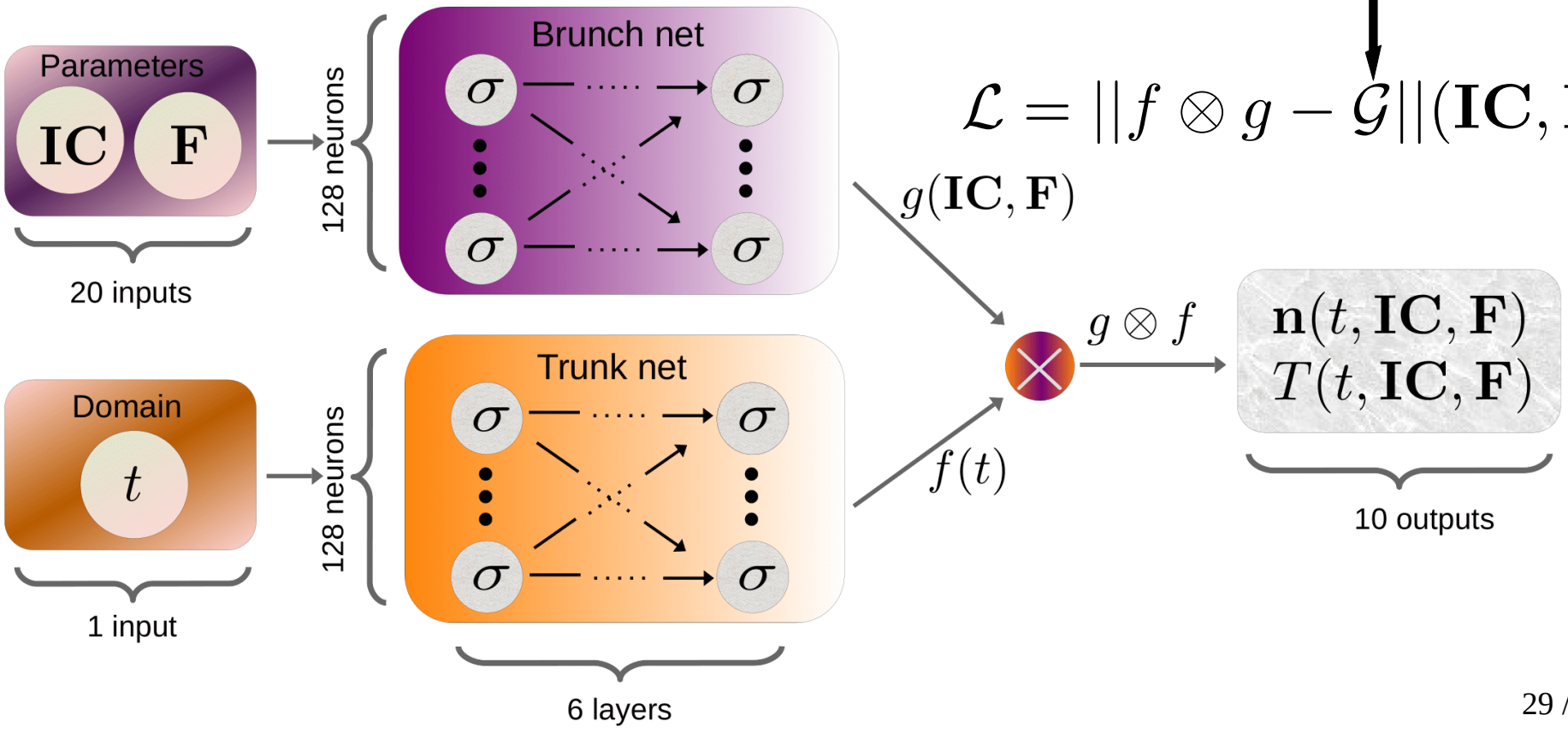
Deep Operator Network



Deep Operator Network

KROMEPACKAGE

Grassi et al. 2014



$$\mathcal{L} = ||f \otimes g - \mathcal{G}||(\mathbf{IC}, \mathbf{F}, t)$$

Chemistry and Data

$$\mathrm{e}^-\ \mathrm{H}^-, \mathrm{H}, \mathrm{H}^+, \mathrm{He}, \mathrm{He}^+, \mathrm{He}^{++}, \mathrm{H}_2, \mathrm{H}_2^+$$

52 photo-chemical reactions, photo-heating, CR-heating, eso/endo-thermic heating/cooling, metal cooling, H_2 cooling

10 photo-bins

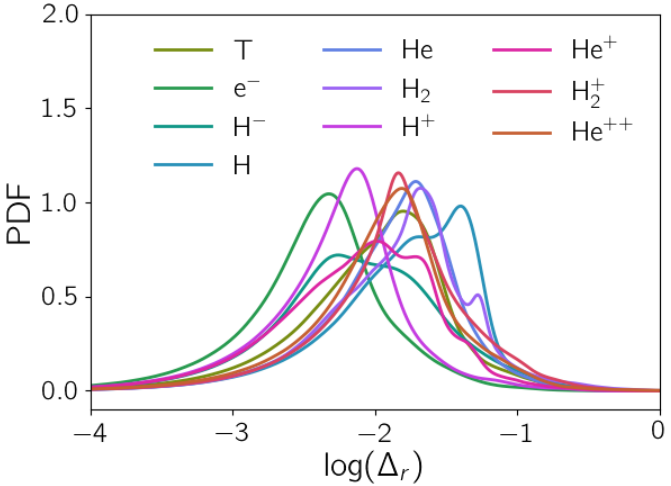
quantity	variable	bins	min	max
gas density	$\log(n/\mathrm{cm}^{-3})$	64	-2	3.5
abundances	$\log(n_i/n)$	512	-6	0
temperature	$\log(T/\mathrm{K})$	64	log(20)	5.5
radiation	$\log(F_i/\mathrm{eV\,cm^{-2}\,s^{-1}\,Hz})$	64	-15	-5
time	t/kyr	16	0	1

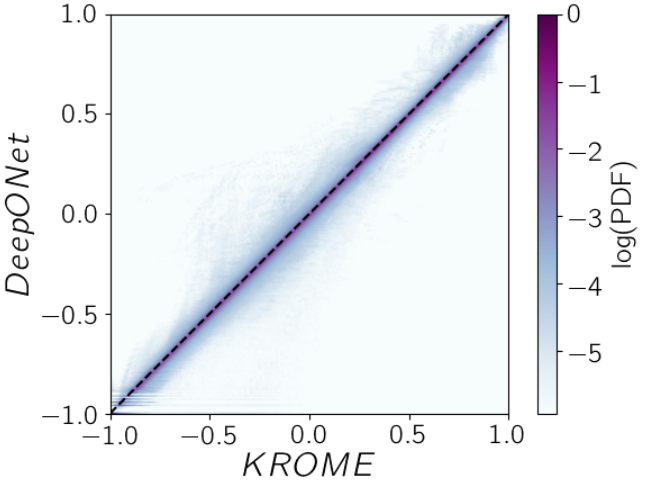
$$|\log(F_i/F_{i\pm1})| \leq 0.15$$

Sampled via constrained LHS

DeepONet results

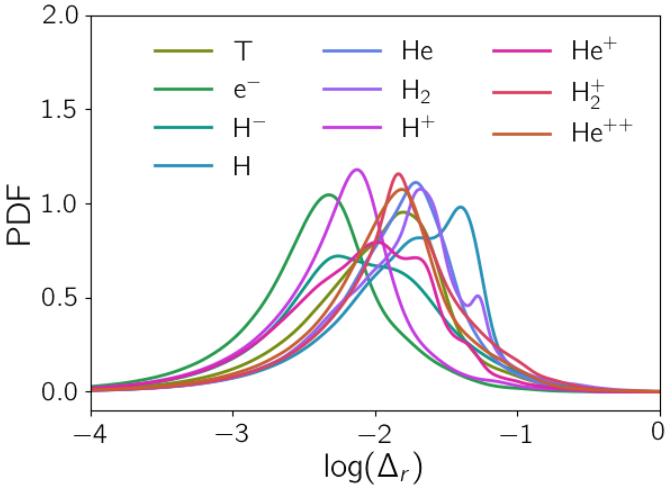
DeepONet results

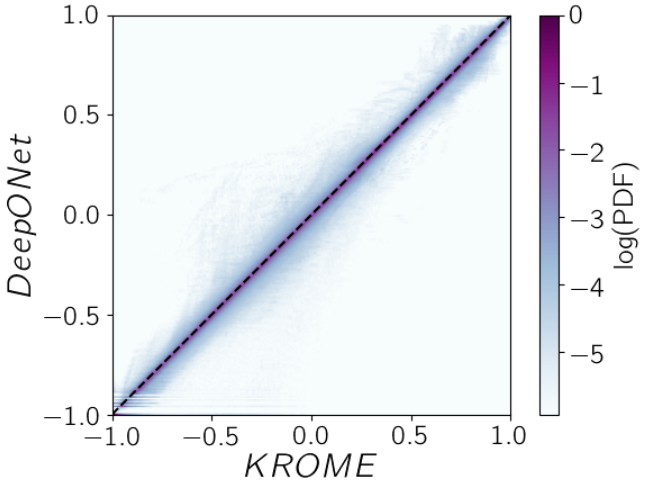




input	MRE	50%	75%	90%
T	0.0179	0.0114	0.0211	0.0353
e^-	0.0074	0.0044	0.0078	0.0151
H^-	0.0176	0.0076	0.0183	0.0398
H	0.0269	0.0190	0.0375	0.0547
He	0.0213	0.0150	0.0258	0.0422
H_2	0.0274	0.0167	0.0292	0.0548
H^+	0.0099	0.0060	0.0101	0.0178
He^+	0.0148	0.0082	0.0177	0.0314
H_2^+	0.0255	0.0146	0.0267	0.0552
He^{++}	0.0234	0.0125	0.0220	0.0436

DeepONet results

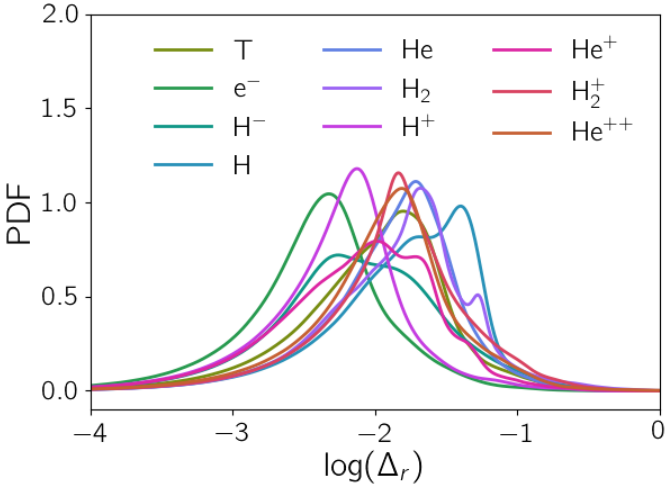


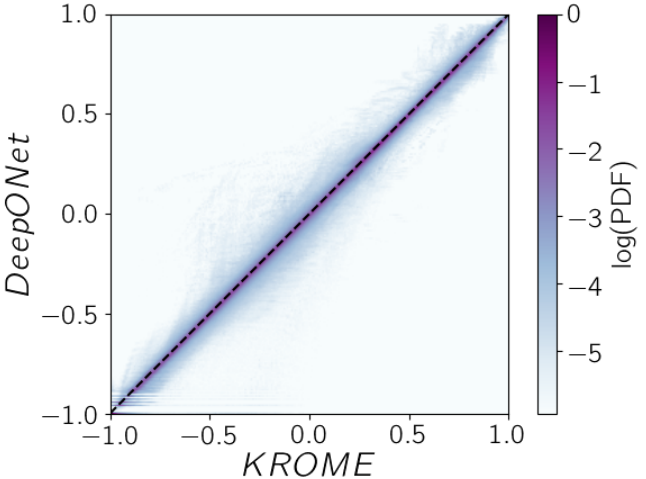


input	MRE	50%	75%	90%
T	0.0179	0.0114	0.0211	0.0353
e^-	0.0074	0.0044	0.0078	0.0151
H^-	0.0176	0.0076	0.0183	0.0398
H	0.0269	0.0190	0.0375	0.0547
He	0.0213	0.0150	0.0258	0.0422
H_2	0.0274	0.0167	0.0292	0.0548
H^+	0.0099	0.0060	0.0101	0.0178
He^+	0.0148	0.0082	0.0177	0.0314
H_2^+	0.0255	0.0146	0.0267	0.0552
He^{++}	0.0234	0.0125	0.0220	0.0436

- Training: ~40 GPUhrs

DeepONet results

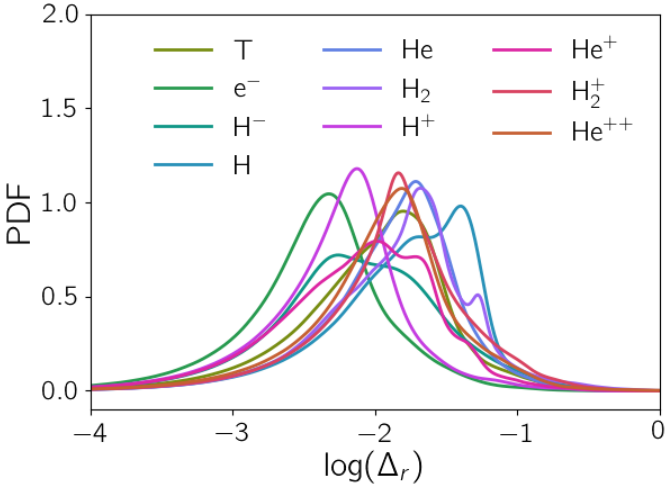


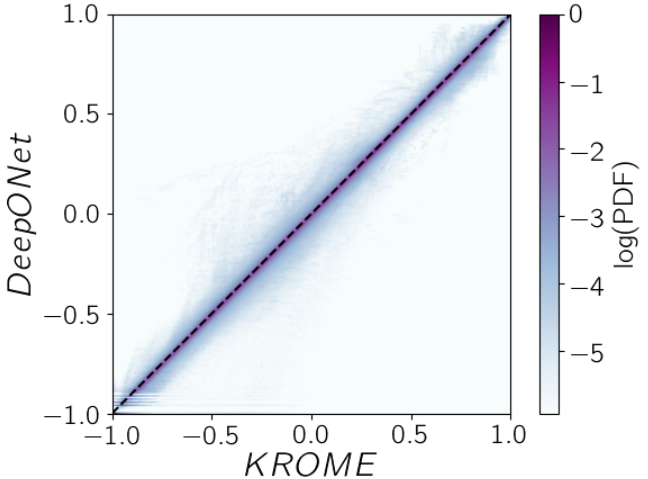


input	MRE	50%	75%	90%
T	0.0179	0.0114	0.0211	0.0353
e^-	0.0074	0.0044	0.0078	0.0151
H^-	0.0176	0.0076	0.0183	0.0398
H	0.0269	0.0190	0.0375	0.0547
He	0.0213	0.0150	0.0258	0.0422
H_2	0.0274	0.0167	0.0292	0.0548
H^+	0.0099	0.0060	0.0101	0.0178
He^+	0.0148	0.0082	0.0177	0.0314
H_2^+	0.0255	0.0146	0.0267	0.0552
He^{++}	0.0234	0.0125	0.0220	0.0436

- Training: ~40 GPUhrs
- Speed-up: 128X

DeepONet results

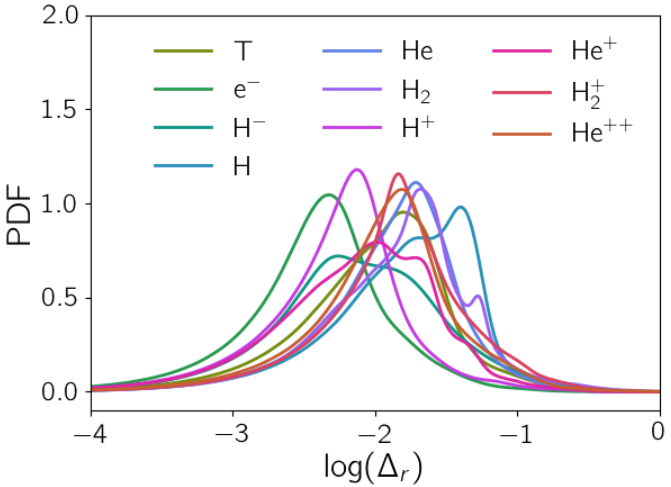


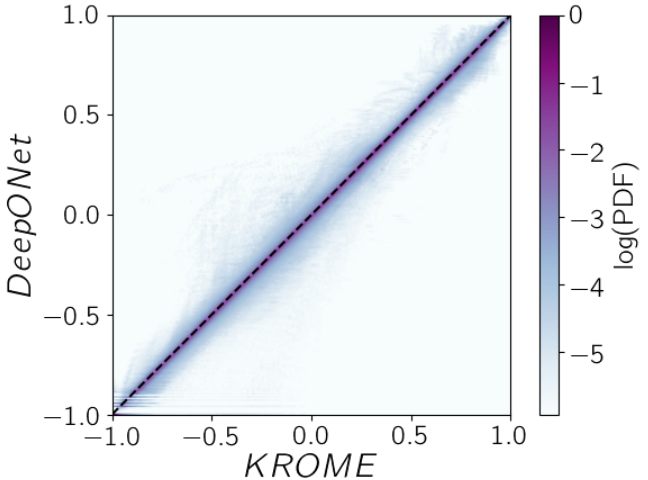


input	MRE	50%	75%	90%
T	0.0179	0.0114	0.0211	0.0353
e^-	0.0074	0.0044	0.0078	0.0151
H^-	0.0176	0.0076	0.0183	0.0398
H	0.0269	0.0190	0.0375	0.0547
He	0.0213	0.0150	0.0258	0.0422
H_2	0.0274	0.0167	0.0292	0.0548
H^+	0.0099	0.0060	0.0101	0.0178
He^+	0.0148	0.0082	0.0177	0.0314
H_2^+	0.0255	0.0146	0.0267	0.0552
He^{++}	0.0234	0.0125	0.0220	0.0436

- Training: ~40 GPUhrs
- Speed-up: 128X
- Accuracy: ~ 1%

DeepONet results

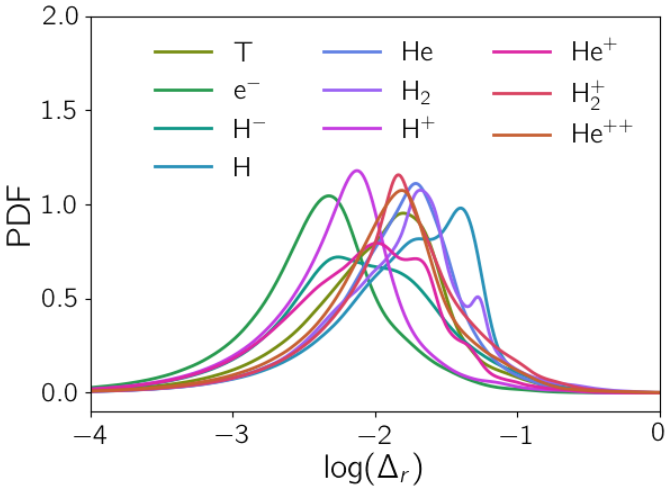


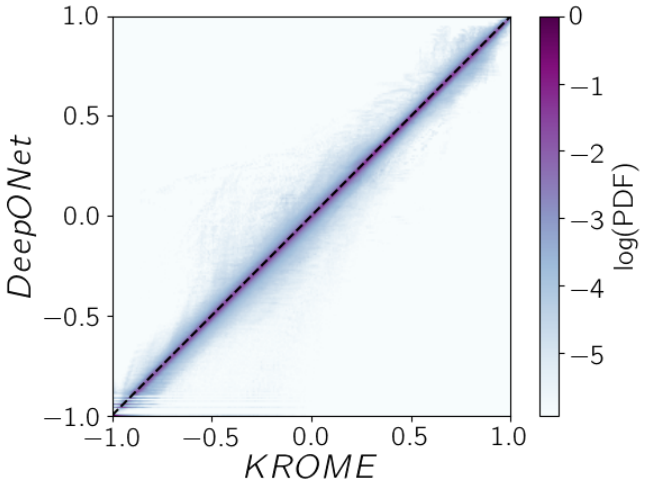


input	MRE	50%	75%	90%
T	0.0179	0.0114	0.0211	0.0353
e^-	0.0074	0.0044	0.0078	0.0151
H^-	0.0176	0.0076	0.0183	0.0398
H	0.0269	0.0190	0.0375	0.0547
He	0.0213	0.0150	0.0258	0.0422
H_2	0.0274	0.0167	0.0292	0.0548
H^+	0.0099	0.0060	0.0101	0.0178
He^+	0.0148	0.0082	0.0177	0.0314
H_2^+	0.0255	0.0146	0.0267	0.0552
He^{++}	0.0234	0.0125	0.0220	0.0436

- Training: ~40 GPUhrs
- Speed-up: 128X
- Accuracy: ~ 1%
- N ouliers: ~ 1/1.000.000

DeepONet results



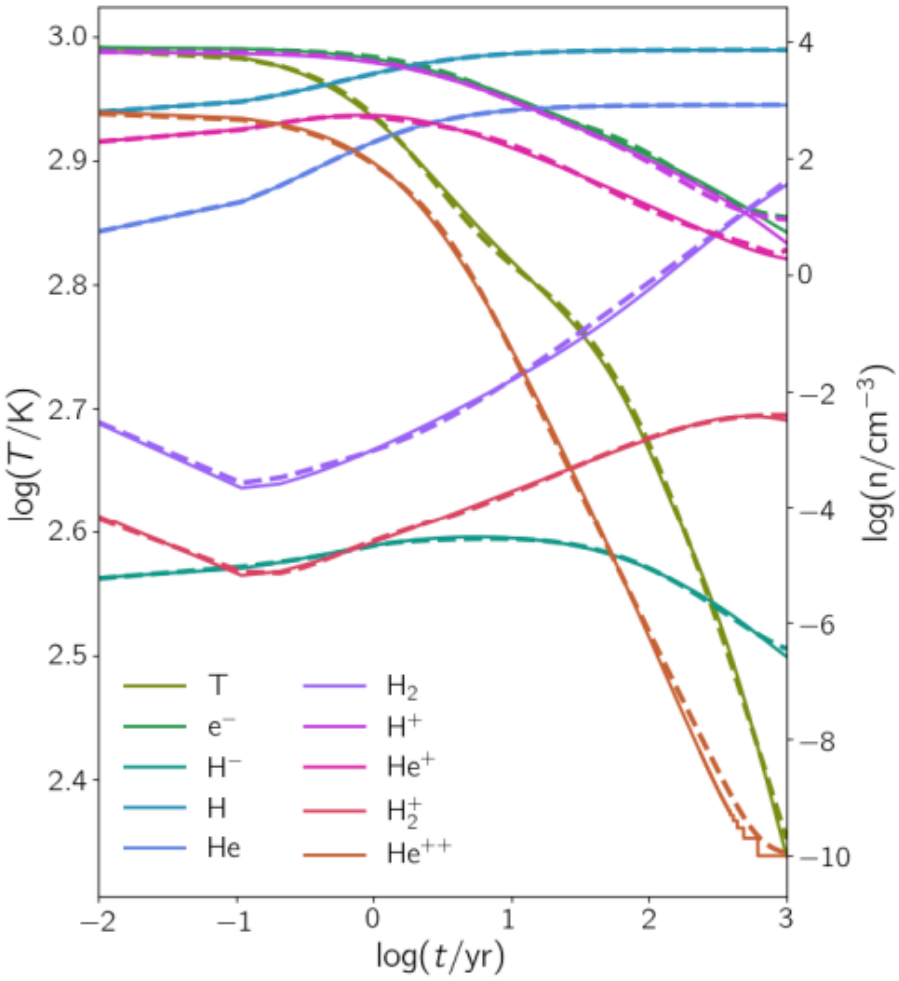


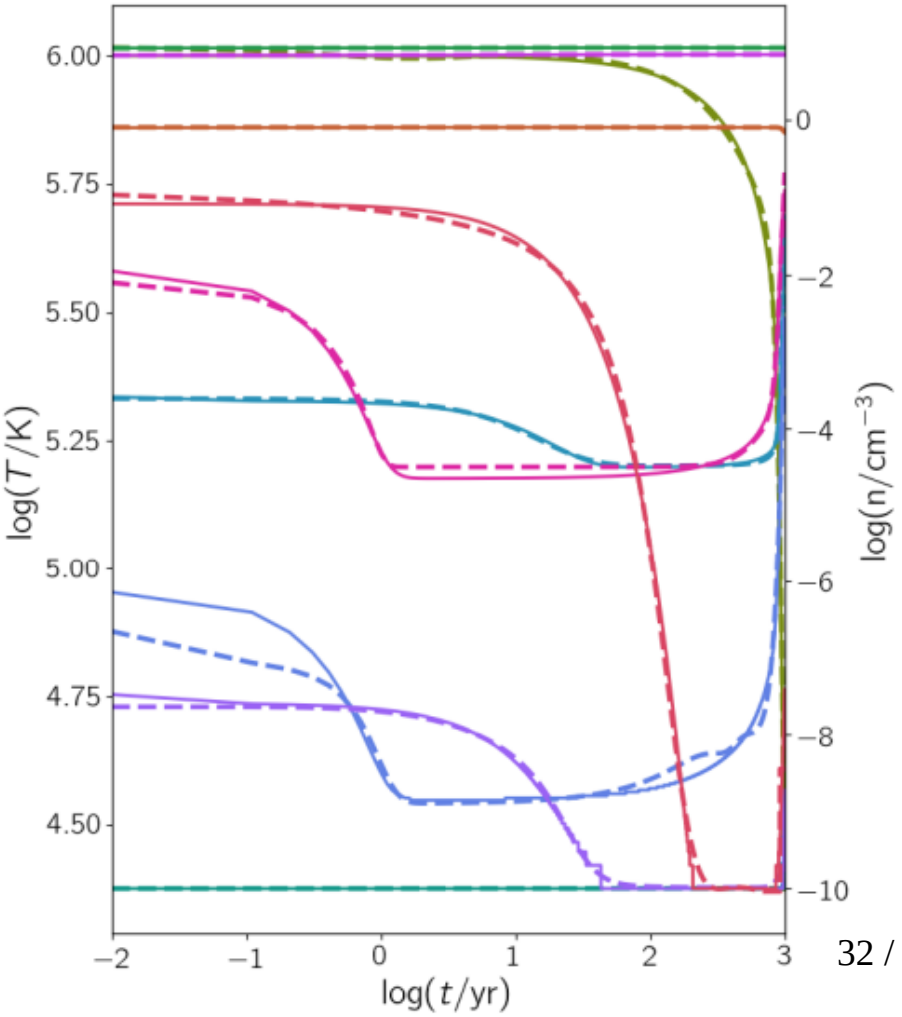
input	MRE	50%	75%	90%
T	0.0179	0.0114	0.0211	0.0353
e^-	0.0074	0.0044	0.0078	0.0151
H^-	0.0176	0.0076	0.0183	0.0398
H	0.0269	0.0190	0.0375	0.0547
He	0.0213	0.0150	0.0258	0.0422
H_2	0.0274	0.0167	0.0292	0.0548
H^+	0.0099	0.0060	0.0101	0.0178
He^+	0.0148	0.0082	0.0177	0.0314
H_2^+	0.0255	0.0146	0.0267	0.0552
He^{++}	0.0234	0.0125	0.0220	0.0436

- Training: ~40 GPUhrs
- Speed-up: 128X
- Accuracy: ~ 1%
- N ouliers: ~ 1/1.000.000

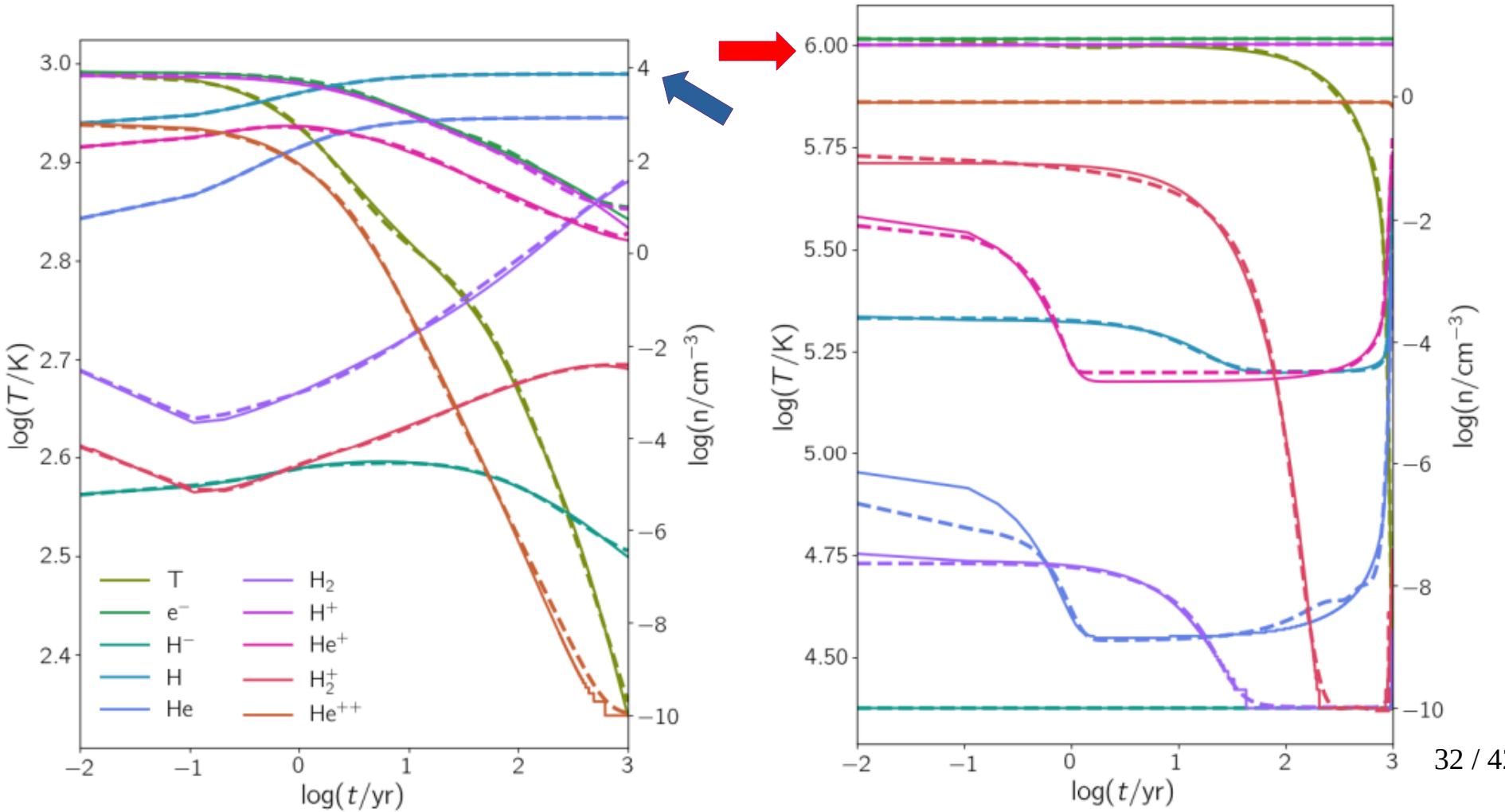
First time with radiation field coupling

Let's look at the solutions



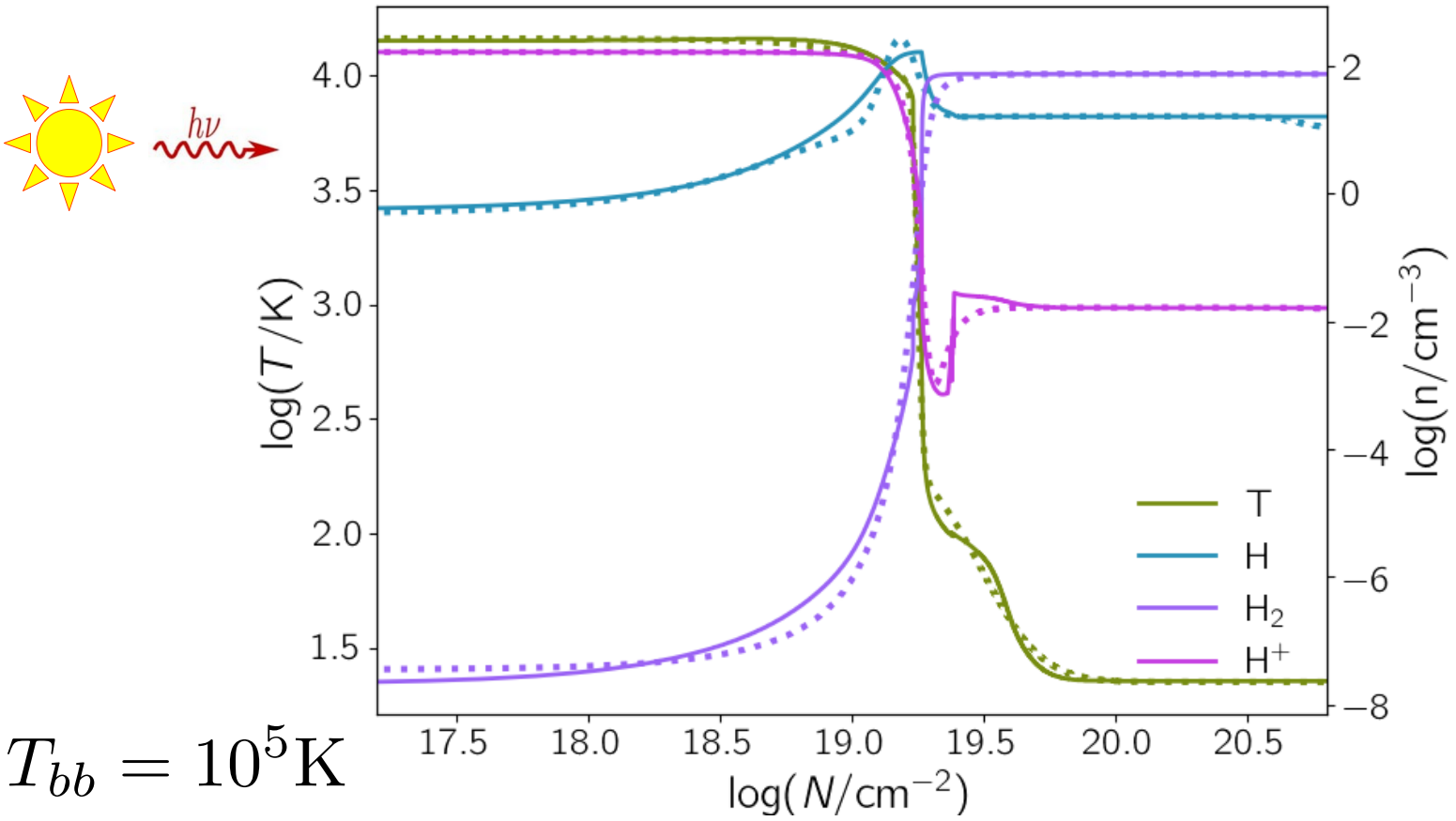


Let's look at the solutions



PDR test

$G/G_0 = 10$



$T_{bb} = 10^5 \mathrm{K}$

$R_c \simeq 16\mathrm{pc}$

$n_{tot} = 10^2 \mathrm{cm}^{-3}$

$t_{end} = 1\mathrm{Kyr}$

$T_{in} = 10^3 \mathrm{K}$

$n_{ion}/n \simeq 10^{-4}$

$n_{H_2}/n \simeq 0.4$

DeepONet summary

DeepONet summary

DeepONet summary

- Huge amount of training data required

DeepONet summary

- Huge amount of training data required
- Low training time required

DeepONet summary

- Huge amount of training data required
- Low training time required
- Accuracy around 1%, start to be competitive

DeepONet summary

- Huge amount of training data required
- Low training time required
- Accuracy around 1%, start to be competitive
- For the first time we include radiation field

Benchmark different emulators

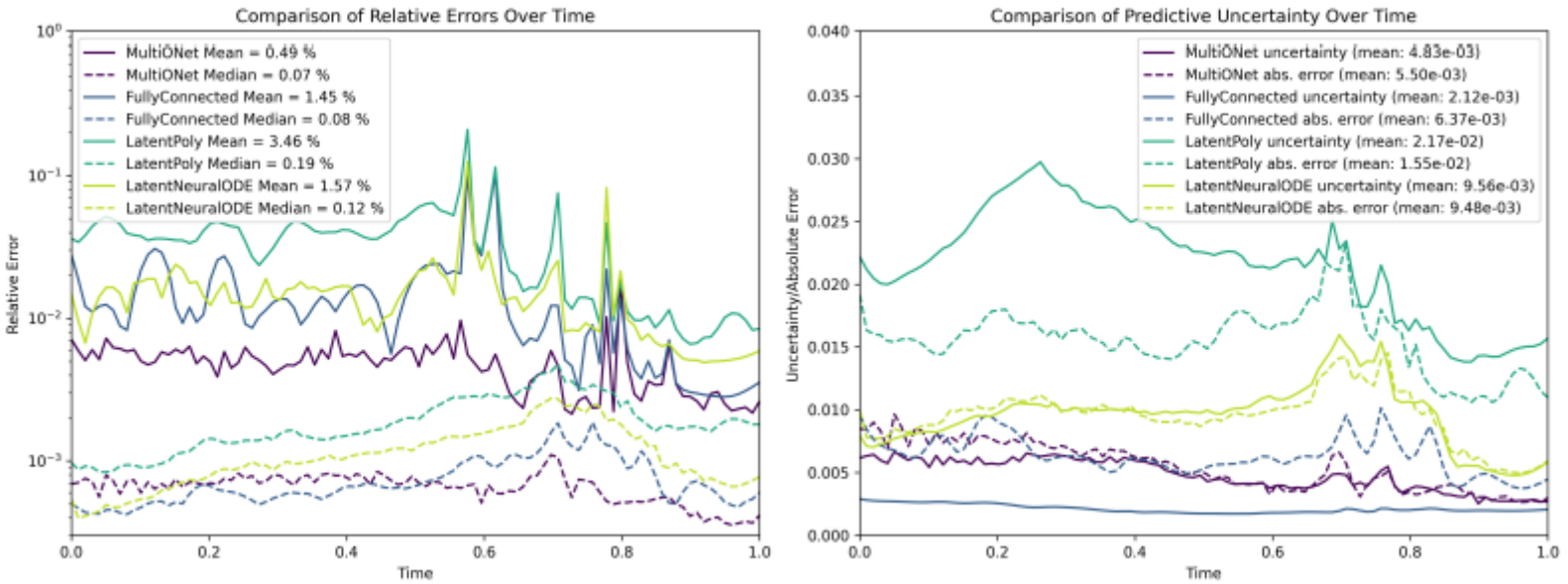
- https://github.com/robin-janssen/CODES-Benchmark



Benchmark different emulators

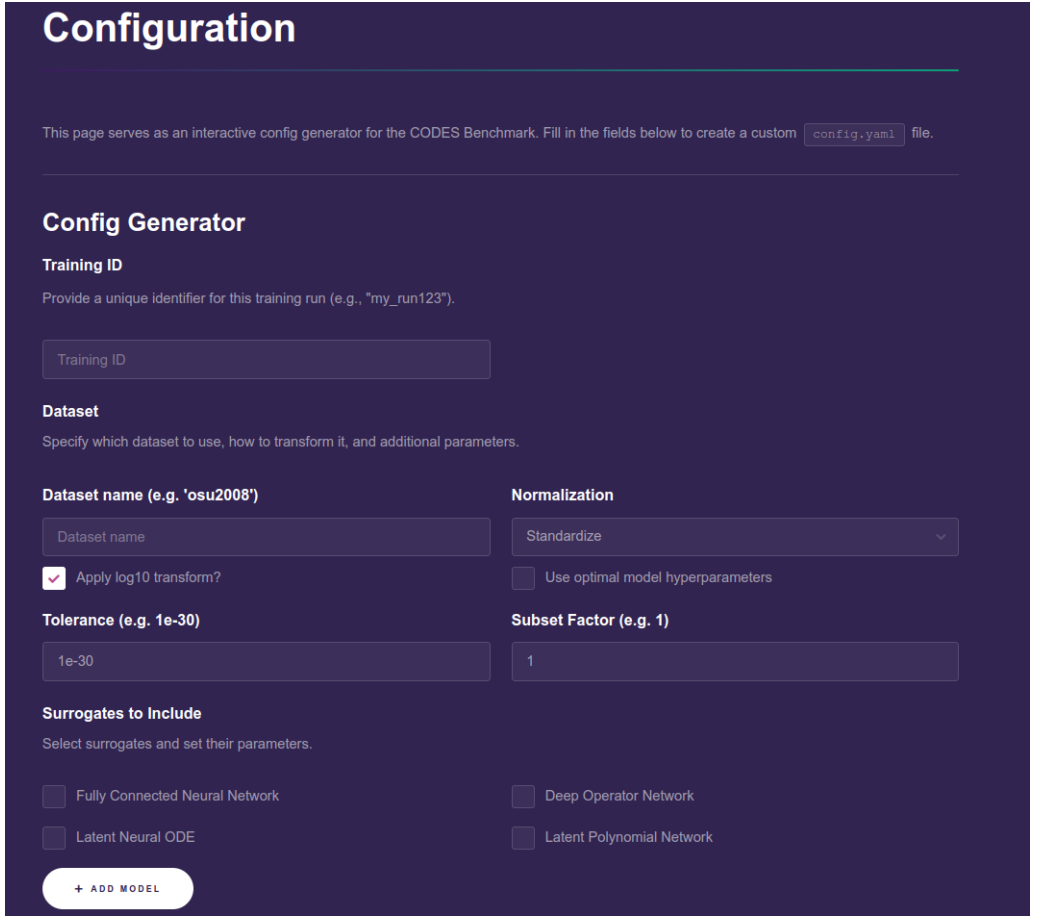
- https://github.com/robin-janssen/CODES-Benchmark

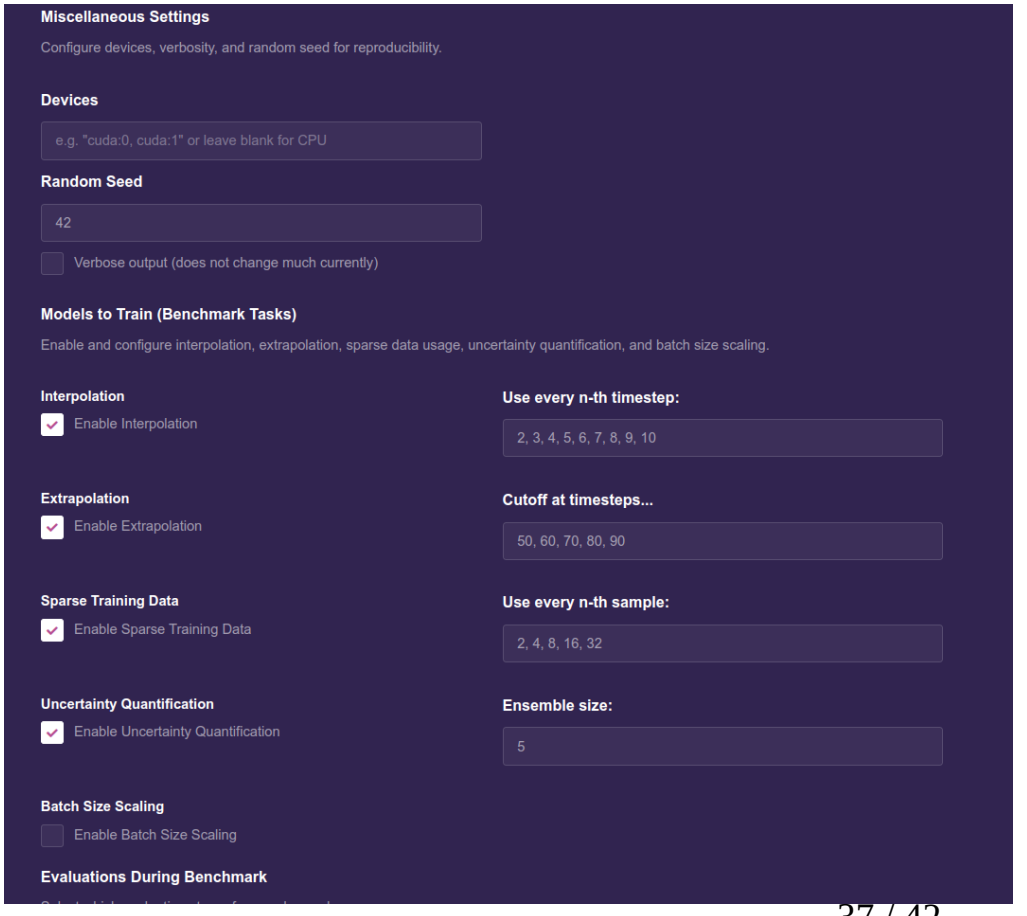




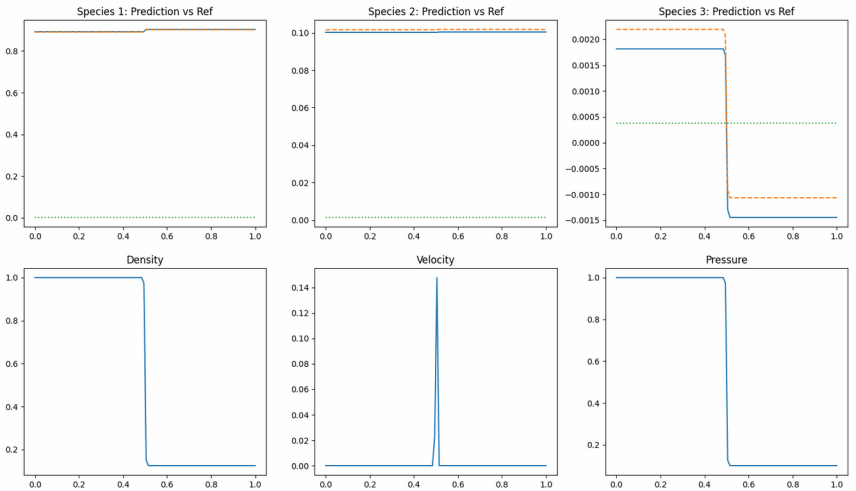
Jensen & Buck, 2024

Check the website!

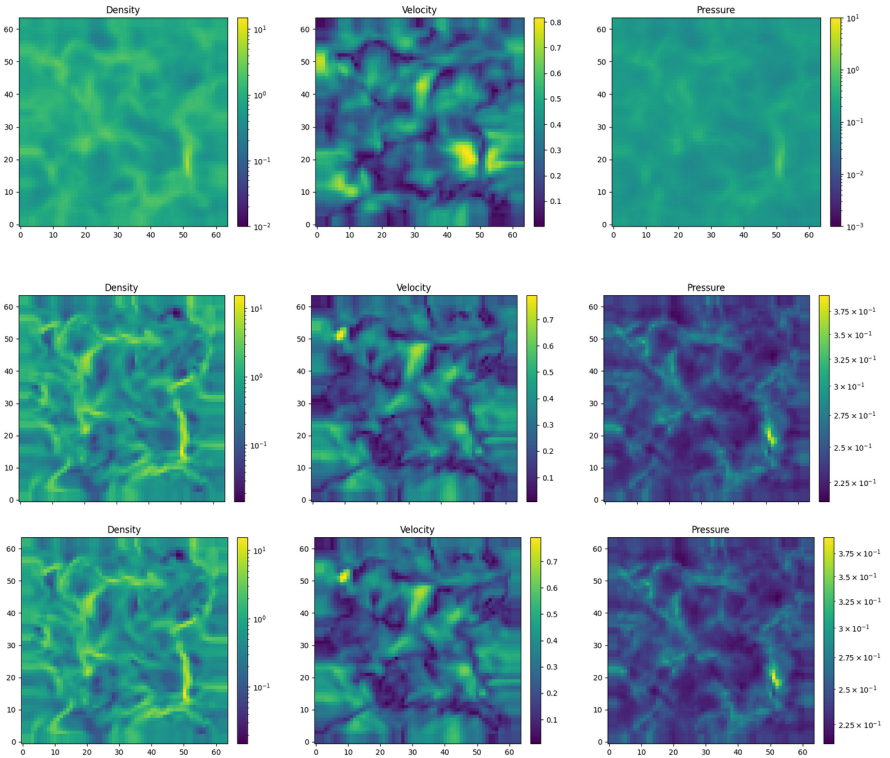




Merging hydro and surrogates



https://github.com/lorenzobranca/CHIMERA



Conclusions and questions

Conclusions and questions

Conclusions and questions

Conclusions and questions

- How much are needed surrogates in astrochemistry?

Conclusions and questions

- How much are needed surrogates in astrochemistry?
- Merge PINNs and Neural Operators (or others surrogates)?

Conclusions and questions

- How much are needed surrogates in astrochemistry?
- Merge PINNs and Neural Operators (or others surrogates)?
- Improve the representativeness of chemical dataset?

Conclusions and questions

- How much are needed surrogates in astrochemistry?
- Merge PINNs and Neural Operators (or others surrogates)?
- Improve the representativeness of chemical dataset?

Acknowledgements

Funded by the European Union

• This work is based upon work from COST Action CA21126 - Carbon molecular nanostructures in space (NanoSpace), supported by COST (European Cooperation in Science & Technology).

• COST (European Cooperation in Science & Technology) is a funding agency for research and innovation networks. Our Actions help connect research initiatives across Europe and enable scientists to grow their ideas by sharing them with their peers. This boosts their research, career and innovation.

• www.cost.eu
• https://www.cost.eu/actions/CA21126/
• https://research.iac.es/proyecto/nanospace/