Platform for Advanced Scientific Computing

Deep Learning Approaches for Solving Chemistry in Radiative Transfer Simulations

Philipp Denzel
Elena Gavagnin

Alexandre Refregier
Michele Bianco
Tommaso Boschi

Florina Ciorba
Osman Seckin Simsek

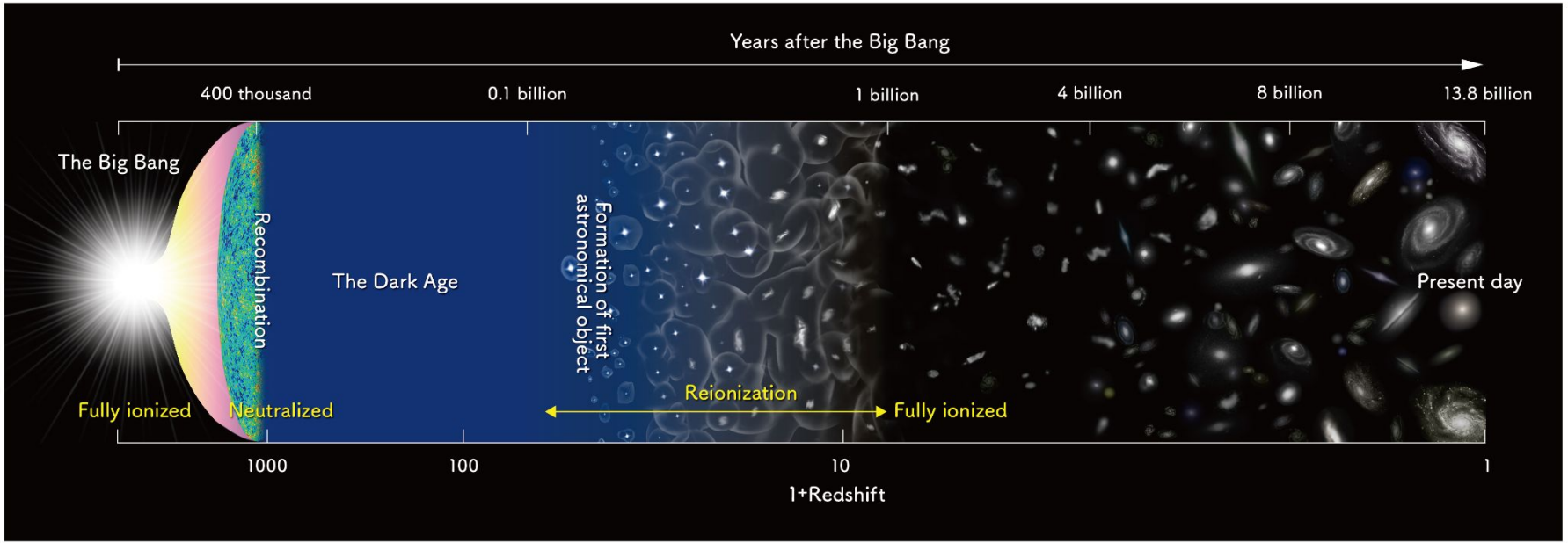
Christopher Bignamini
Sebastian Keller

Yves Revaz

Overview

1. **Epoch of Reionization**: chemistry is a key process
2. **ARTS4SKA project**: better EoR simulations (with AI and good SW)
3. **Neural ODEs**: accelerate chemistry ODE solutions
4. **Neural ODEs vs. PINNs** (physics-informed neural networks)

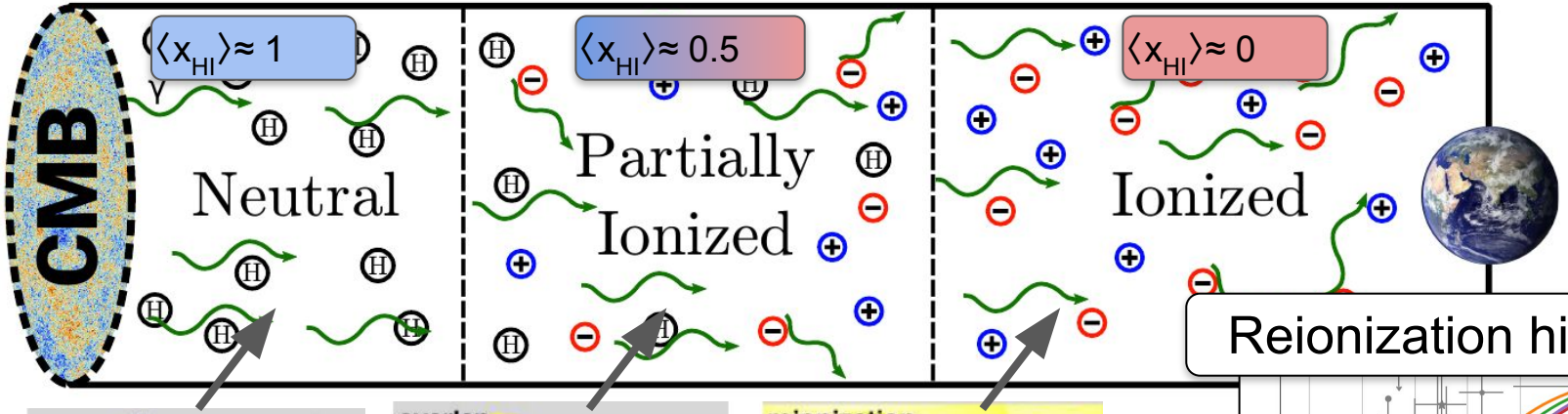
Cosmological timeline

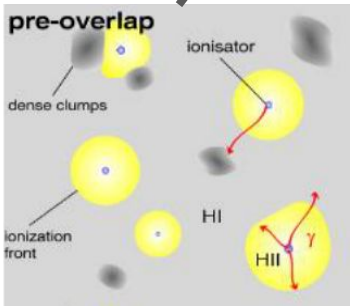


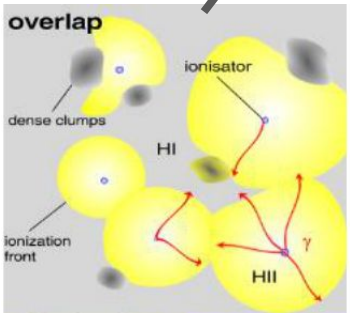
Credit: NAOJ

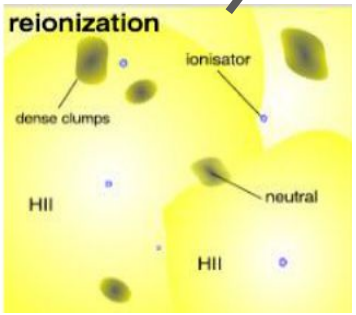
The Epoch of Reionization… in a Nutshell

The observed signal, **$\delta T_b(\theta,z)$, depends** on the redshift/frequency evolution of the **hydrogen (HI) volume-averaged neutral fraction**: $\langle x_{HI} \rangle \in [0, 1]$

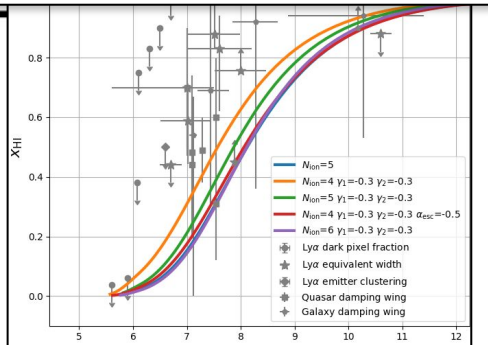


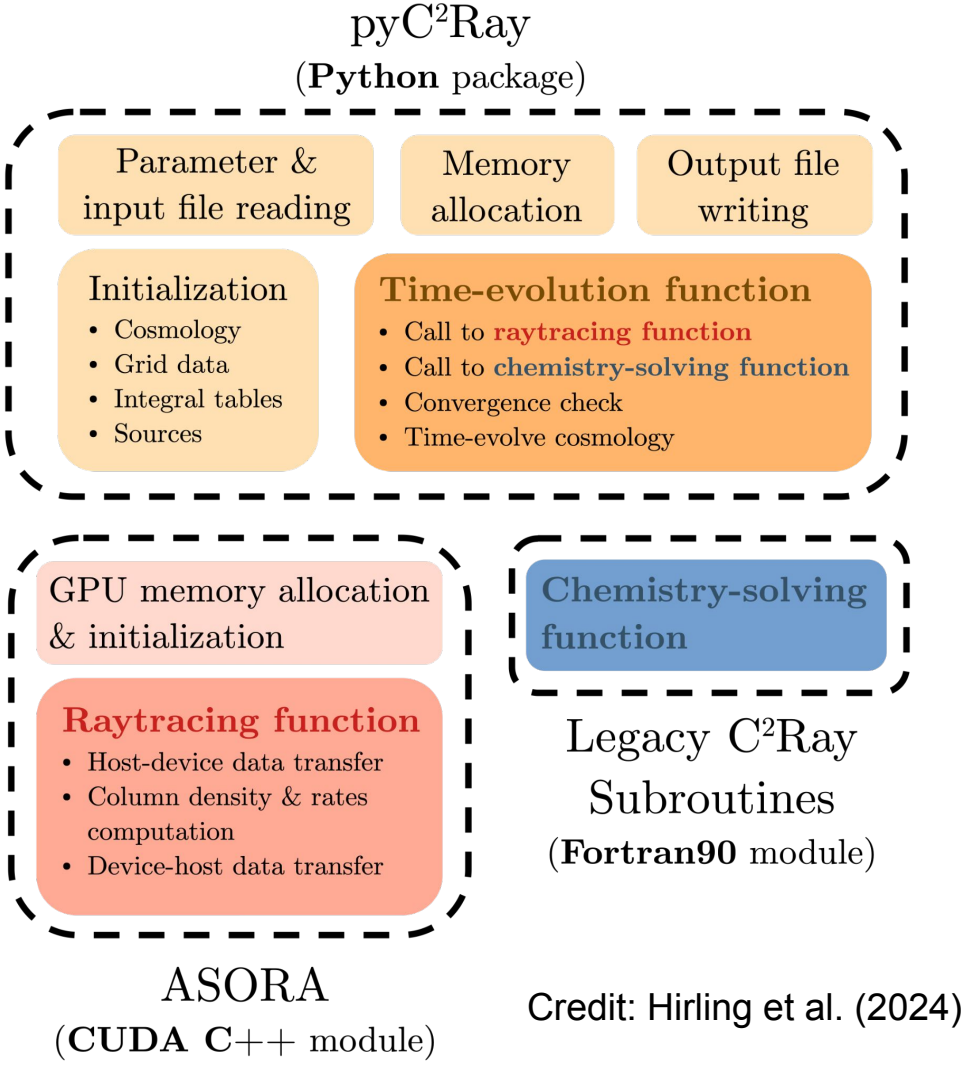




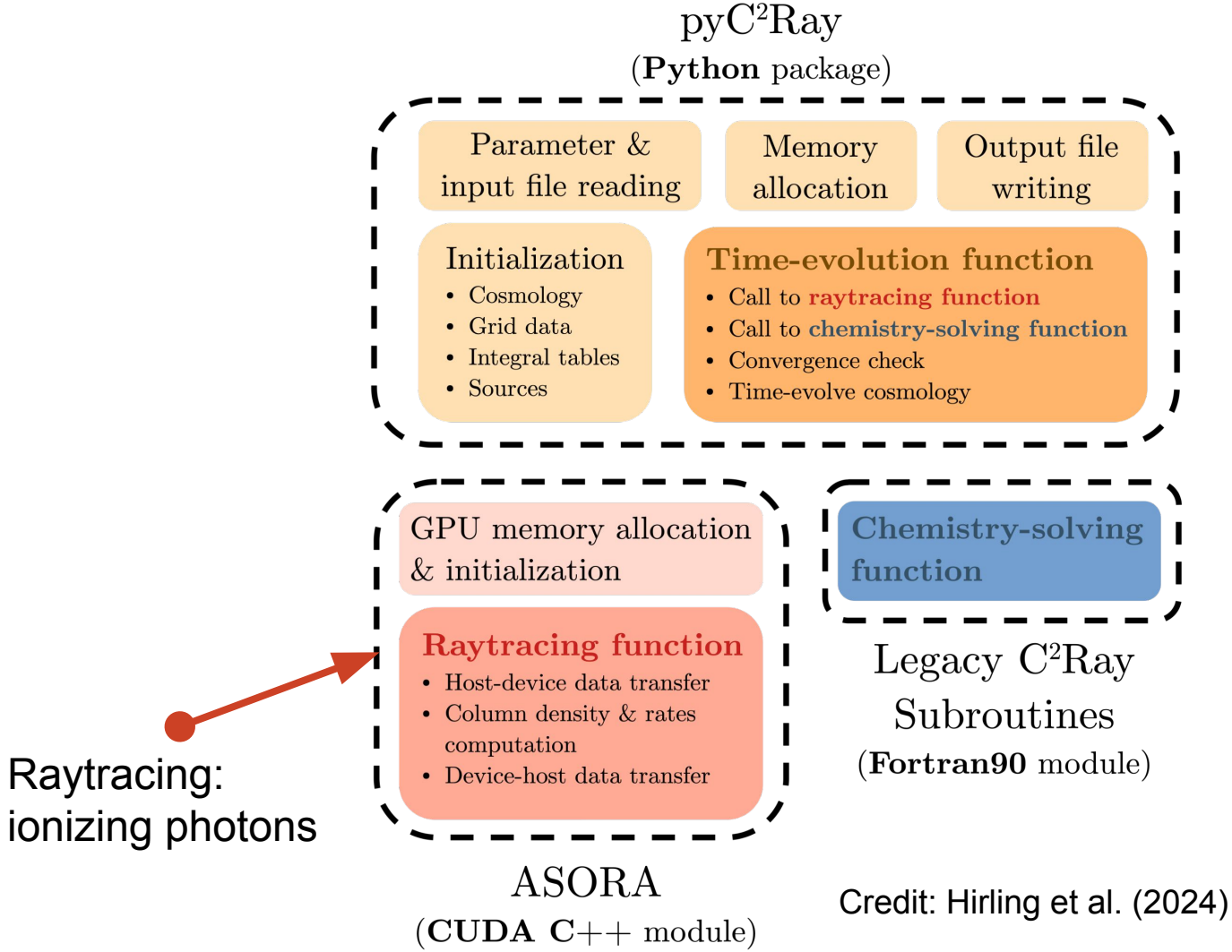


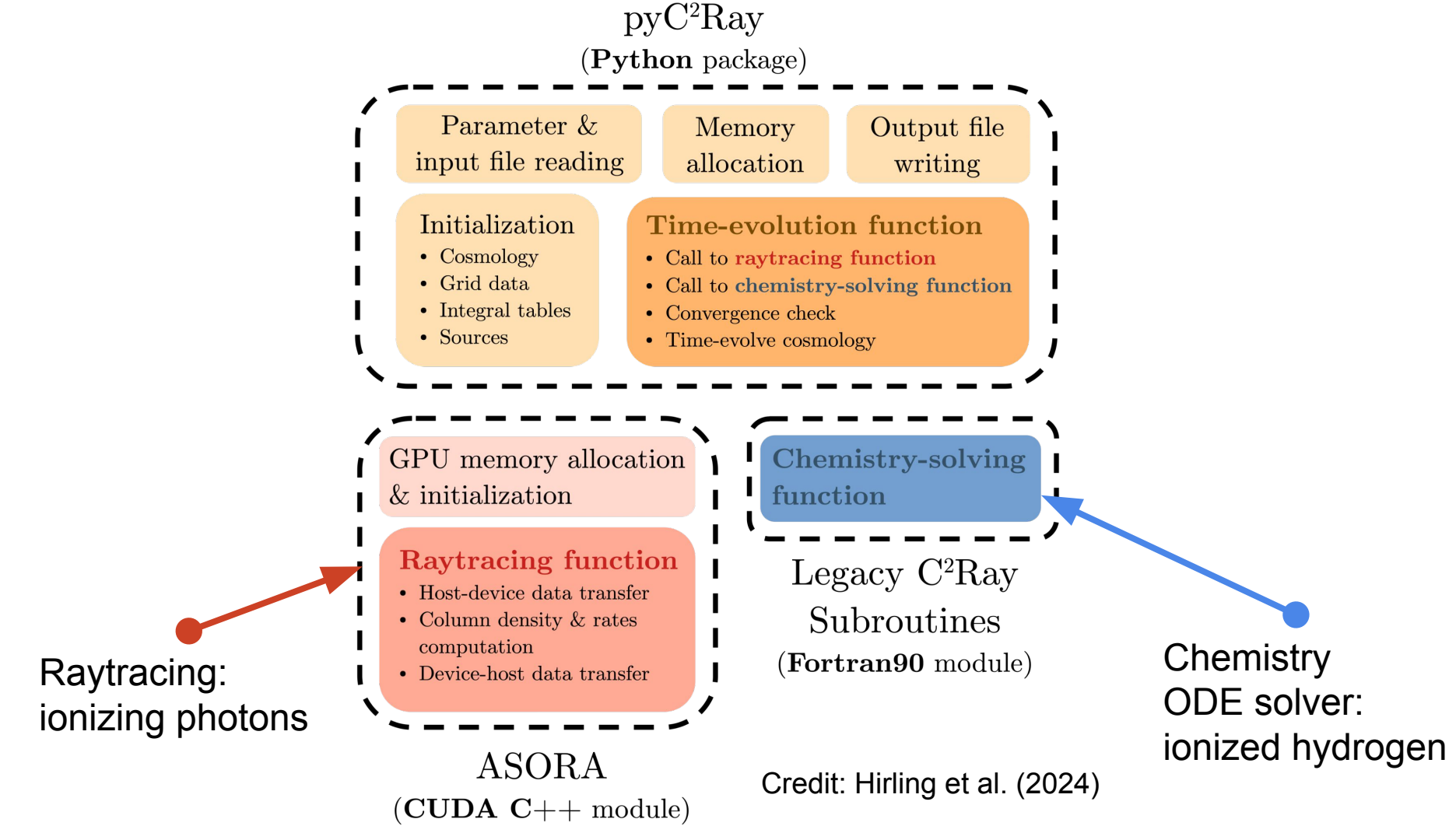
Reionization history





Credit: Hirling et al. (2024)


pyC²Ray
(Python package)
Parameter & input file reading
Memory allocation
Output file writing
Initialization
• Cosmology
• Grid data
• Integral tables
• Sources
Time-evolution function
• Call to raytracing function
• Call to chemistry-solving function
• Convergence check
• Time-evolve cosmology
GPU memory allocation & initialization
Raytracing function
• Host-device data transfer
• Column density & rates computation
• Device-host data transfer
ASORA
(CUDA C++ module)
Chemistry-solving function
Legacy C²Ray Subroutines
(Fortran90 module)
Raytracing: ionizing photons
Credit: Hirling et al. (2024)


pyC²Ray
(Python package)
Parameter & input file reading
Memory allocation
Output file writing
Initialization
• Cosmology
• Grid data
• Integral tables
• Sources
Time-evolution function
• Call to raytracing function
• Call to chemistry-solving function
• Convergence check
• Time-evolve cosmology
GPU memory allocation & initialization
Raytracing function
• Host-device data transfer
• Column density & rates computation
• Device-host data transfer
ASORA
(CUDA C++ module)
Chemistry-solving function
Legacy C²Ray Subroutines
(Fortran90 module)
Raytracing: ionizing photons
Chemistry ODE solver: ionized hydrogen
Credit: Hirling et al. (2024)

Goals of the ARTS4SKA project

pyC^{2}Ray code

Raytracing + ODE Solver

- Only UV radiation
- Post-processing of N-body
- Slow convergence to ODE solution
- No thermal- & gas hydro-dynamics
- Computational efficiency?

Goal 1: Improve pyC^{2}Ray post-process large scale EoR simulations

Goal 2: RT on-the-fly on existing N-body+hydro simulations:

- SPH-EXA (intermed. scale)
- SWIFT (galactic scale)

Goals of the ARTS4SKA project

pyC^{2}Ray code

Raytracing + ODE Solver

- Only UV radiation
- Post-processing of N-body
- Slow convergence to ODE solution
- No thermal- & gas hydro-dynamics
- Computational efficiency?

Extend Raytracing algorithm to include:

- Multi-frequency radiative transfer (X-ray, e^- collisional ionization, Lyman Warner)
- Photoheating of the intergalactic medium

- Radiative transfer on irregular grid
- Particle-to-mesh framework

Goal 1: Improve pyC^{2}Ray post-process large scale EoR simulations

Goal 2: RT on-the-fly on existing N-body+hydro simulations:

- SPH-EXA (intermed. scale)
- SWIFT (galactic scale)

Goals of the ARTS4SKA project

pyC^{2}Ray code

Raytracing + ODE Solver

- Only UV radiation
- Post-processing of N-body
- Slow convergence to ODE solution
- No thermal- & gas hydro-dynamics
- Computational efficiency?

AI models:

- Neural function(al)s: speed up convergence
- Include Helium chemistry equation

Extend Raytracing algorithm to include:

- Multi-frequency radiative transfer (X-ray, e^- collisional ionization, Lyman Warner)
- Photoheating of the intergalactic medium

- Radiative transfer on irregular grid
- Particle-to-mesh framework

Goal 1: Improve pyC^{2}Ray post-process large scale EoR simulations

Goal 2: RT on-the-fly on existing N-body+hydro simulations:

- SPH-EXA (intermed. scale)
- SWIFT (galactic scale)

Cosmological EoR Simulations with Ray Tracing

Solve ODE to simulate HI evolution during EoR:

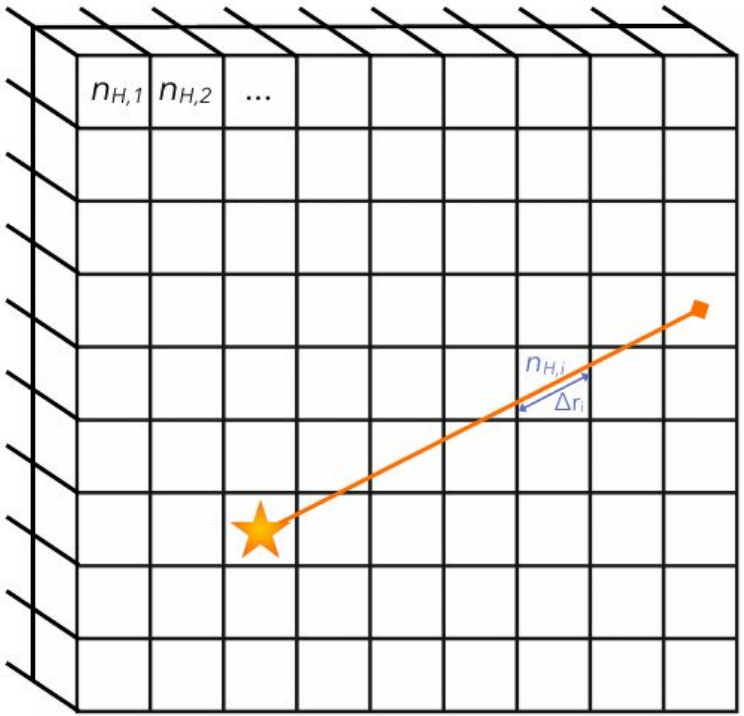
$$\frac{\mathrm{d}x}{\mathrm{d}t} = (1-x)(\Gamma + n_{\mathrm{e}} C_H) - x n_{\mathrm{e}} \alpha_H$$

Photo-ionization | Collisional ionization | Recombination

Compute the Ray Tracing for all ionising sources:

$$\Gamma_{\mathrm{local}}(r) = \frac{1}{4\pi r^2} \int_{\nu_{\mathrm{th}}}^{\infty} \frac{L_\nu \sigma_\nu e^{-\tau_\nu(r)}}{h\nu} \mathrm{d}\nu$$

$$\tau_\nu = \sigma_\nu N_{HI}$$

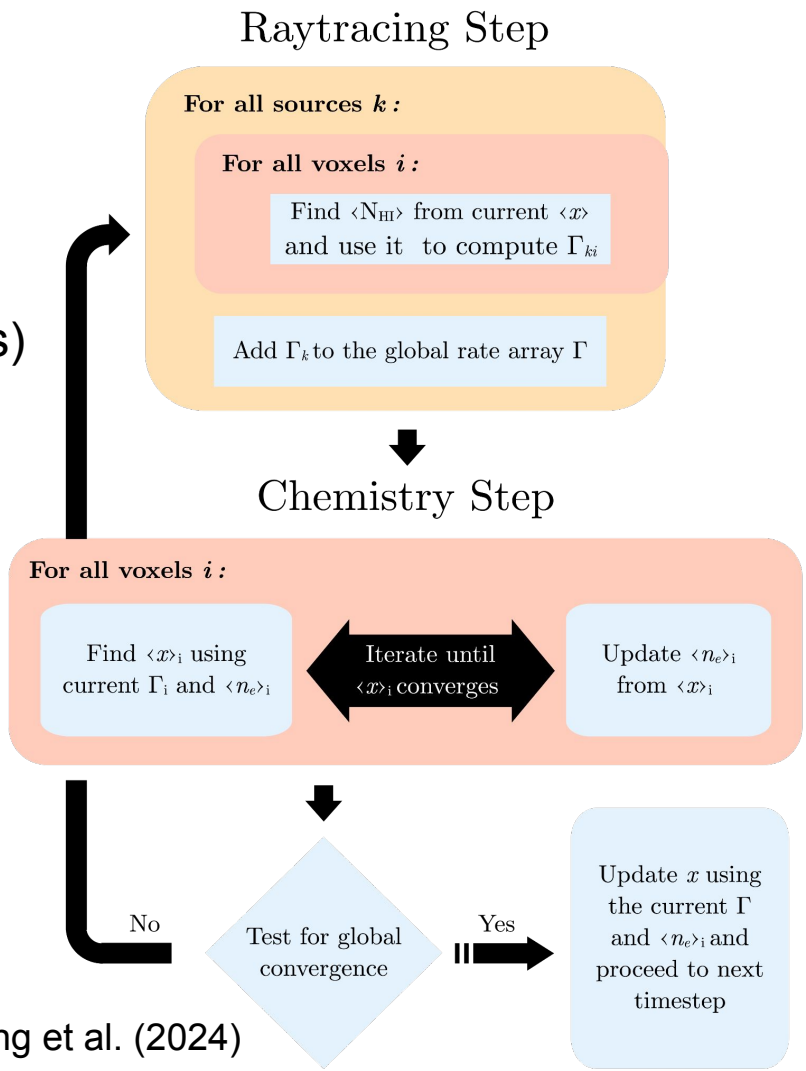


Optical depth:

quantify the UV-photons absorption from the source to the target

pyC^{2}Ray's TACS
(time-averaged chemistry solution)

- *in principle*, the chemistry step is easy to solve (e.g. with finite differences)
- **problem is Γ_i**: dependent
  - dependent on the solution $\langle x \rangle_i$
  - dependence highly non-local
- requires **extremely small timestep**
- pyC^{2}Ray's approach: **TACS** with time-averaged quantities $\langle x \rangle_i$, $\langle \Gamma \rangle_i$
- alternate RT and TACS steps until converged



Credit: Hirling et al. (2024)

Chemistry ODE

- Evolution of the neutral hydrogen fraction

$$\frac{\mathrm{d}x_{\mathrm{HII}}}{\mathrm{d}t} = (\Gamma_{\mathrm{HI}}(t) + n_e C_{\mathrm{HI}}(t))(1 - x_{\mathrm{HII}}(t)) - x_{\mathrm{HII}}(t) n_e \alpha_{\mathrm{HI}}(t)$$

$$\frac{\mathrm{d}x_{\mathrm{HII}}}{\mathrm{d}t} = A\,(1 - x_{\mathrm{HII}}(t)) - x_{\mathrm{HII}}(t)\,(B - A) = A - B x_{\mathrm{HII}}(t)$$

- Time-averaged chemistry solution (TACS) assumes this to be a first-order linear ODE (for a small timestep) with constant $n_e, \Gamma, C_H, \alpha_H$

$$x_{\mathrm{HII}}(t + \Delta t) = \left[x_{\mathrm{HII}}(t) - \frac{A}{B}\right] e^{-B\Delta t} + \frac{A}{B}$$

$$\langle x \rangle_{\mathrm{HII}} = \frac{A}{B} + \left(x_0 - \frac{A}{B}\right) \frac{(1 - e^{-B\Delta t})}{B\Delta t}$$

$$\frac{\mathrm{d}x_{\mathrm{HII}}}{\mathrm{d}t}=(\Gamma_{\mathrm{HI}}(t)-n_eC_{\mathrm{HI}}(t))(1-x_{\mathrm{HII}}(t))-x_{\mathrm{HII}}(t)n_e\alpha_{\mathrm{HI}}(t)$$

$$\frac{\mathrm{d}x_{\mathrm{HII}}}{\mathrm{d}t}=A\left(1-x_{\mathrm{HII}}(t)\right)-x_{\mathrm{HII}}(t)\left(B-A\right)=A-Bx_{\mathrm{HII}}(t)$$

$$x_{\mathrm{HII}}(t+\Delta t)=\left[x_{\mathrm{HII}}(t)-\frac{A}{B}\right]e^{-B\Delta t}+\frac{A}{B}$$

$$\left\langle x\right\rangle_{\mathrm{HII}} = \frac{A}{B} + (x_0 - \frac{A}{B}) \frac{(1 - e^{-B\Delta t})}{B\Delta t}$$

Beyond hydrogen things look more complicated

$$\frac{d}{dt}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{g}$$

$$\boldsymbol{x} = \begin{pmatrix} x_{\rm HII} \\ x_{\rm HeII} \\ x_{\rm HeIII} \end{pmatrix}, \; \boldsymbol{g} = \begin{pmatrix} U_{\rm HI} \\ U_{\rm HeI} \\ 0 \end{pmatrix}, \text{ and}$$

$$\boldsymbol{A} = \begin{pmatrix} -U_{\rm HI} \\ +\mathcal{R}_{\rm HII\to HI} n_{\rm e} & \frac{n_{\rm He}}{n_{\rm H}}\mathcal{R}_{\rm HeII\to HI} n_{\rm e} & \frac{n_{\rm He}}{n_{\rm H}}\mathcal{R}_{\rm HeIII\to HI} n_{\rm e} \\ 0 & \begin{matrix} -U_{\rm HeI} - U_{\rm HeII} \\ +\mathcal{R}_{\rm HeII\to HeI} n_{\rm e} \end{matrix} & \begin{matrix} -U_{\rm HeI} + \\ \mathcal{R}_{\rm HeIII\to HeI} n_{\rm e} \end{matrix} \\ 0 & U_{\rm HeII} & \mathcal{R}_{\rm HeIII\to HeII} n_{\rm e} \end{pmatrix}.$$

We explain our notation below.

U_i is the photo-ionization rate of species i

$$U_{\rm HI} = \Gamma^{\rm HI} + C_{\rm HI} n_{\rm e}$$

$$U_{\rm HeI} = \Gamma^{\rm HeI} + C_{\rm HeI} n_{\rm e}$$

$$U_{\rm HeII} = \Gamma^{\rm HeII} + C_{\rm HeII} n_{\rm e}$$

$$\begin{aligned}
\mathcal{R}_{\rm HII\to HI} &= -\alpha^B_{\rm HII} \\
\mathcal{R}_{\rm HeII\to HI} &= p\alpha^B_{\rm HeII} + y\alpha^1_{\rm HeII} \\
\mathcal{R}_{\rm HeII\to HeI} &= (1-y)\alpha^1_{\rm HeII} - \alpha^A_{\rm HeII} \\
\mathcal{R}_{\rm HeIII\to HI} &= (1-y^a_2-y^b_2)\alpha^1_{\rm HeIII} + \alpha^2_{\rm HeIII} \\
&\quad + \left[v(l-m+my) + (1-v)fz\right]\alpha^B_{\rm HeIII} \\
\mathcal{R}_{\rm HeIII\to HeI} &= y^b_2\alpha^1_{\rm HeIII} + \left[vm(1-y) + (1-v)f(1-z)\right]\alpha^B_{\rm HeIII} \\
&\quad + \alpha^A_{\rm HeIII} - y^a_2\alpha^1_{\rm HeIII} \\
\mathcal{R}_{\rm HeIII\to HeII} &= y^a_2\alpha^1_{\rm HeIII} - \alpha^A_{\rm HeIII}\,.
\end{aligned}$$

Analytical solution (provided time step is small)

$$\begin{aligned}
x_{\rm HII}(t+\Delta t) &= b_{11}c_1 e^{\lambda_1 t} + b_{12}c_2 e^{\lambda_2 t} + b_{13}c_3 e^{\lambda_3 t} + p_1, \\
x_{\rm HeII}(t+\Delta t) &= b_{21}c_1 e^{\lambda_1 t} + b_{22}c_2 e^{\lambda_2 t} + b_{23}c_3 e^{\lambda_3 t} + p_2, \\
x_{\rm HeIII}(t+\Delta t) &= b_{31}c_1 e^{\lambda_1 t} + b_{32}c_2 e^{\lambda_2 t} + b_{33}c_3 e^{\lambda_3 t} + p_3.
\end{aligned}$$

Time-averaged solution $\quad \langle x_{\rm HI}\rangle = 1 - \langle x_{\rm HII}\rangle$

$$\langle x_{\rm HII}\rangle = \frac{b_{11}c_1}{\lambda_1\Delta t}\left(e^{\lambda_1\Delta t}-1\right) + \frac{b_{12}c_2}{\lambda_2\Delta t}\left(e^{\lambda_2\Delta t}-1\right) + \frac{b_{13}c_3}{\lambda_3\Delta t}\left(e^{\lambda_3\Delta t}-1\right)$$

$$\langle x_{\rm HeI}\rangle = 1 - \langle x_{\rm HeII}\rangle - \langle x_{\rm HeIII}\rangle$$

$$\langle x_{\rm HeII}\rangle = \frac{b_{21}c_1}{\lambda_1\Delta t}\left(e^{\lambda_1\Delta t}-1\right) + \frac{b_{22}c_2}{\lambda_2\Delta t}\left(e^{\lambda_2\Delta t}-1\right) + \frac{b_{23}c_3}{\lambda_3\Delta t}\left(e^{\lambda_3\Delta t}-1\right)$$

$$\langle x_{\rm HeIII}\rangle = \frac{b_{31}c_1}{\lambda_1\Delta t}\left(e^{\lambda_1\Delta t}-1\right) + \frac{b_{32}c_2}{\lambda_2\Delta t}\left(e^{\lambda_2\Delta t}-1\right) + \frac{b_{33}c_3}{\lambda_3\Delta t}\left(e^{\lambda_3\Delta t}-1\right)$$

AI approaches to be compared

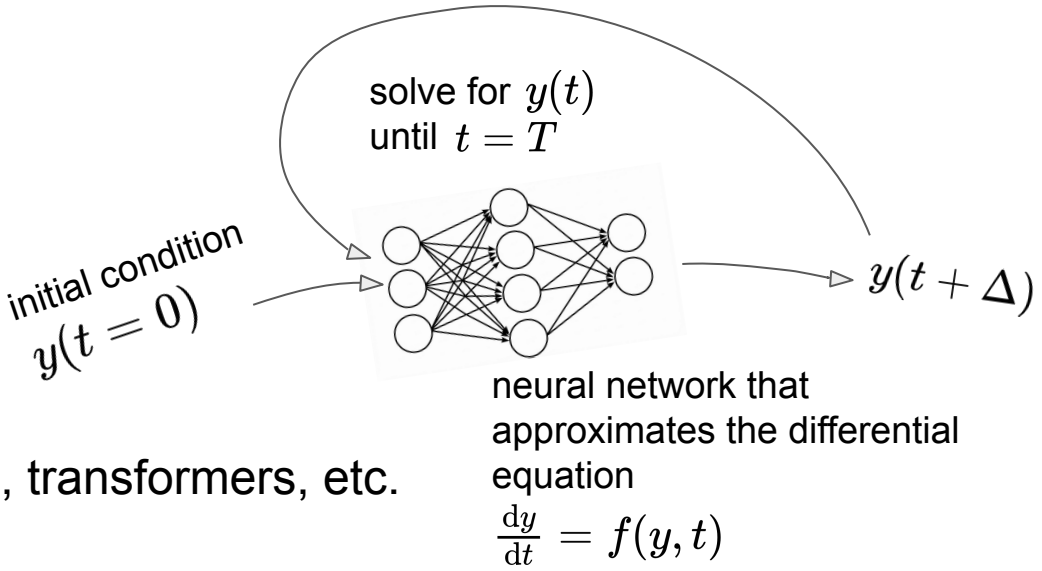
- Deep generative models trained on simulation data:

  latent diffusion models (rectified flows)

- Deep generative models as PINNs (regularization on physical properties)
- Neural ODEs
- Hybrid approach (PINNs + Neural ODE)?

Neural ODEs

- **Core idea**: replace discrete neural network layers with a "time-continuous" differentiable solver
- **Advantages**:
  - memory-efficient
  - adaptive computation (various ODE solvers)
  - continuous in time
- Continuous normalizing flows, transformers, etc.



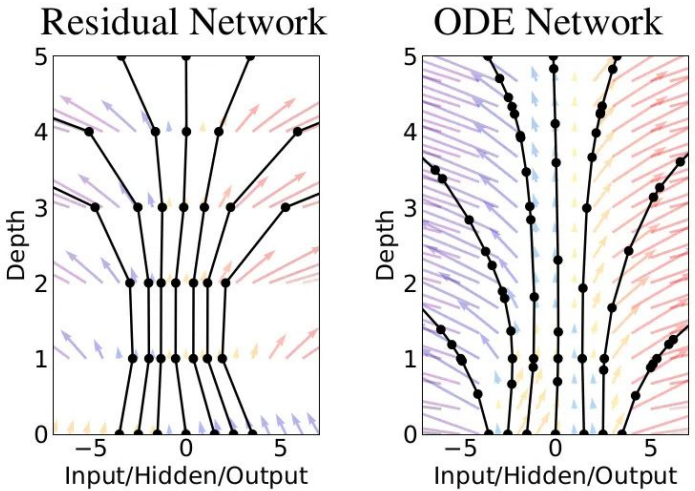
Intuition: Neural ODEs from ResNets

- ResNets provide incremental updates of hidden states via $h_{l+1} = h_l + f(h_l, \theta_l)$

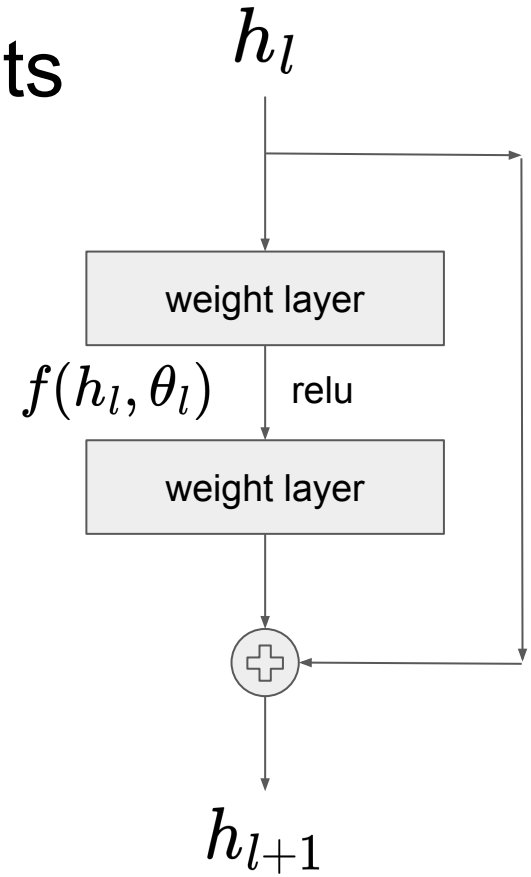
- Euler's formula

- In the limit for l:

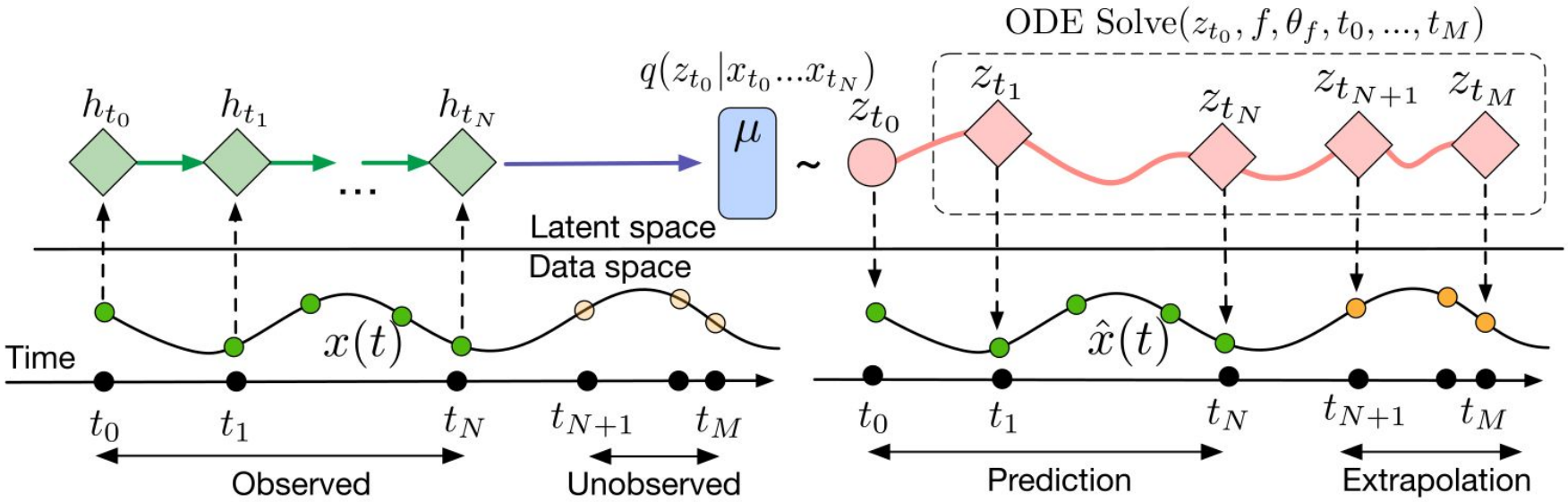
$$\frac{\mathrm{d}h}{\mathrm{d}t} = f(h(t), t, \theta)$$



Credit: arXiv:1806.07366



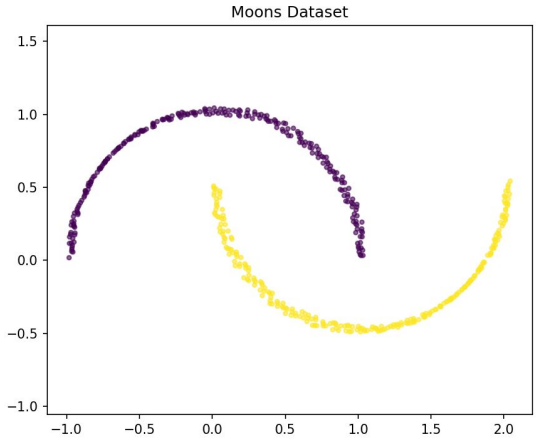
Generative Neural ODEs



Credit: arXiv:1806.07366

Benchmarks against common neural ODE frameworks

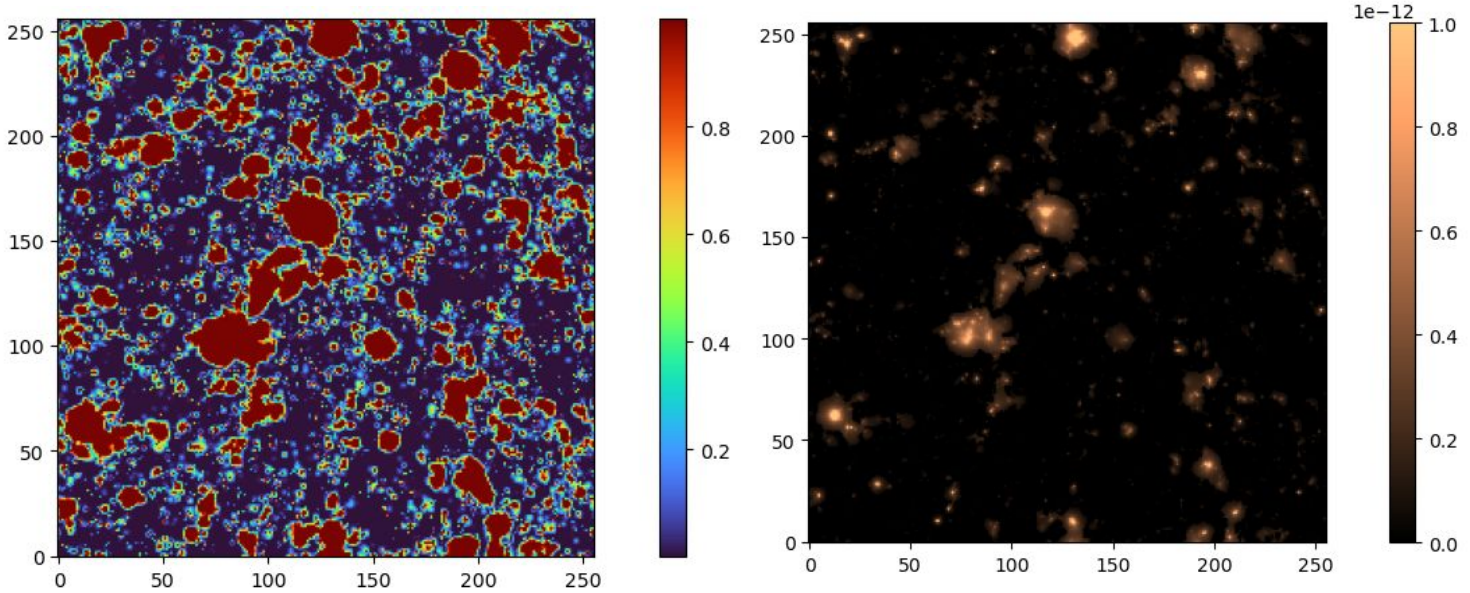
- **torchdiffeq**: most prominent framework (many solvers, poor speed)
- **torchdyn**: poorly maintained, limited set of numerical solvers (Tsitouras5)
- **diffrax**: faster, only solvers (Tsitouras5; no neural ODE utils)
- **my own** direct CUDA/Pytorch implementation (c3li for reference):
  - Euler, RK 4, RK45, explicit/implicit solvers (expansion WIP)
  - ROCm/HIP versions work (are slower though)



Test case: 2 Moons dataset, 3 layer MLP, 500 AdamW steps

Framework	**CUDA** (A100) / avg. acc.	**AMD** (RX 7900XTX)
torchdiffeq (DoPri5)	182.3s / 2.43e-6	242.4s
torchdyn (Tsitouras5)	283.1s / 9.30e-7	-
diffrax (Tsitouras5)	139.8s / 4.86e-7	-
own (c3li, RK45)	142.9s / 8.3e-5	183.7.s

Dataset generated with pyC^{2}Ray for training



redshifts:

- from 21.482
- to 4.989
- 187 steps

volume:

- 256^3 voxels
- L = 200 Mpc

Inferred hydrogen fractions (left) and photoionization maps (right) at redshift 7.875. Accuracy is the main problem for convergence and needs multiple raytracing and chemistry steps for proper EoR transition.

Preliminary tests:

- simple transformer does not converge reliably (more than 10x slower compared to simulation)
  $\longrightarrow$PINN regularizations needed
- Neural ODE faster compared to simulations, but still higher numerical errors (more convergence steps required).

Platform for Advanced Scientific Computing

Get in contact

Philipp Denzel
denp@zhaw.ch

Centre for Artificial Intelligence
Zurich University of Applied Sciences
ZHAW
Technikumstr. 71, 8400 Winterthur

Alexandre Refregier
Michele Bianco

Elena Gavagnin
Philipp Denzel

Florina Ciorba
Osman Seckin Simsek

Yves Revaz