Speeding up cosmological simulations with machine learning

Elif Ozlem Adiguzel

June 05, 2025

Supervised by: Dr. David Harvey, Dr. Yves Revaz

Modeling the Cosmos Through Simulations

Modeling the Cosmos Through Simulations

Model

ΛCDM

EPFL

Modeling the Cosmos Through Simulations

Model

ΛCDM

Observations

JWST, Euclid

Planck

Modeling the Cosmos Through Simulations

Model

ΛCDM

Observations

JWST, Euclid

Planck

Simulations

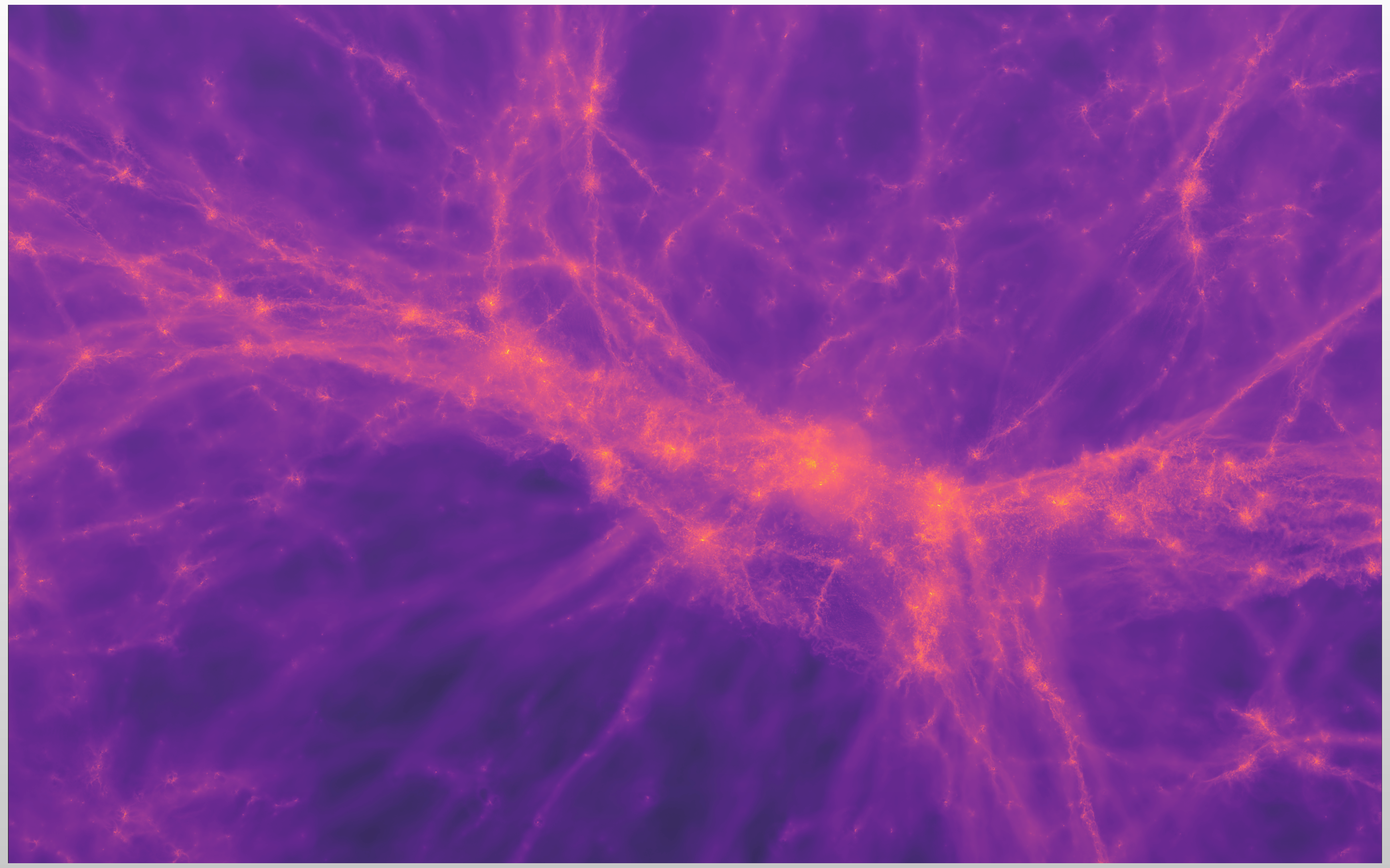


Image Credit: TNG Collaboration

Modeling the Cosmos Through Simulations

Observations

JWST, Euclid

Planck

Model

ΛCDM

Simulations

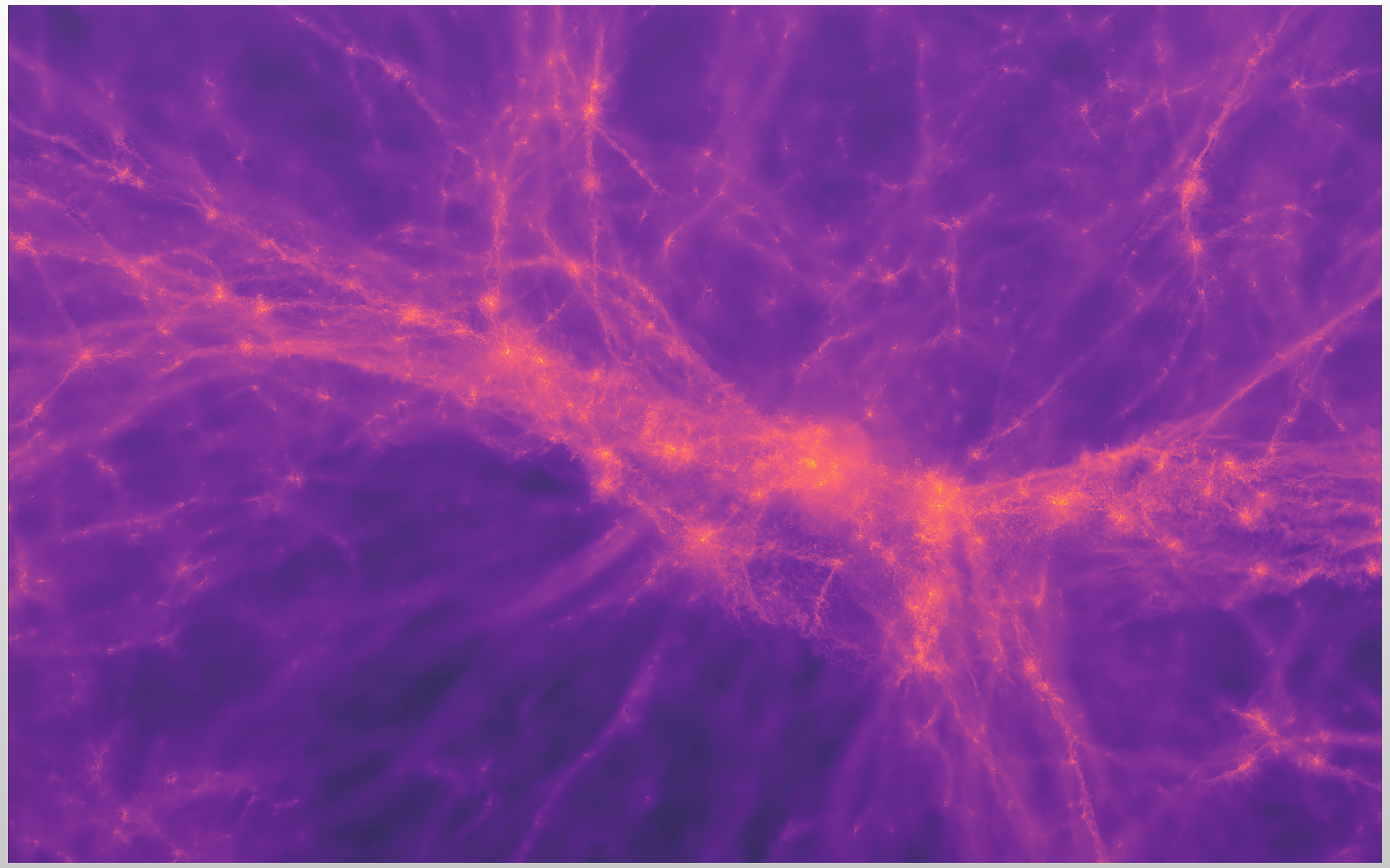


Image Credit: TNG Collaboration

EPFL

Modeling the Cosmos Through Simulations

Model

ΛCDM

Observations

JWST, Euclid

Planck

Simulations

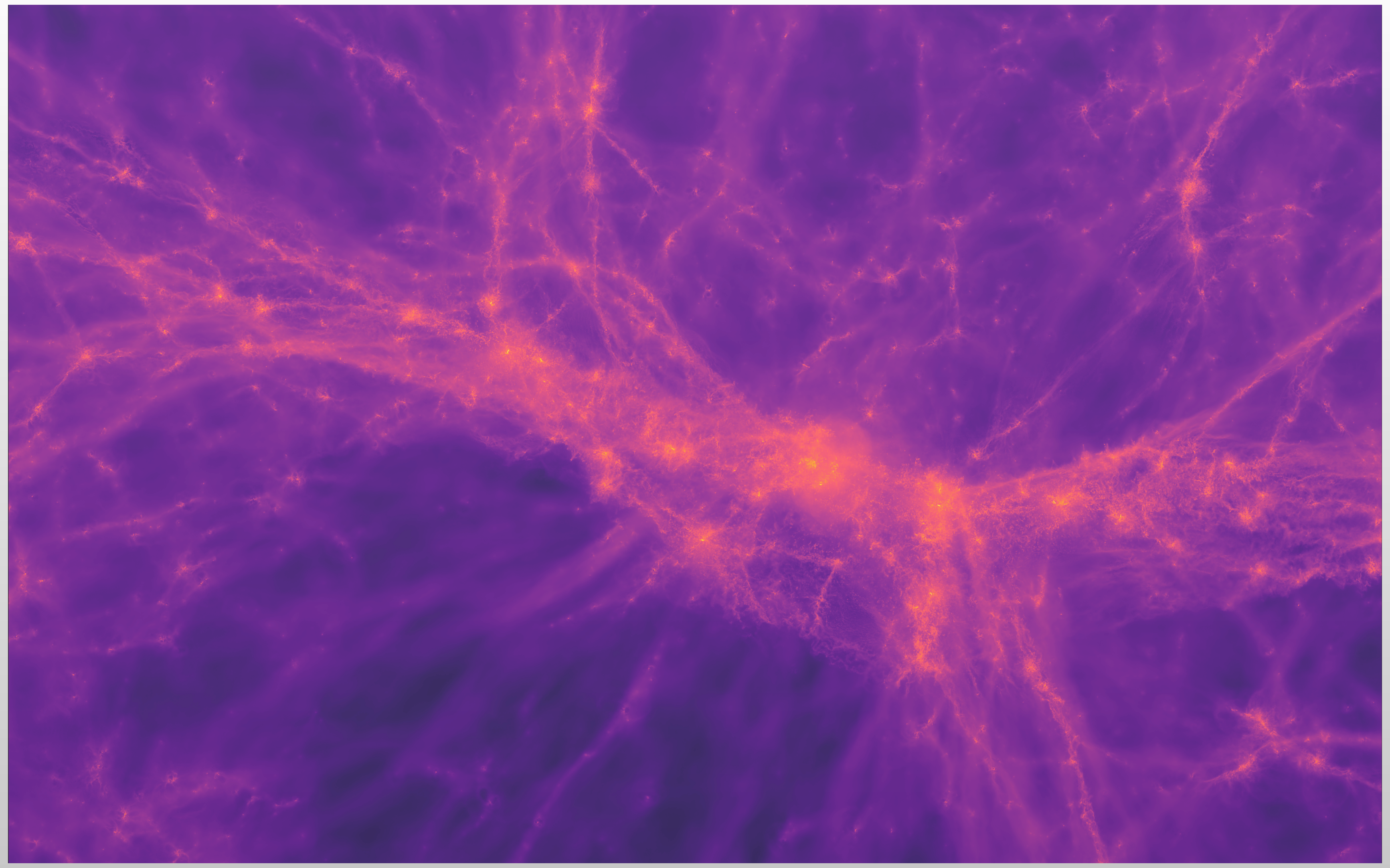


Image Credit: TNG Collaboration

Modeling the Cosmos Through Simulations

Model

ΛCDM

Observations

JWST, Euclid

Planck

Simulations

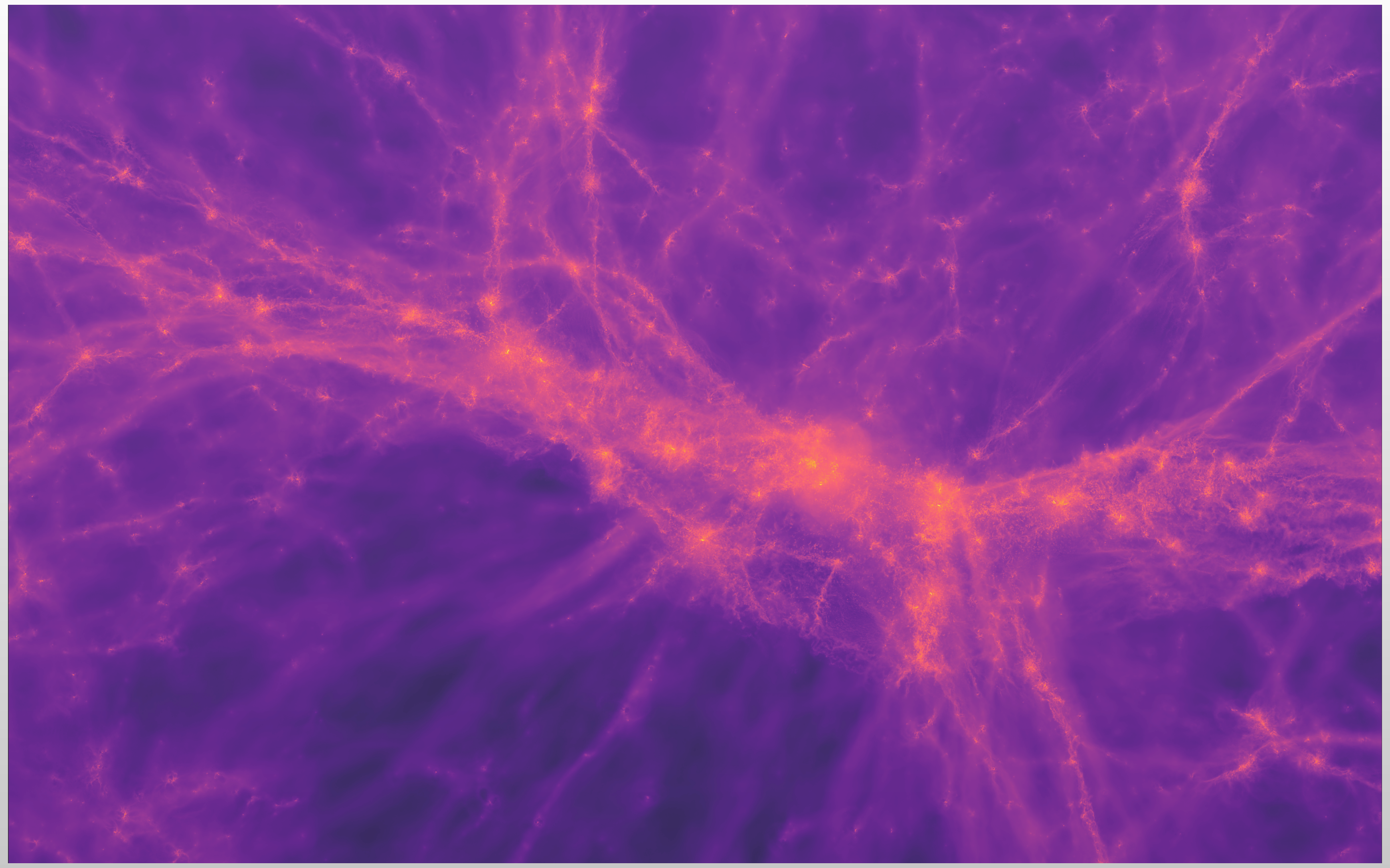


Image Credit: TNG Collaboration

compare

EPFL

Modeling the Cosmos Through Simulations

Model

ΛCDM

test

Observations

JWST, Euclid

Planck

compare

Simulations

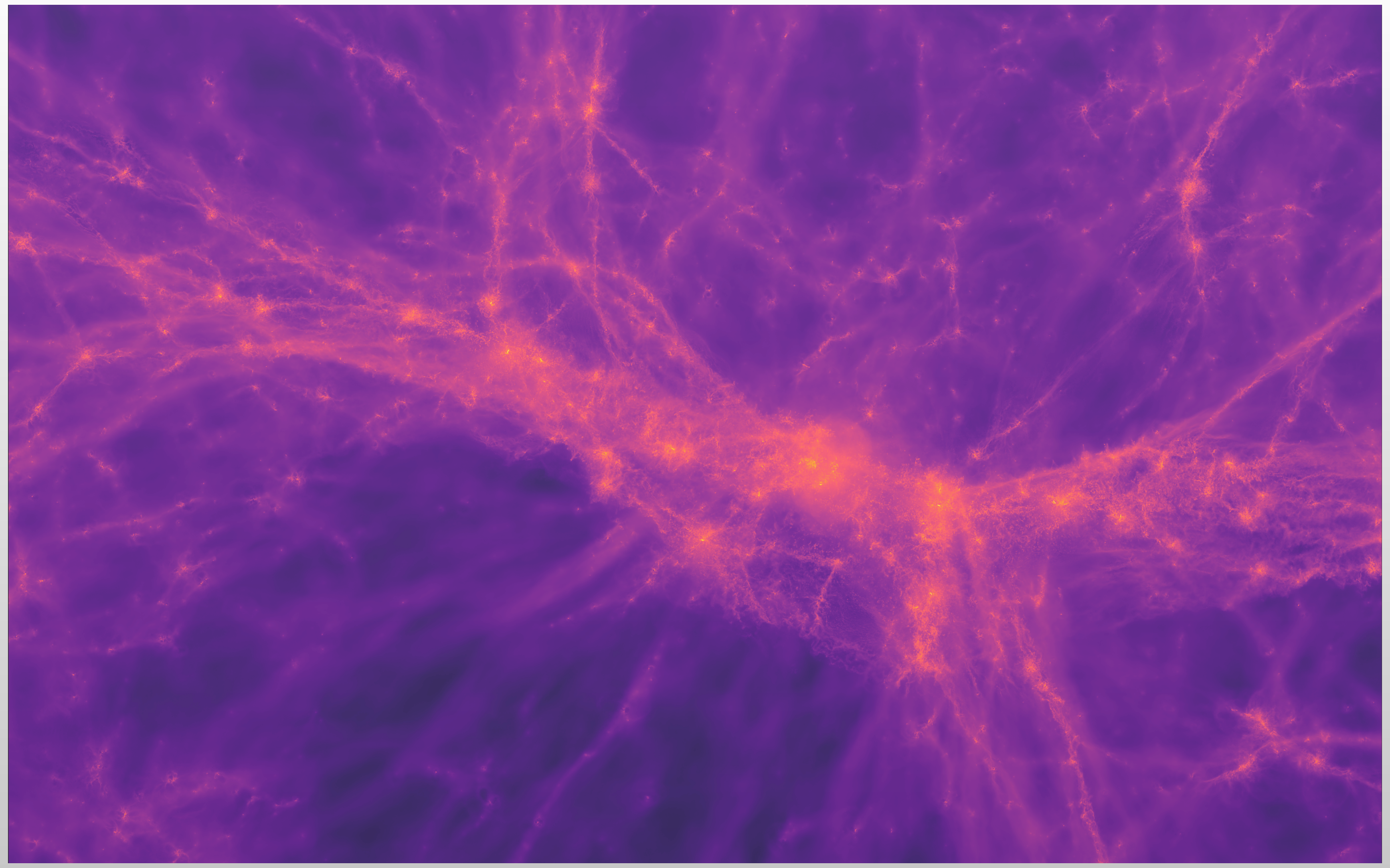


Image Credit: TNG Collaboration

Modelling the Cosmos Through Simulations

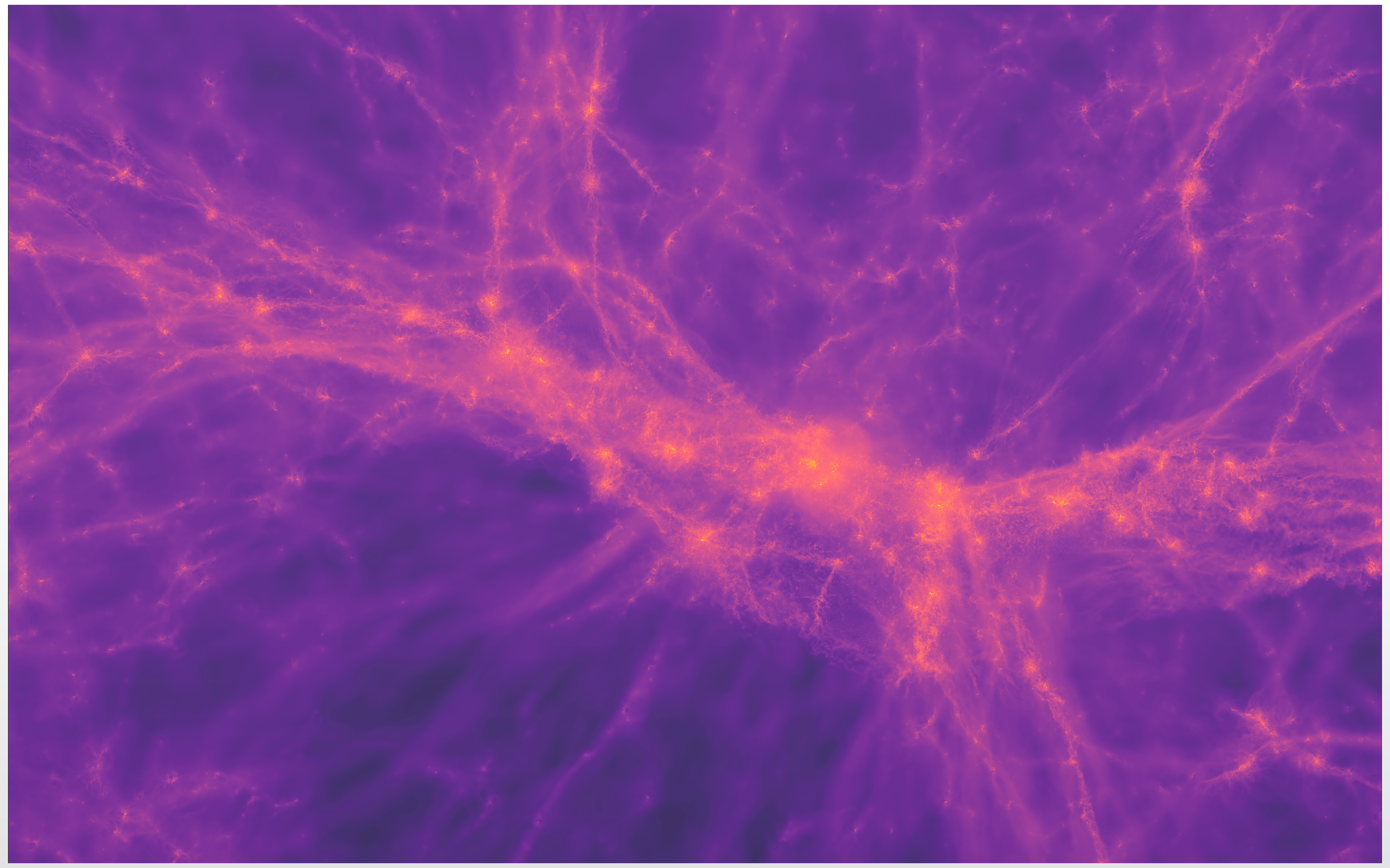


Image Credit: TNG Collaboration

Modelling the Cosmos Through Simulations

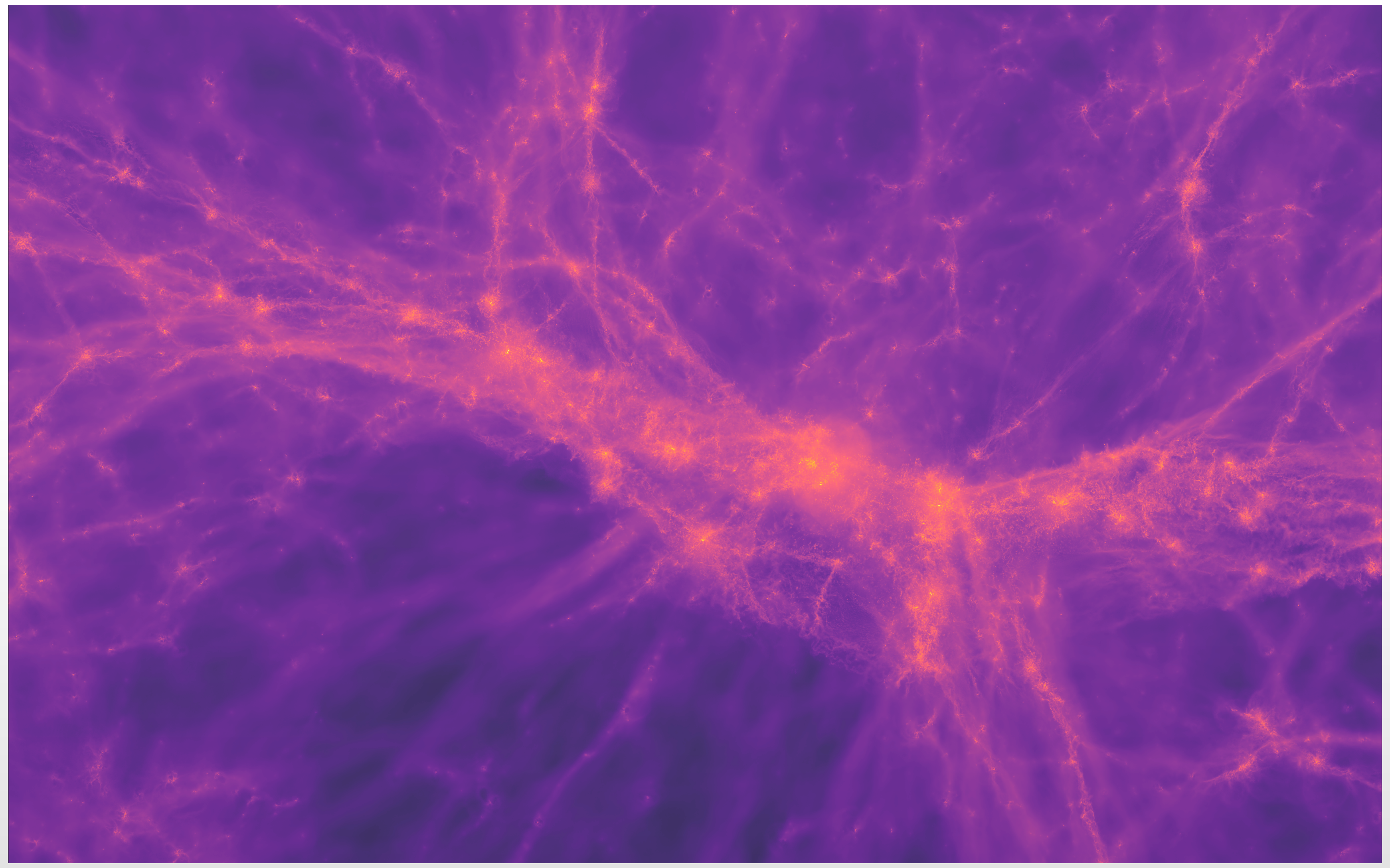
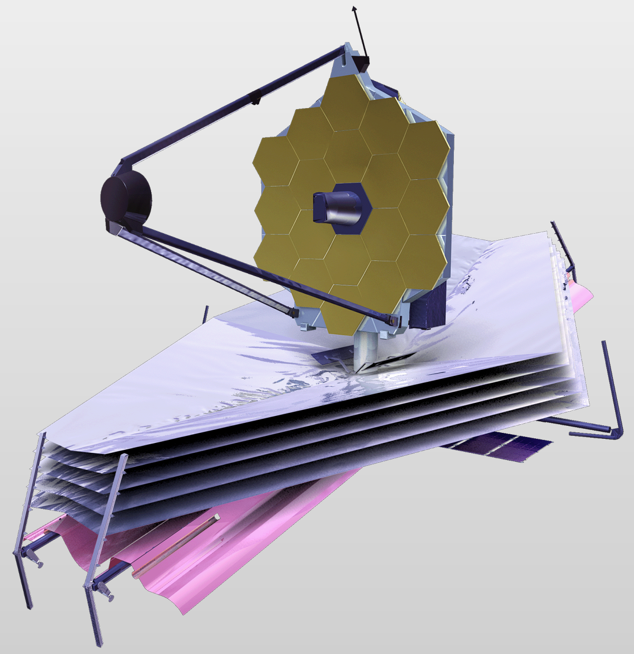


Image Credit: TNG Collaboration

- New observations challenge current models



Modelling the Cosmos Through Simulations

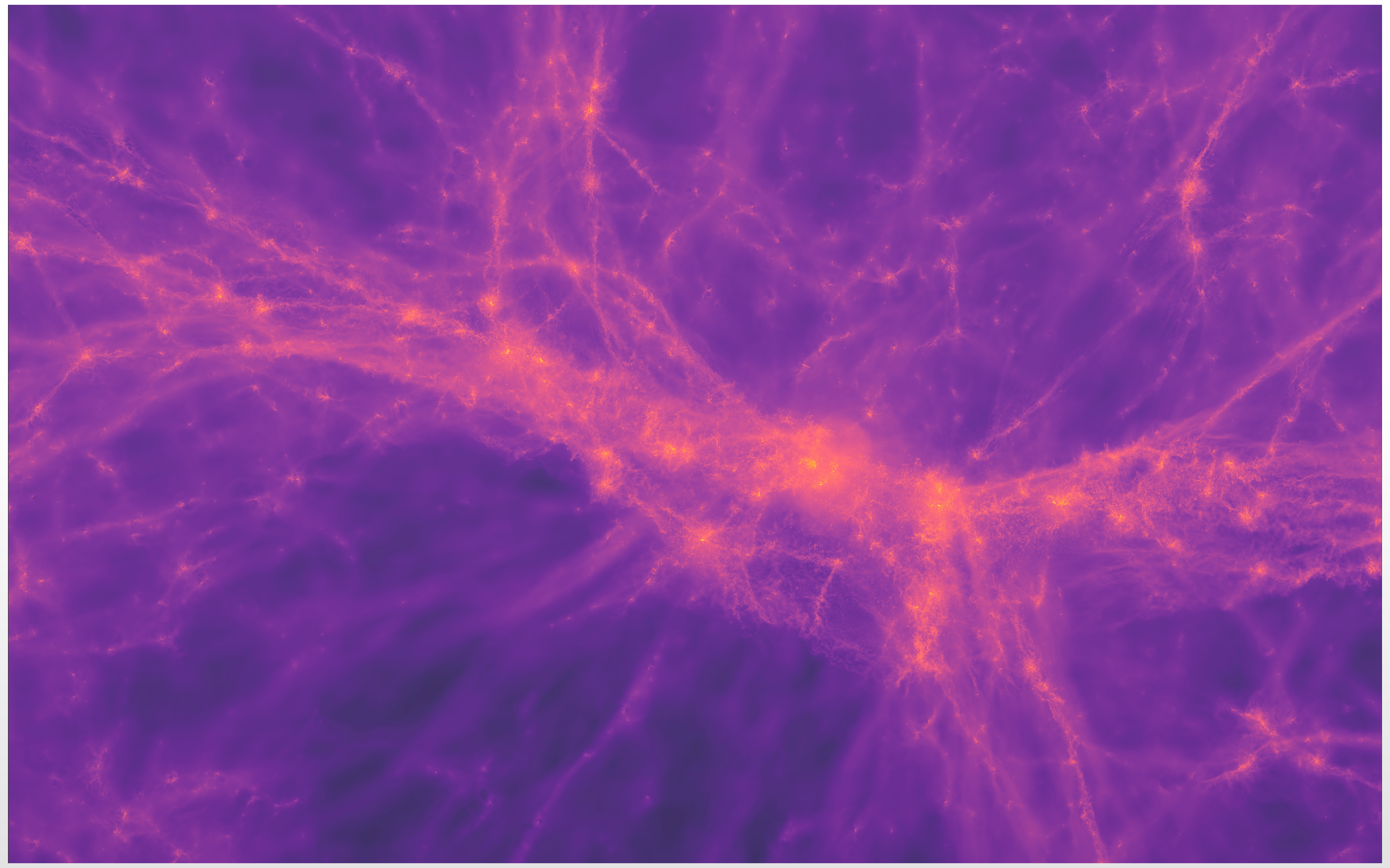
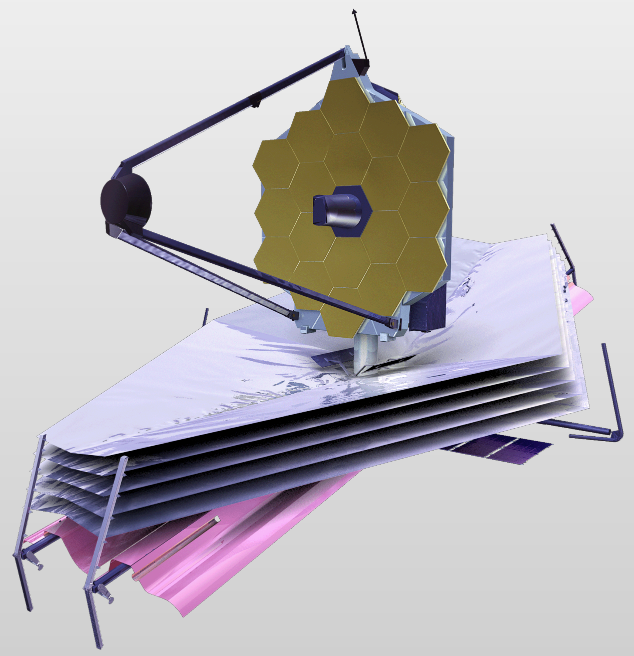


Image Credit: TNG Collaboration

- New observations challenge current models
- Bigger, higher resolution simulations are needed



Modelling the Cosmos Through Simulations

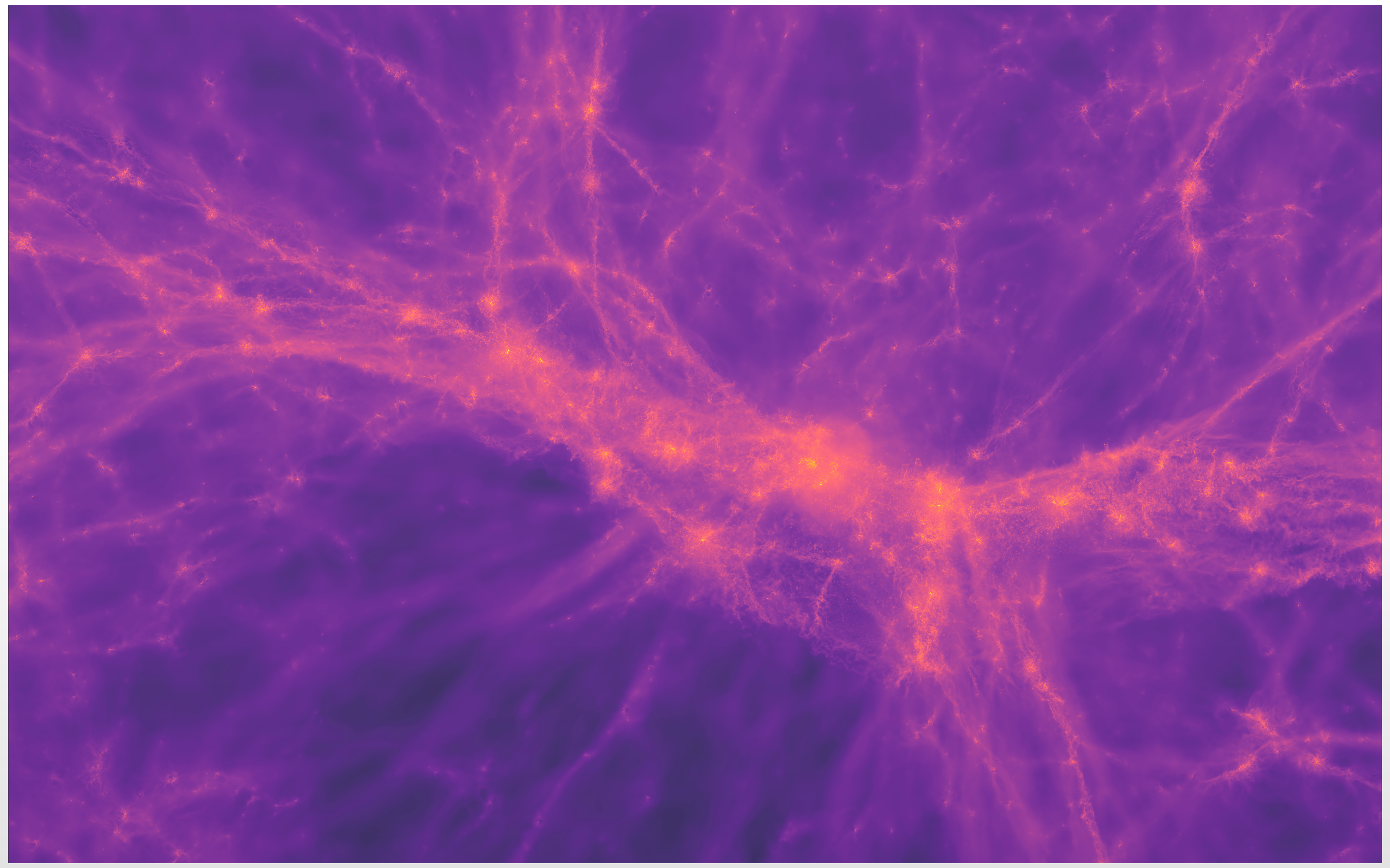
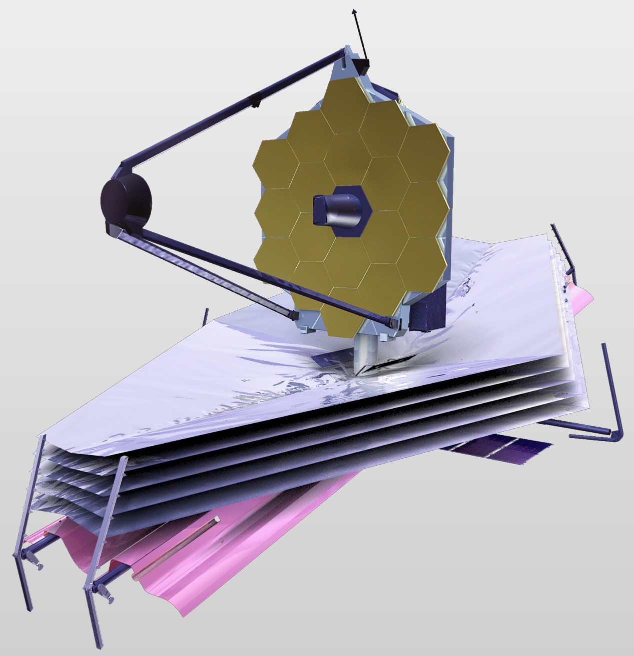


Image Credit: TNG Collaboration

- New observations challenge current models
- Bigger, higher resolution simulations are needed

→ **Time consuming**



Modelling the Cosmos Through Simulations

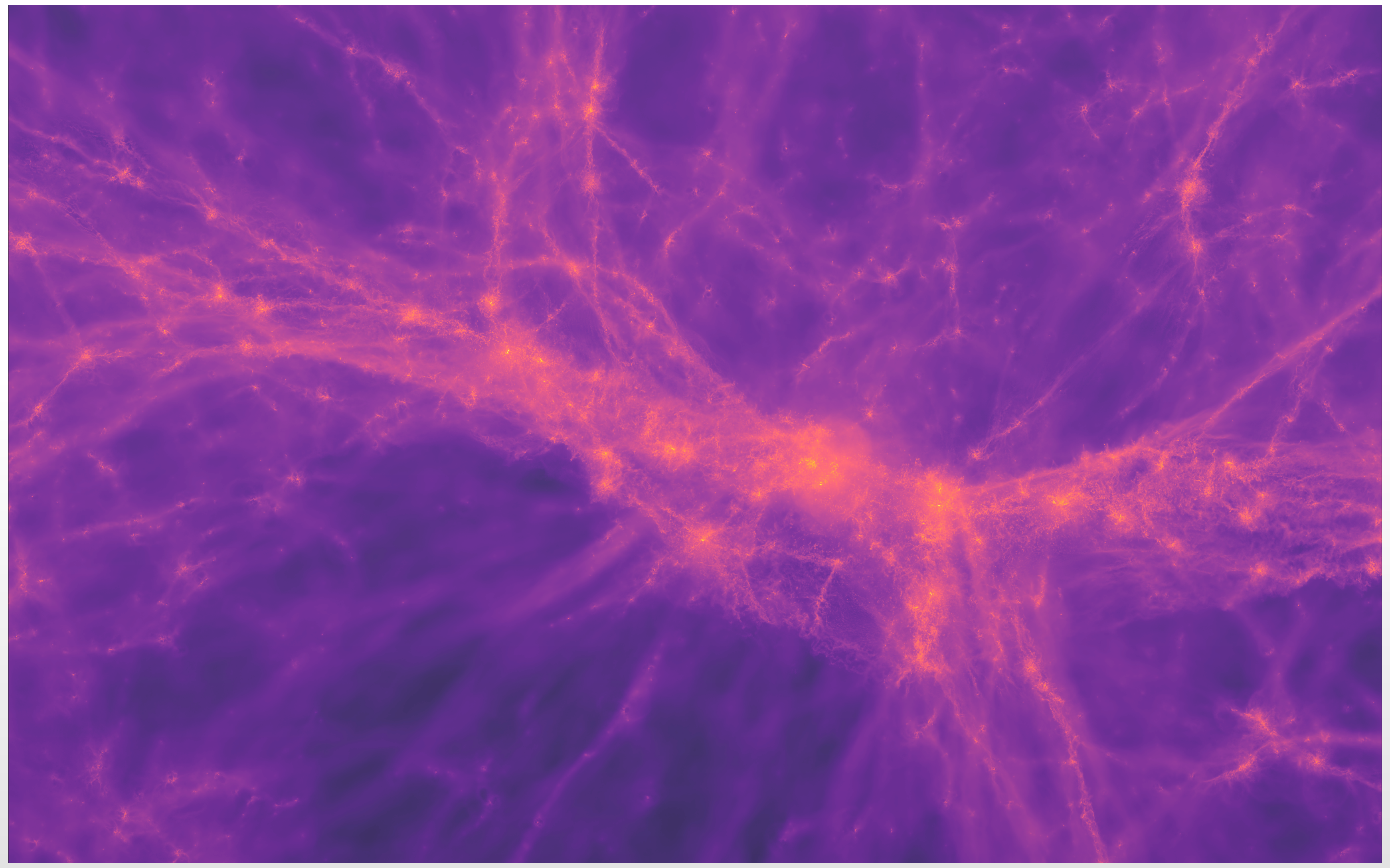
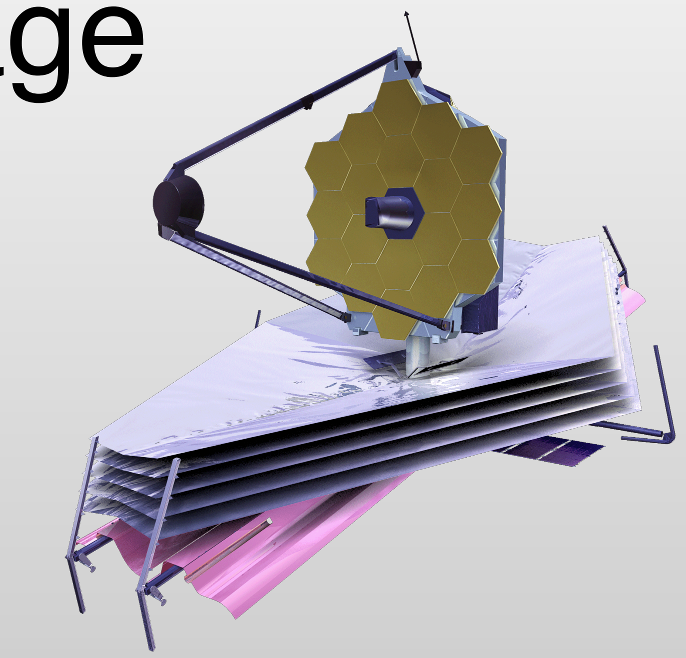


Image Credit: TNG Collaboration

- New observations challenge current models
- Bigger, higher resolution simulations are needed

→ **Time consuming**

Different modules for each stage



EPFL

Modelling the Cosmos Through Simulations

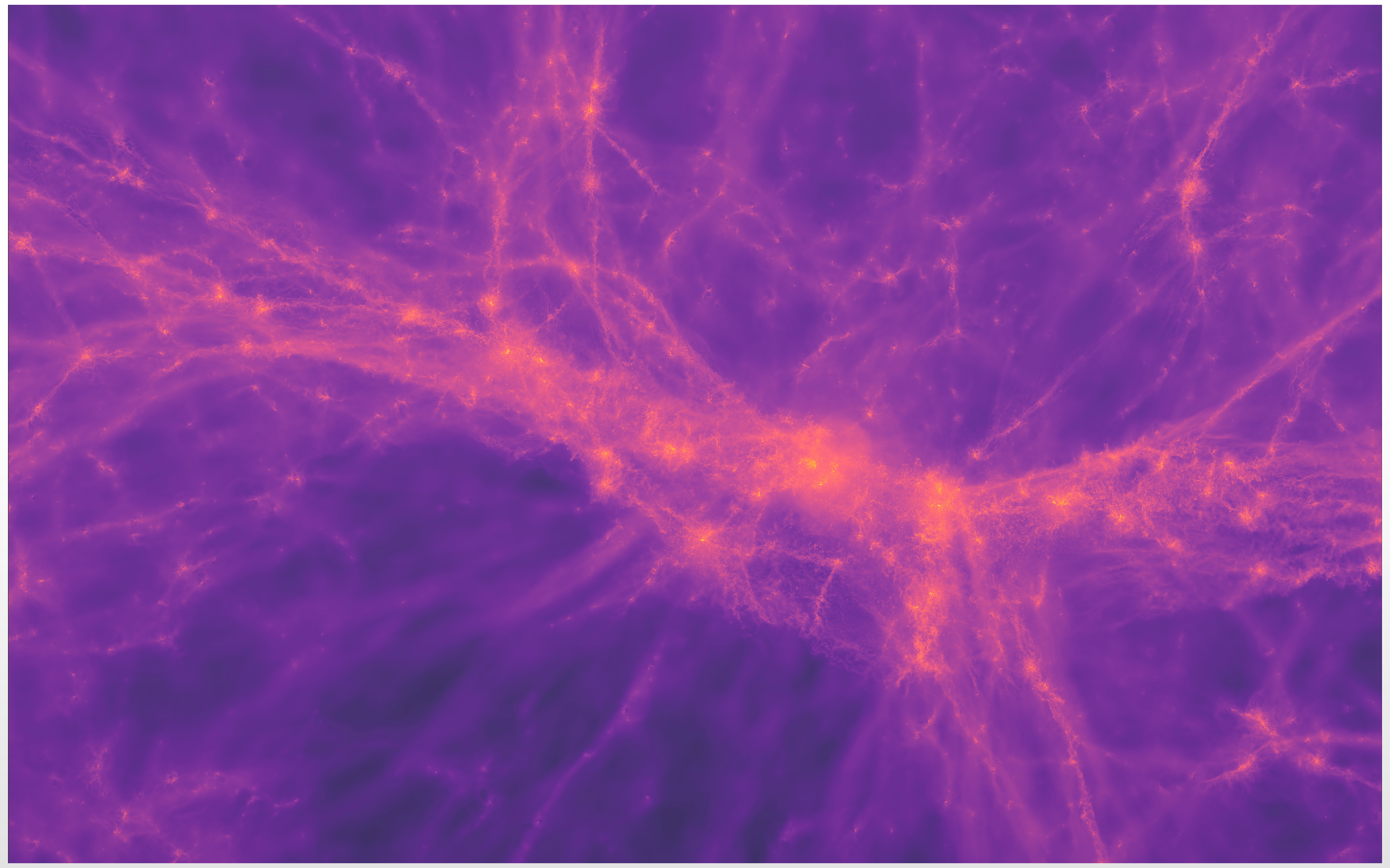


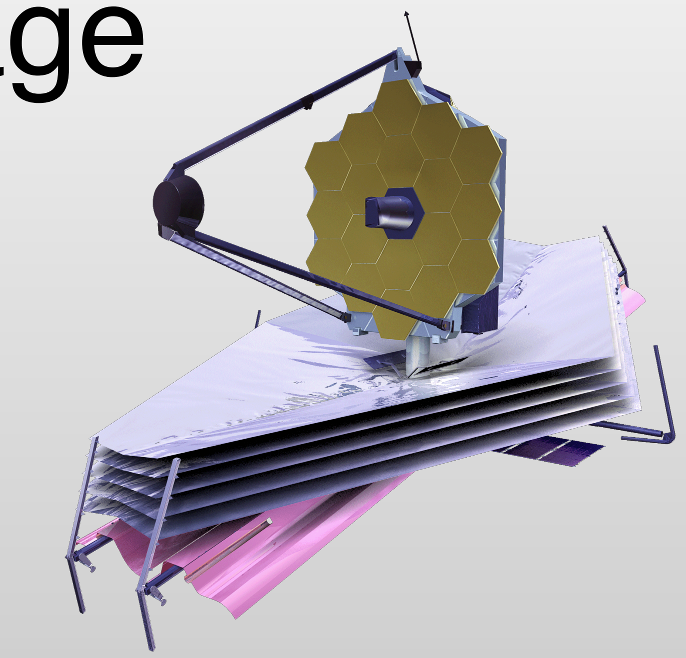
Image Credit: TNG Collaboration

- New observations challenge current models
- Bigger, higher resolution simulations are needed

→ **Time consuming**

Different modules for each stage

Many complex equations

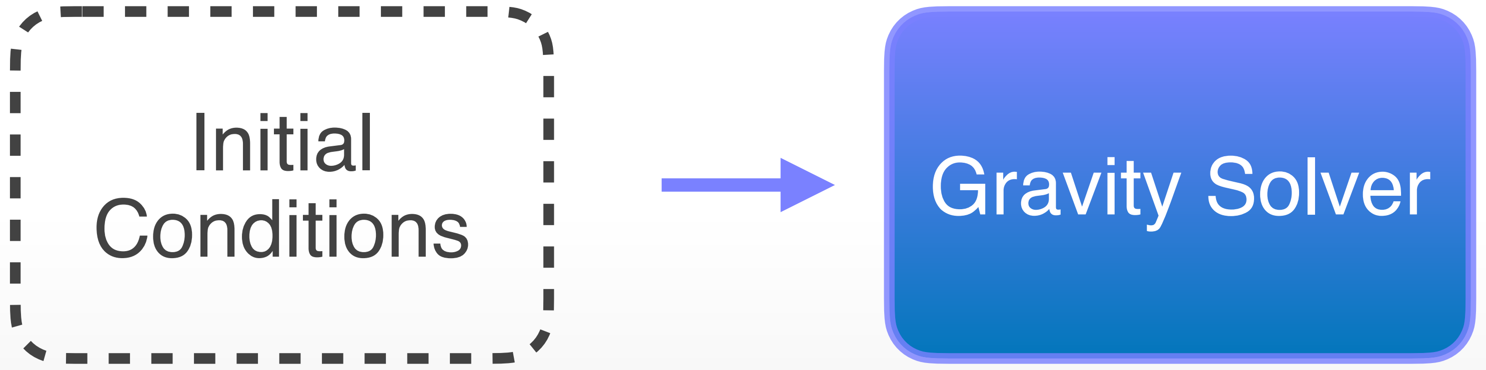


General Workflow of Simulations

General Workflow of Simulations

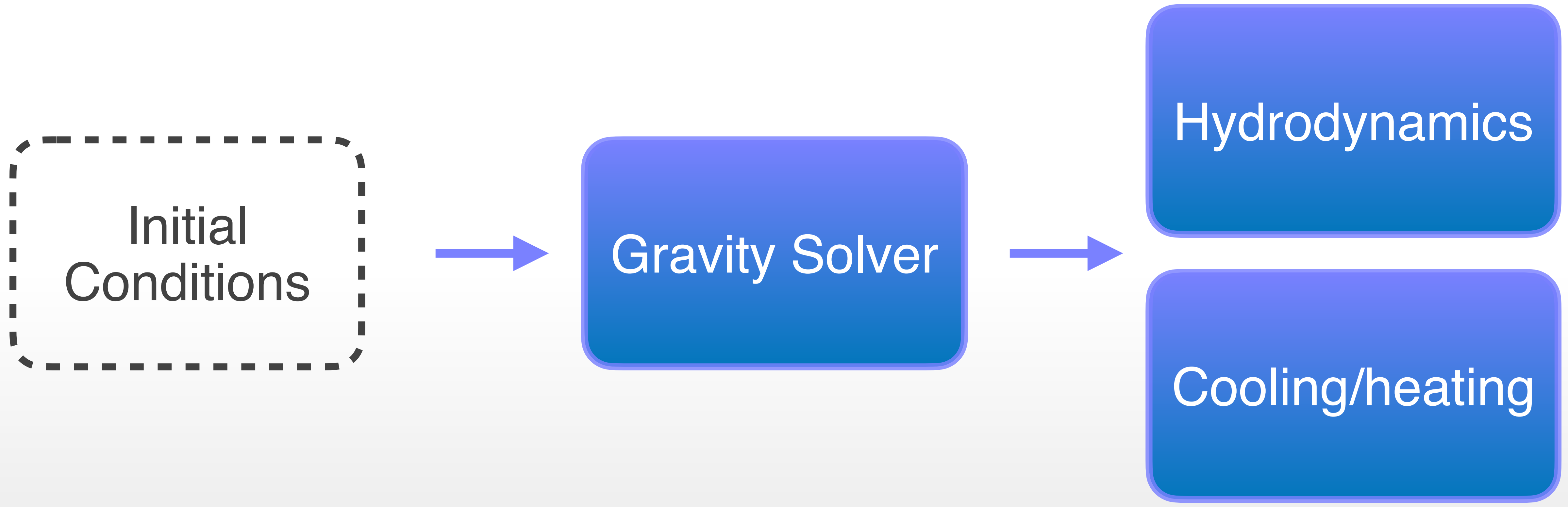
Gravity Solver

General Workflow of Simulations

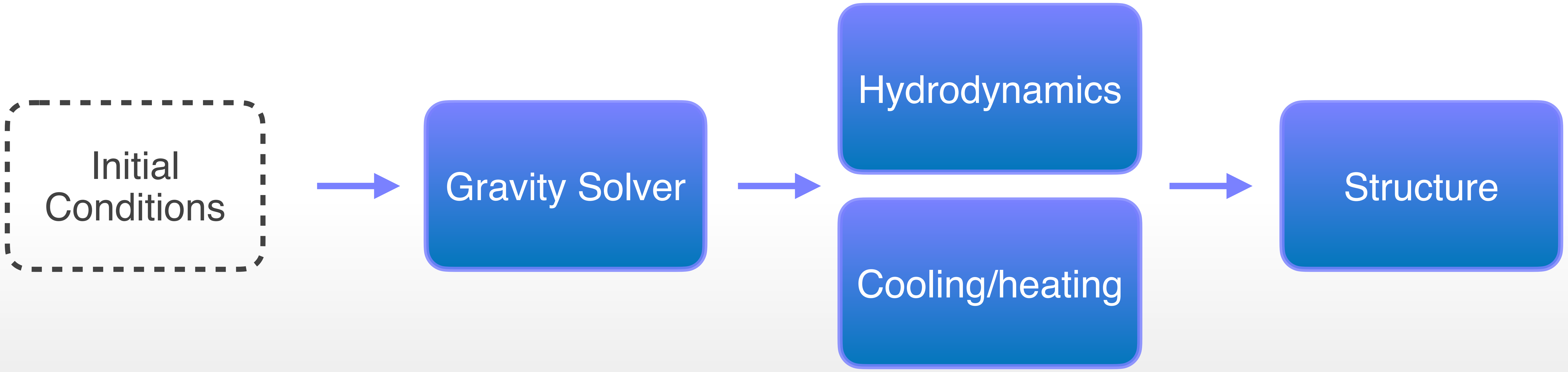


EPFL

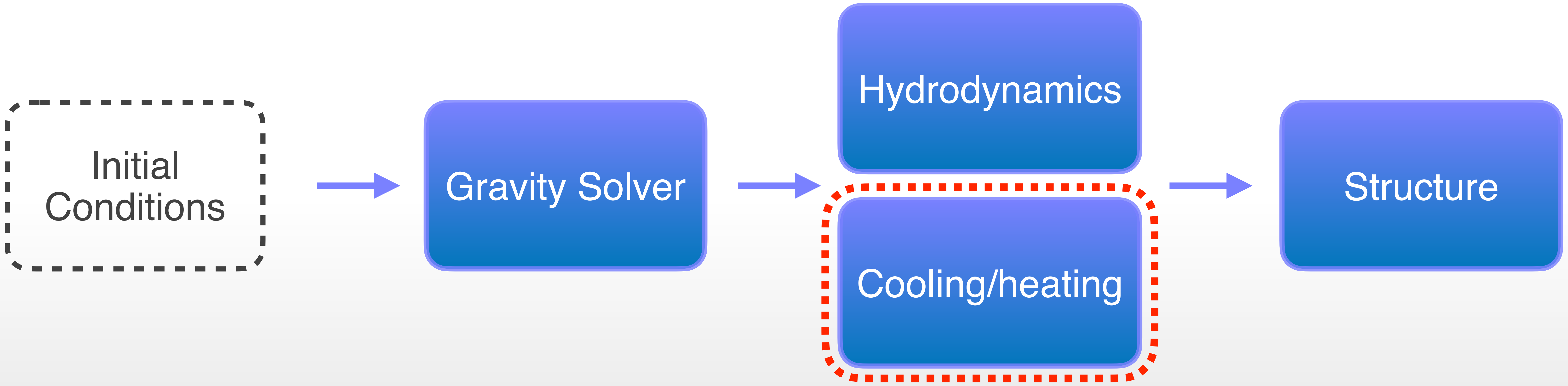
General Workflow of Simulations



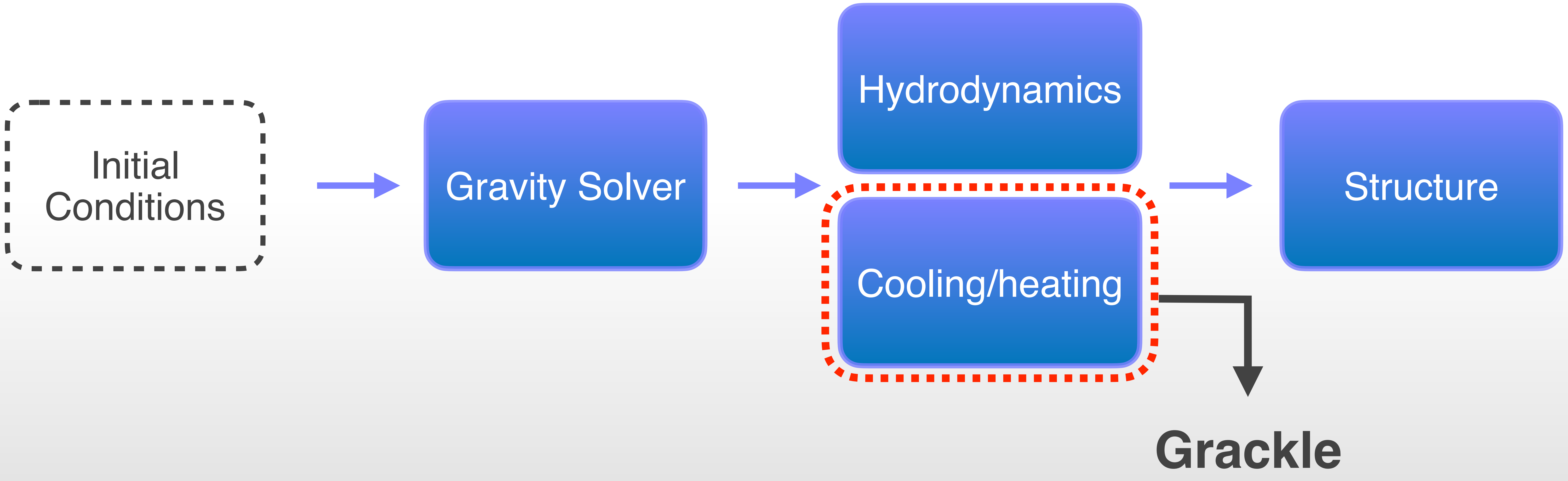
General Workflow of Simulations



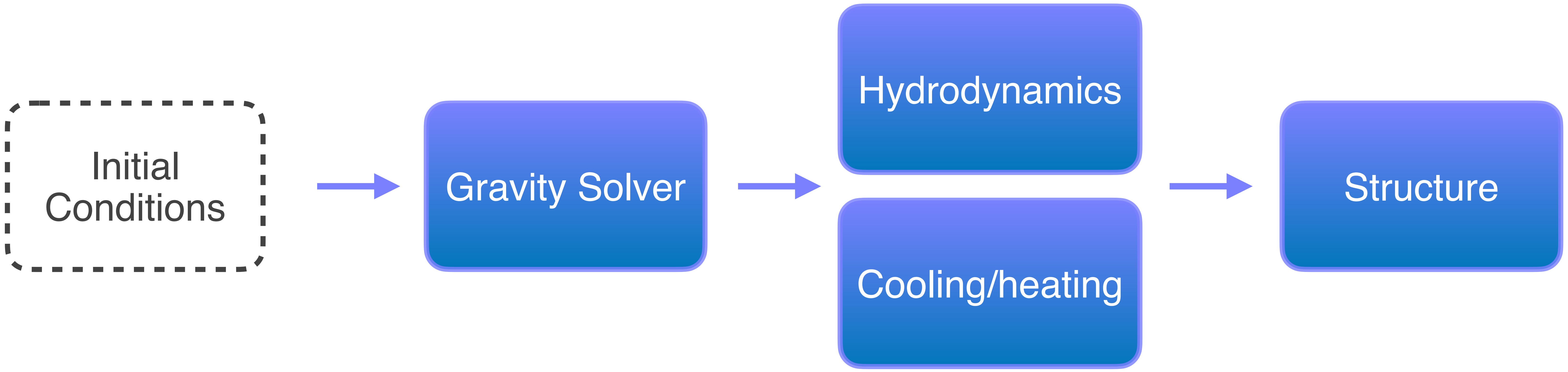
General Workflow of Simulations



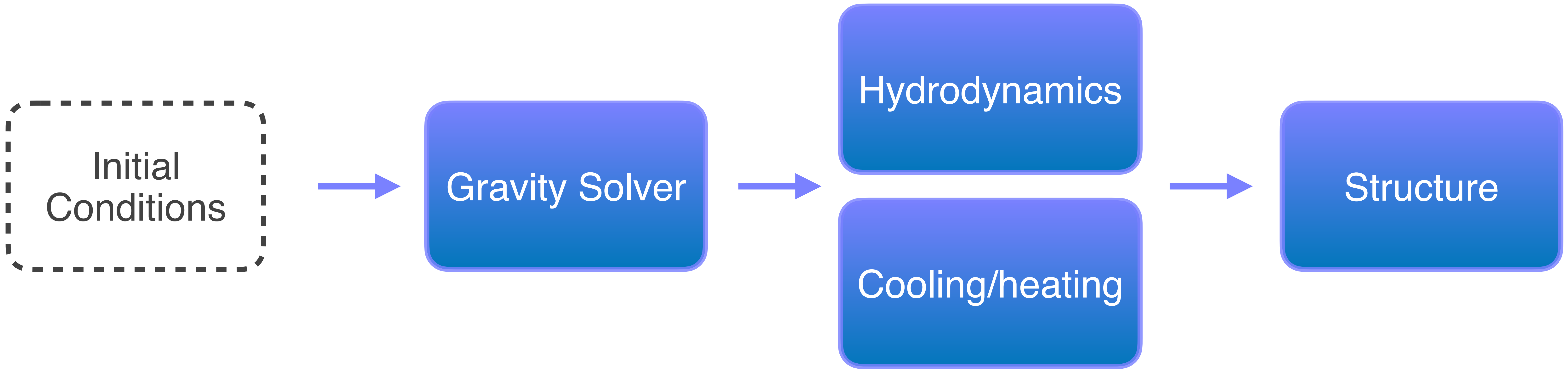
General Workflow of Simulations



EPFL


Initial Conditions
Gravity Solver
Hydrodynamics
Cooling/heating
Structure

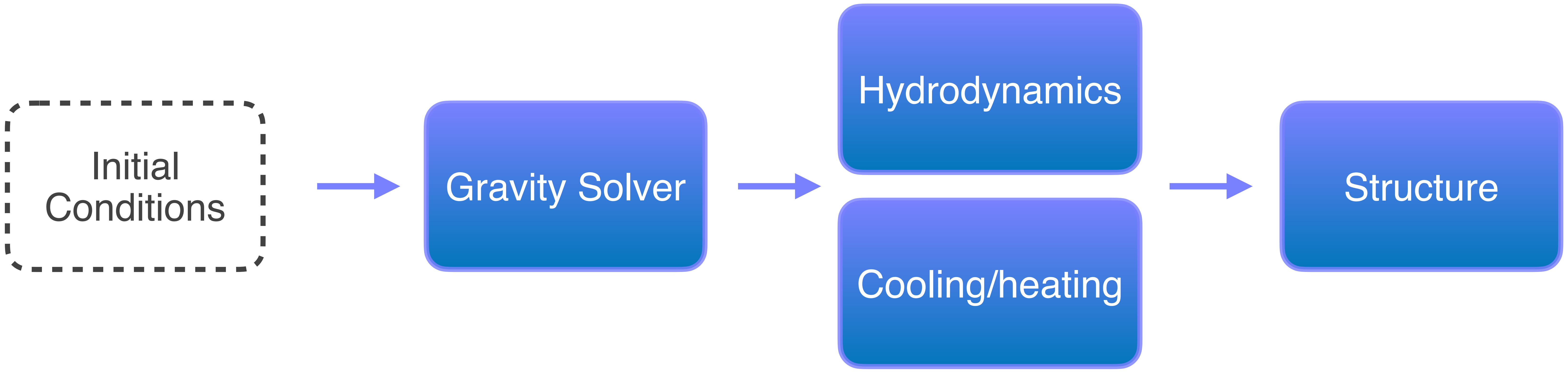
Grackle



Grackle

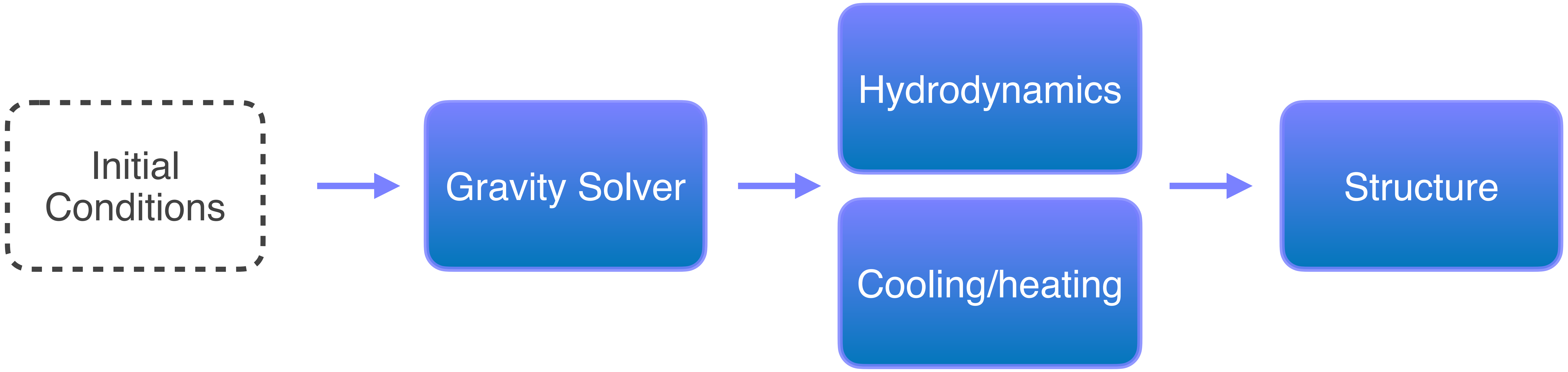
- Chemistry and cooling library

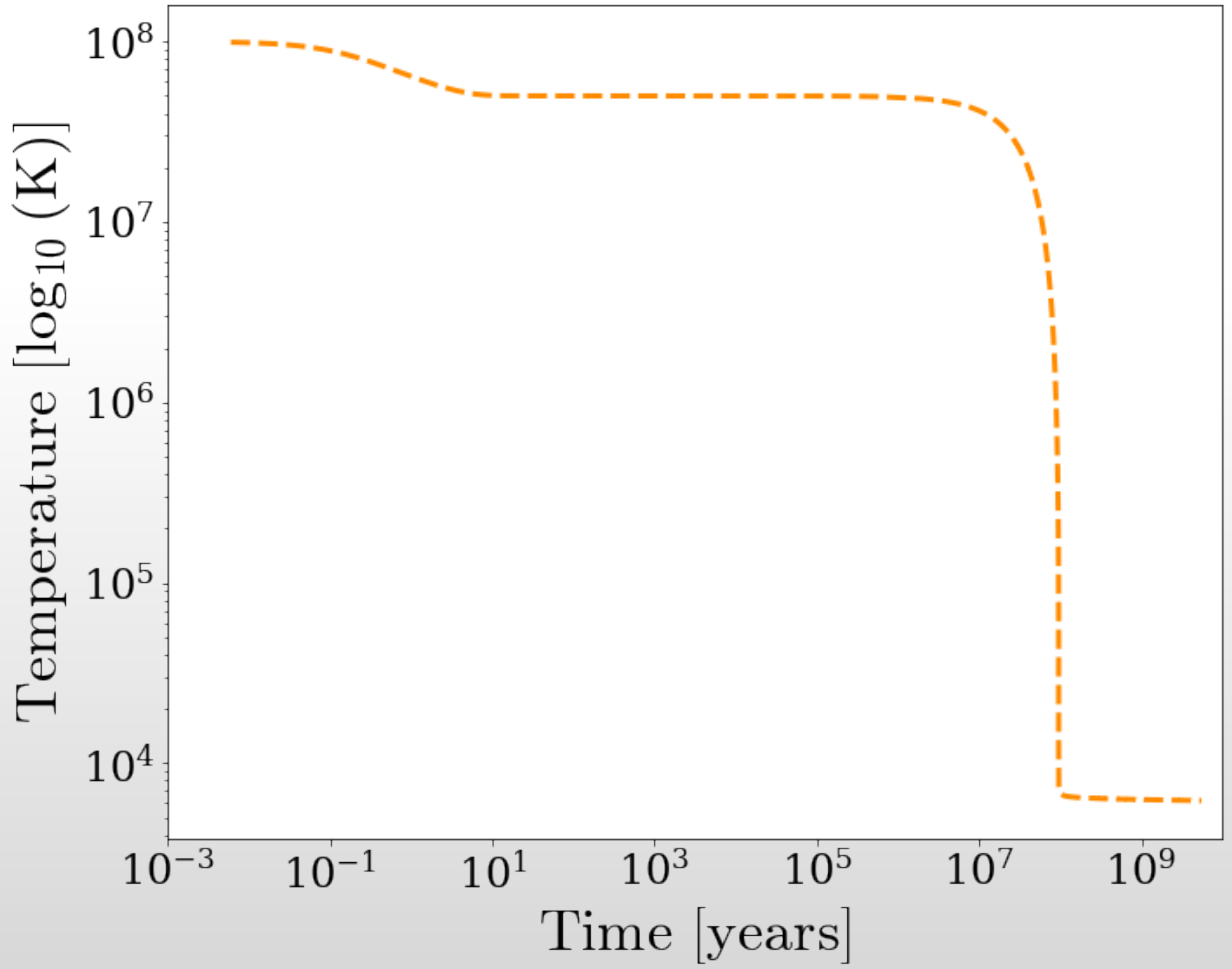
EPFL



Grackle

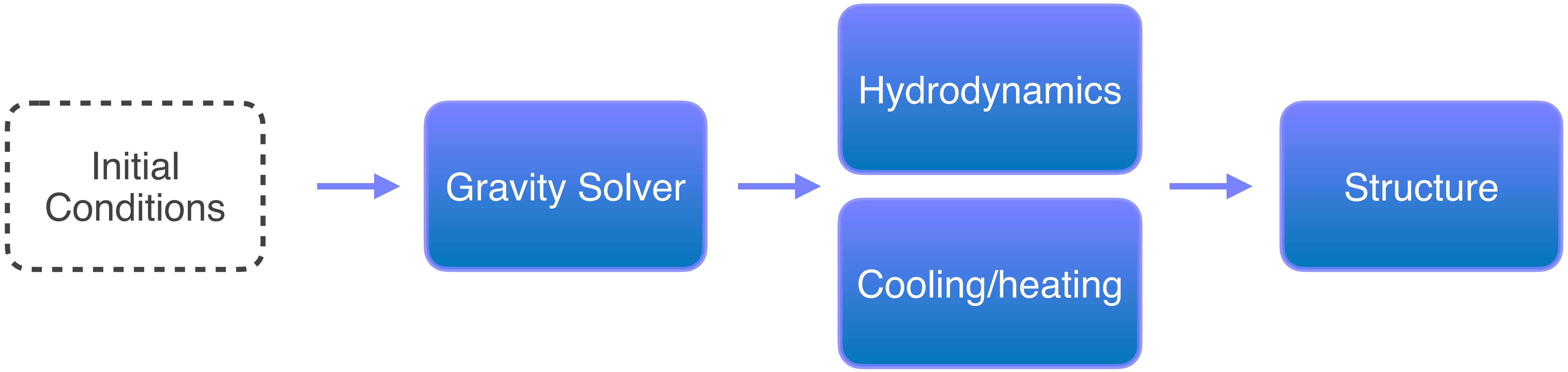
- Chemistry and cooling library
- Initial values for $T,\ \rho_i$

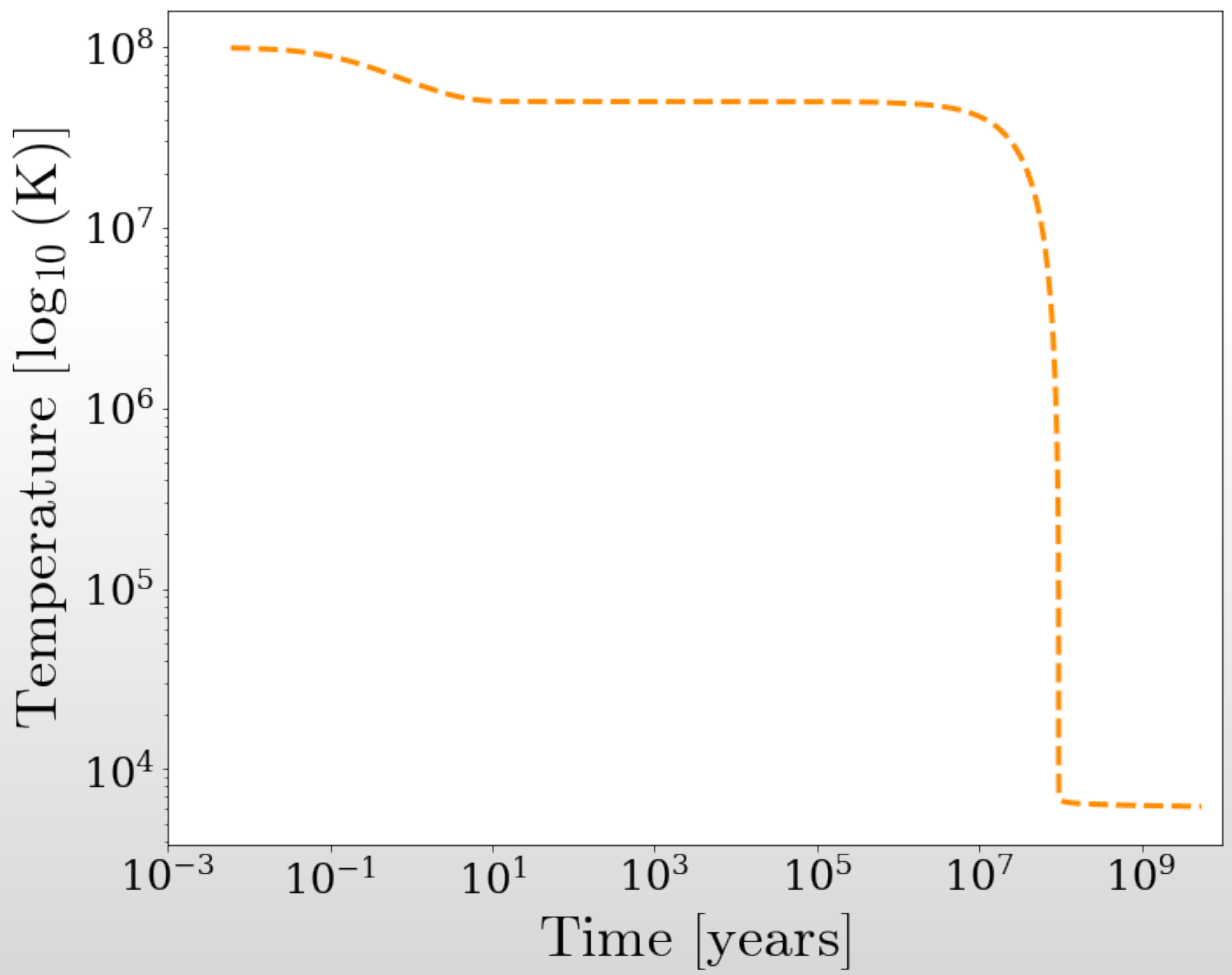


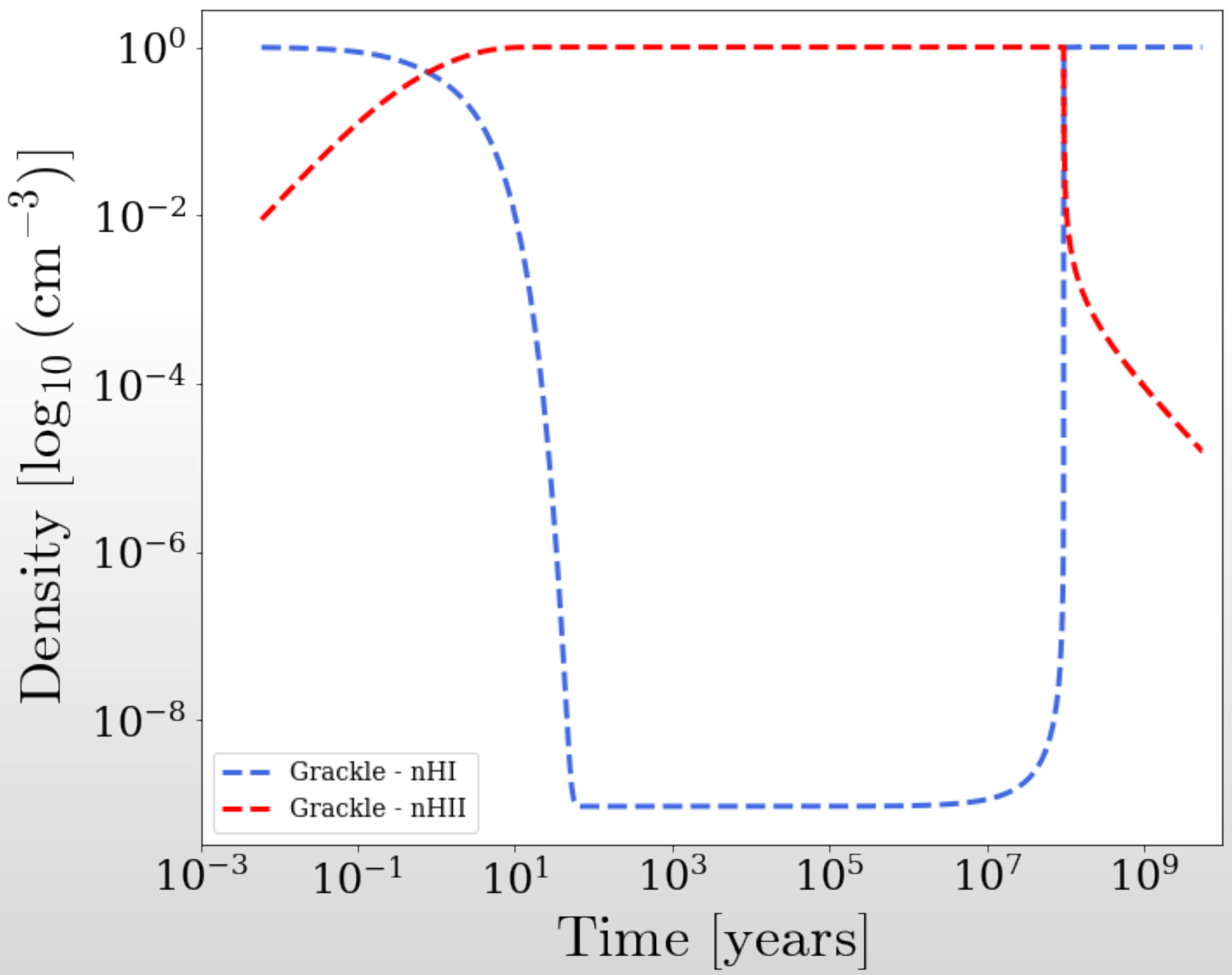


Grackle

- Chemistry and cooling library
- Initial values for $T,\ \rho_i$
- Update T at each iteration



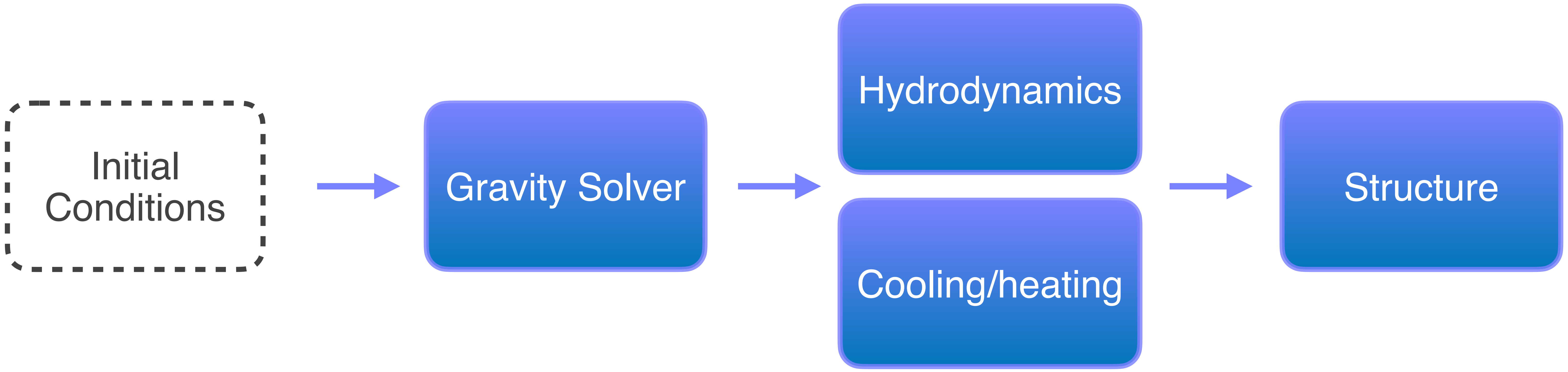


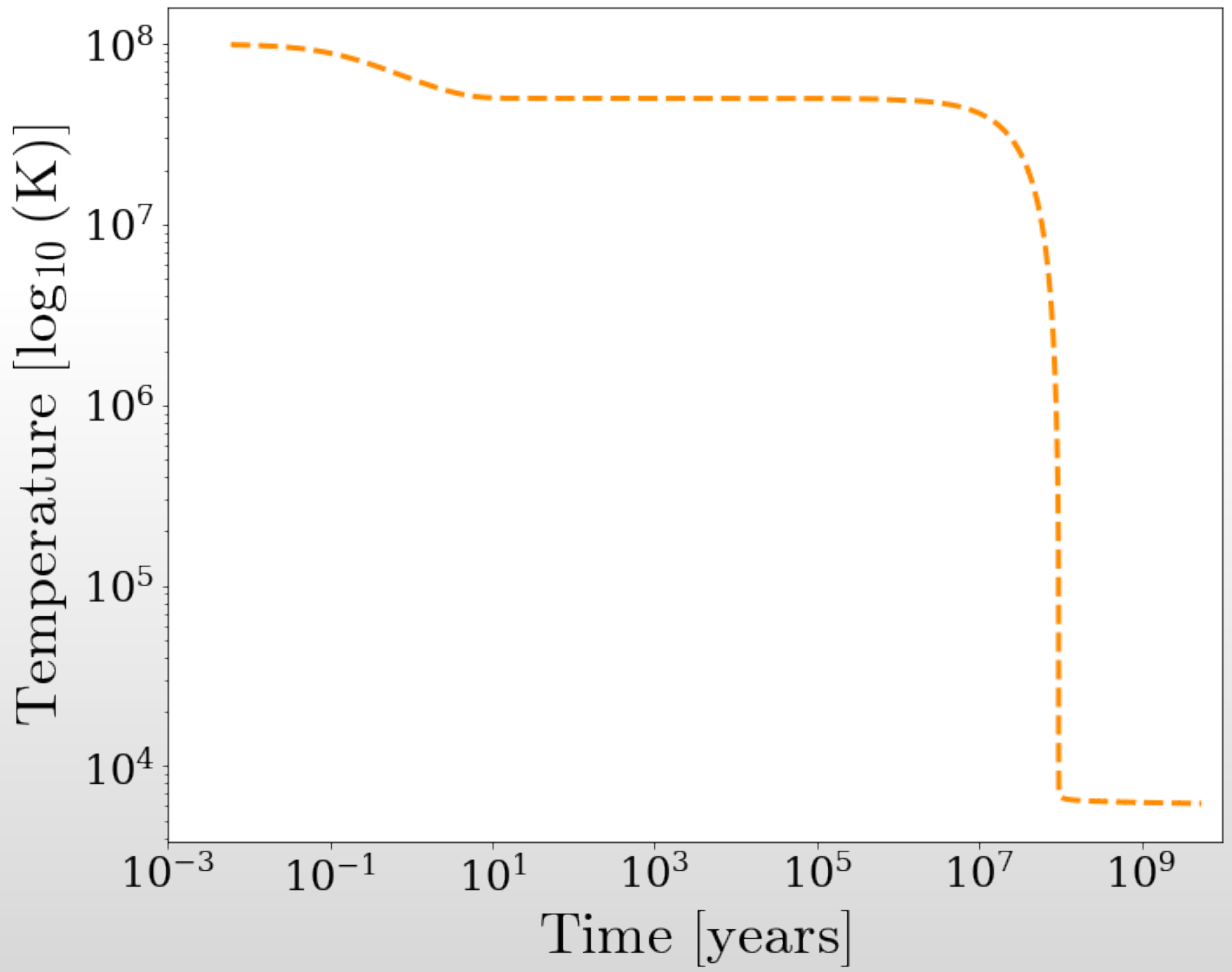


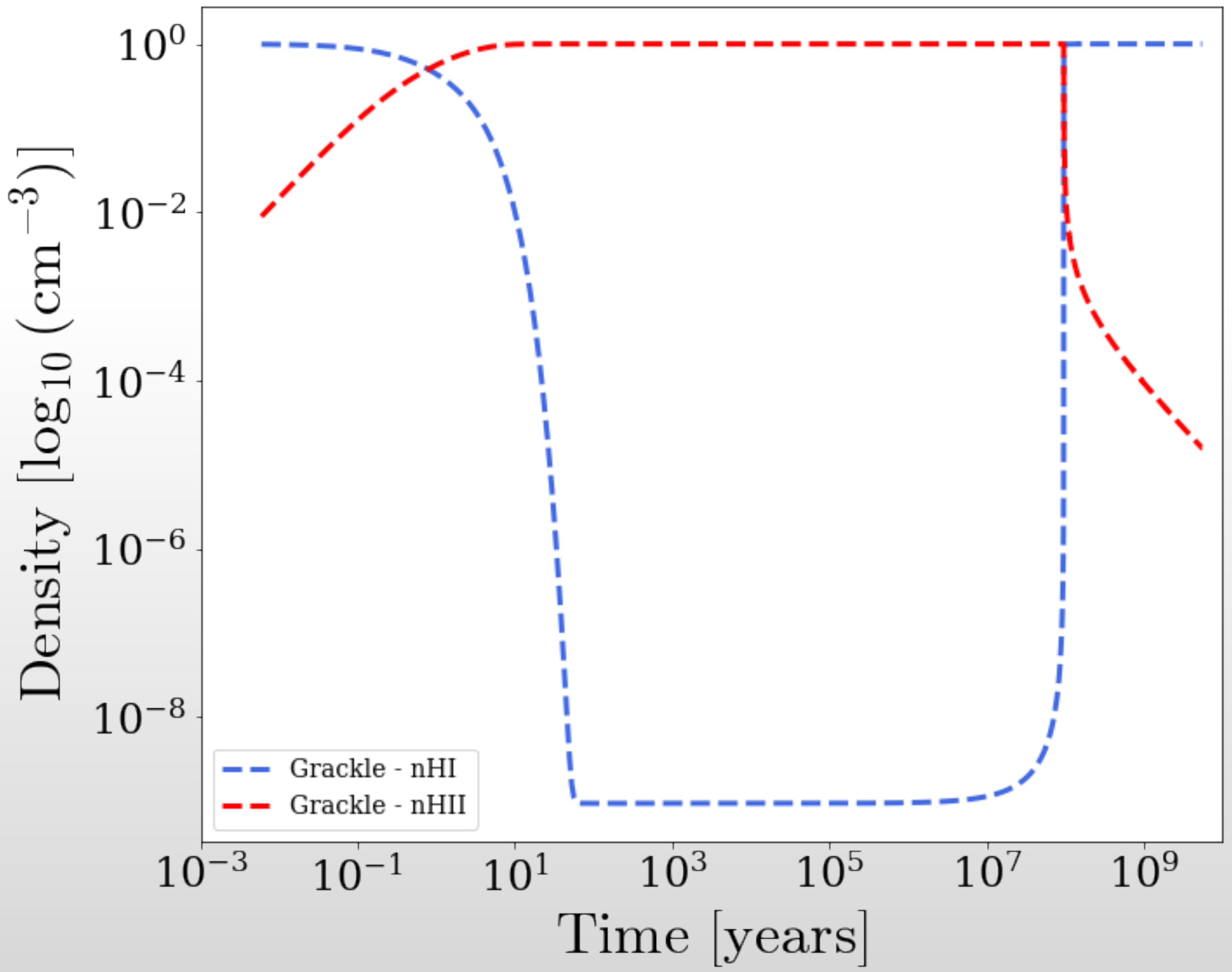
Grackle

- Chemistry and cooling library
- Initial values for $T,\ \rho_i$
- Update T at each iteration
- Update densities through rate equations:

$$\frac{\partial n_i}{\partial t} = \sum_j \sum_l k_{jl}(T) n_j n_l$$





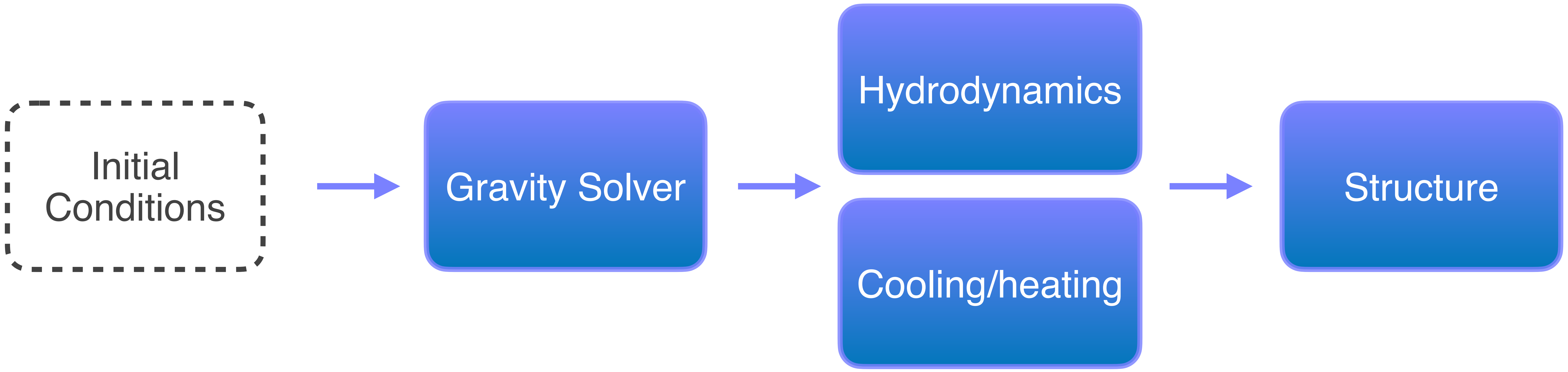


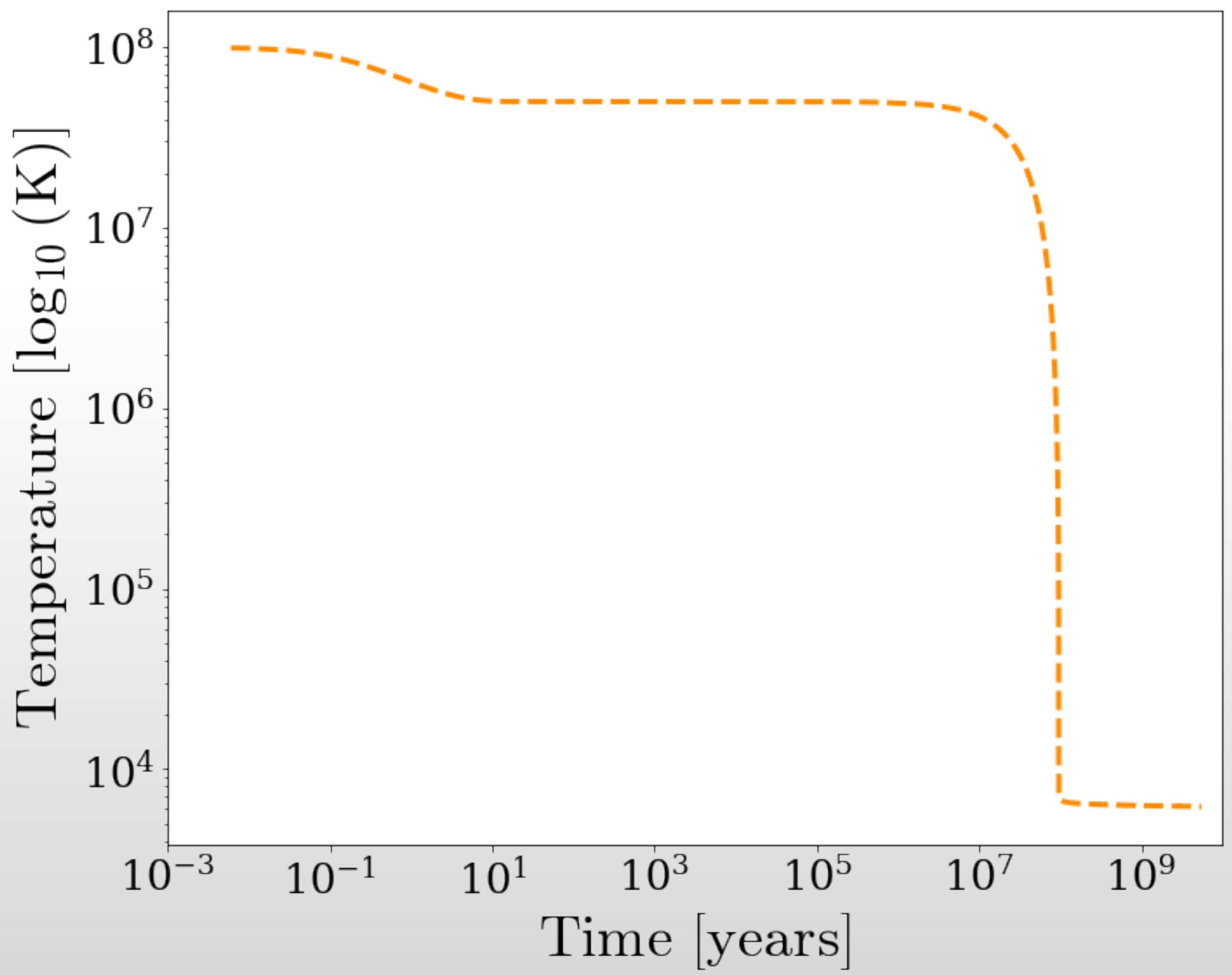
Grackle

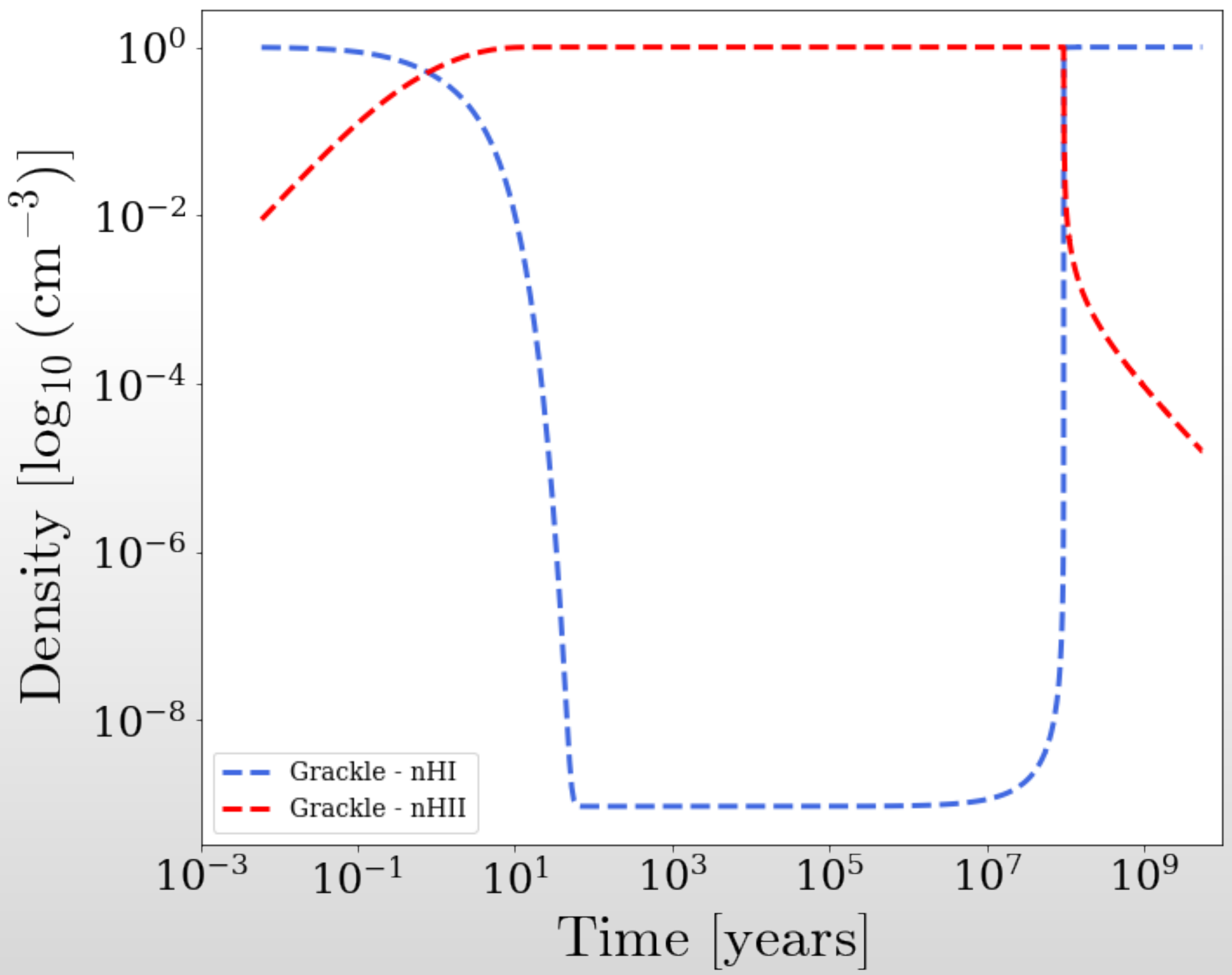
- Chemistry and cooling library
- Initial values for T, ρ_i
- Update T at each iteration
- Update densities through rate equations:

$$\frac{\partial n_i}{\partial t} = \sum_j \sum_l k_{jl}(T) n_j n_l + \sum_j I_j n_l$$

EPFL



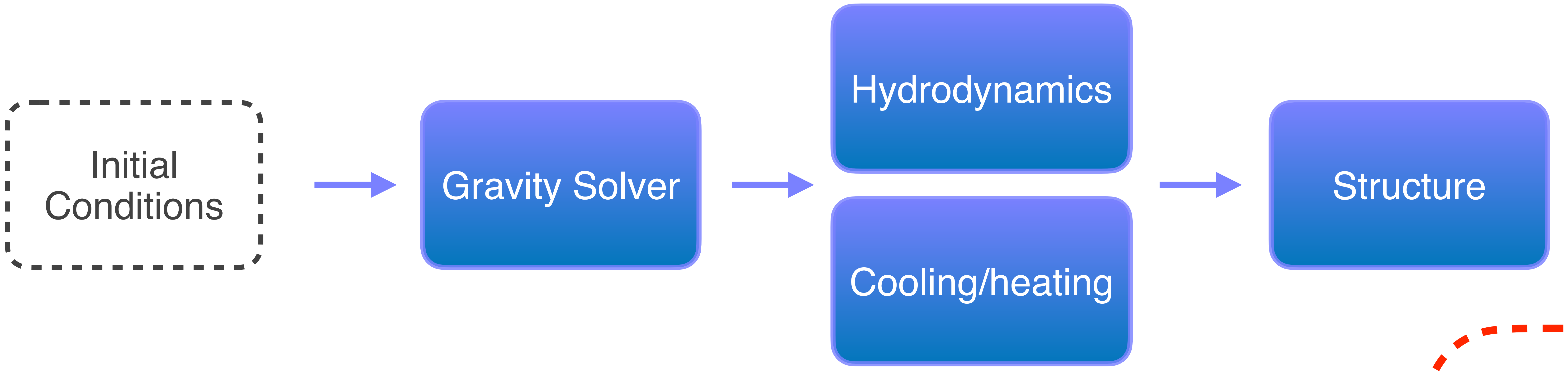


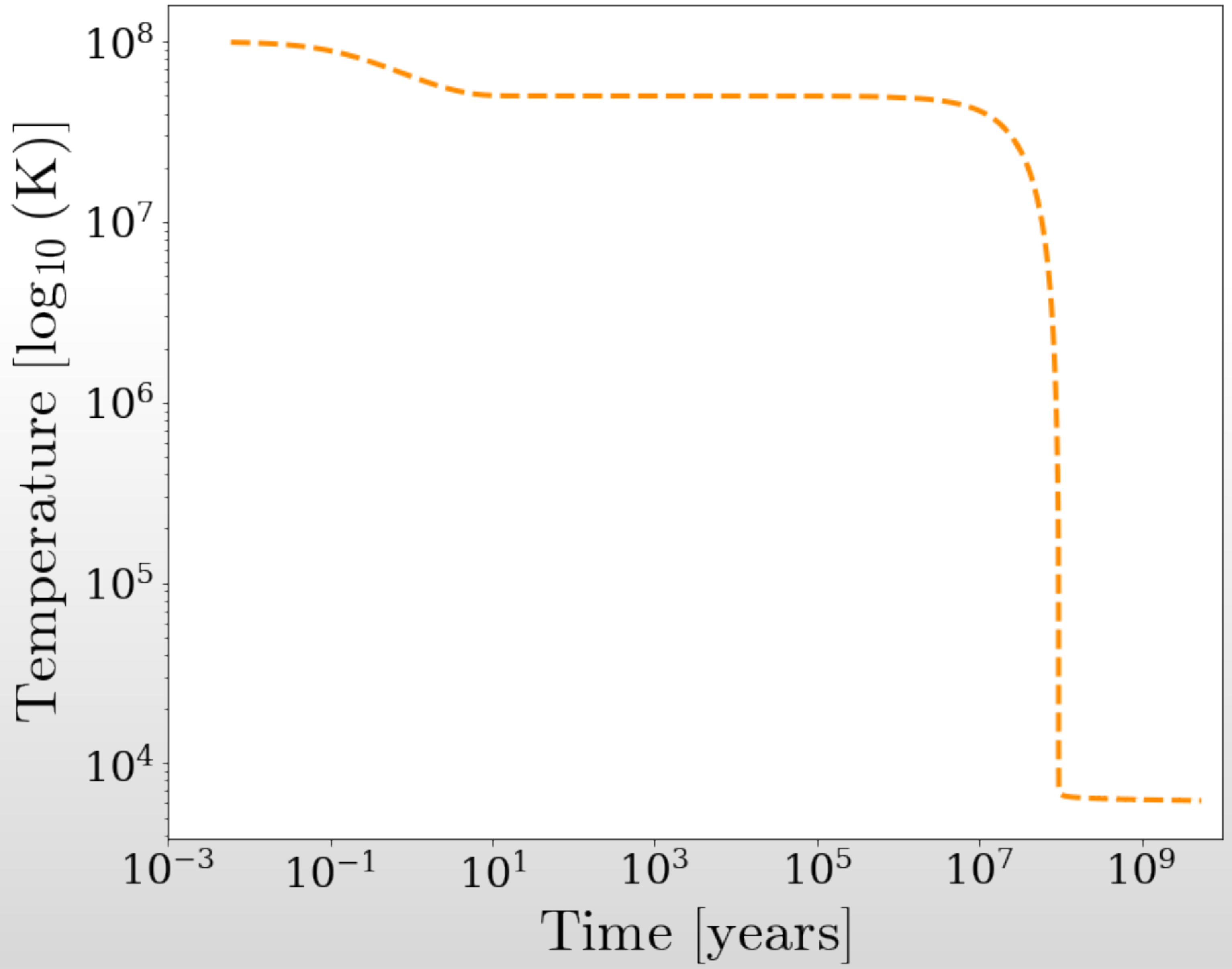


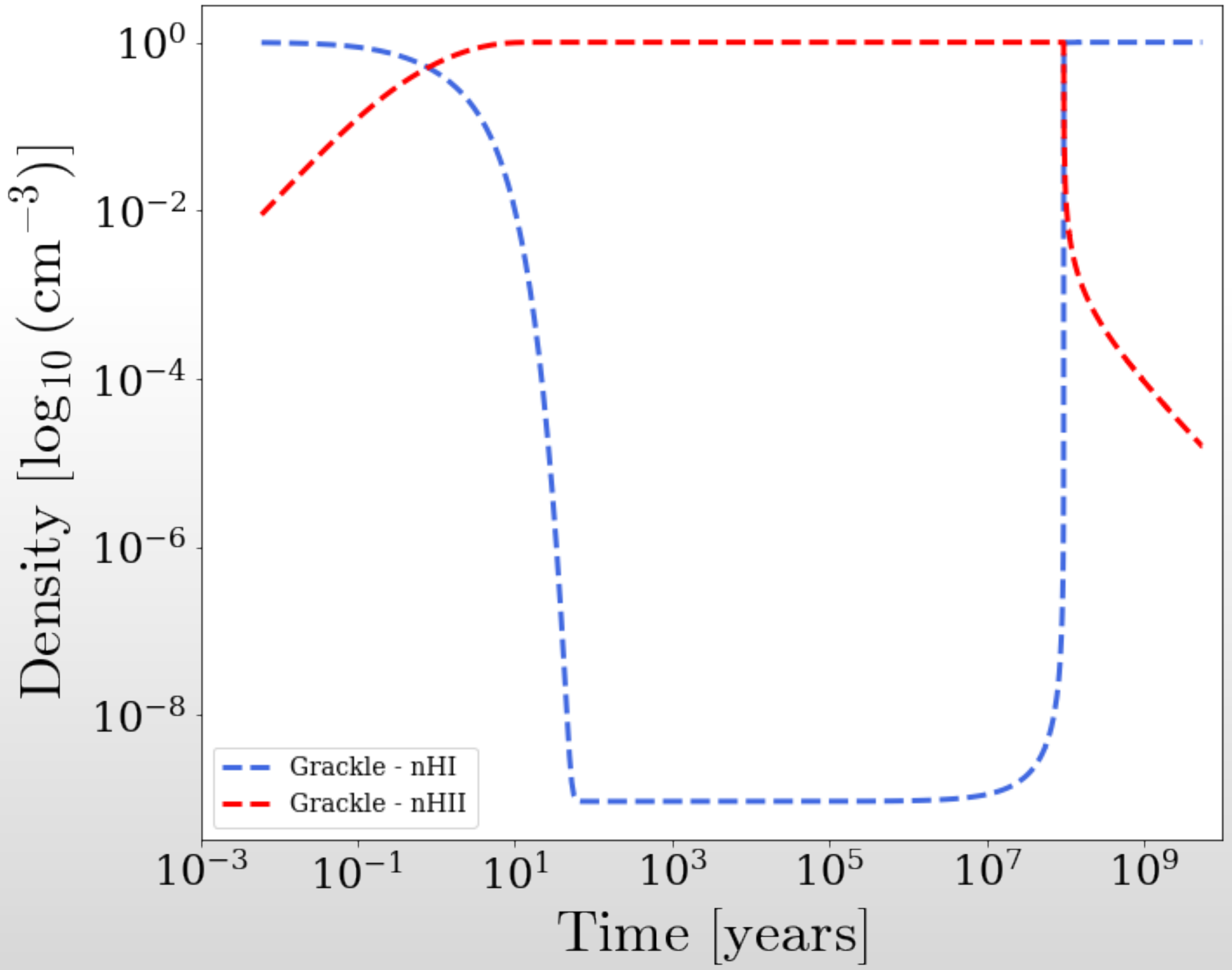
Grackle

- Chemistry and cooling library
- Initial values for T, ρ_i
- Update T at each iteration
- Update densities through rate equations:

$$\frac{\partial n_i}{\partial t} = \sum_j \sum_l k_{jl}(T) n_j n_l + \cancel{\sum_j I_j n_l}$$







Grackle

- Chemistry and cooling library
- Initial values for T, ρ_i
- Update T at each iteration
- Update densities through rate equations:

$$\frac{\partial n_i}{\partial t} = \sum_j \sum_l k_{jl}(T) n_j n_l + \cancel{\sum_j I_j n_l}$$

Time Consuming!

Grackle

- Chemistry and cooling library
- Initial values for $T,\ \rho_i$
- Update T at each iteration
- Update densities through rate equations:

$$\frac{\partial n_i}{\partial t} = \sum_j \sum_l k_{jl}(T) n_j n_l + \cancel{\sum_j I_j n_l}$$

Time Consuming!

EPFL

Grackle

- Chemistry and cooling library
- Initial values for $T,\ \rho_i$
- Update T at each iteration
- Update densities through rate equations:

$$\frac{\partial n_i}{\partial t} = \sum_j \sum_l k_{jl}(T) n_j n_l + \cancel{\sum_j I_j n_l}$$

Time Consuming!

Emulate using

Grackle

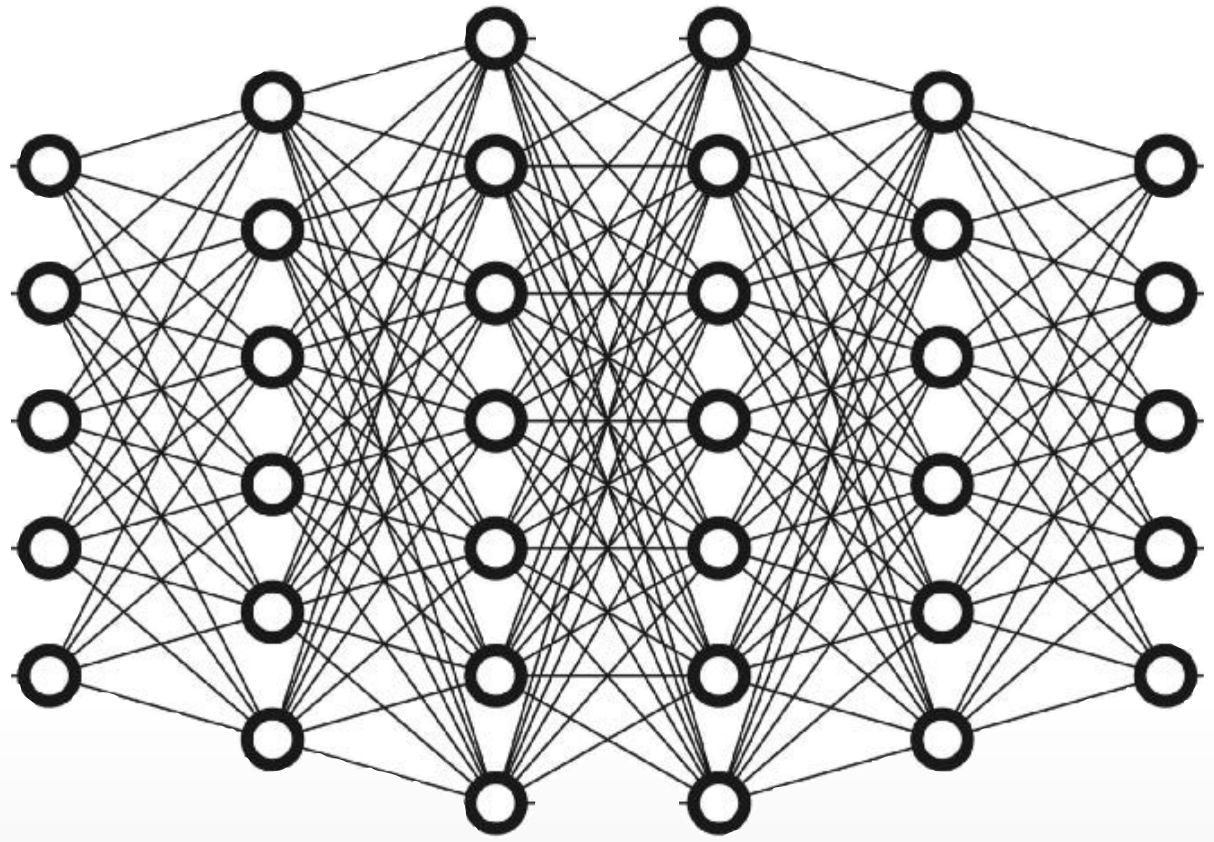
- Chemistry and cooling library
- Initial values for T, ρ_i
- Update T at each iteration
- Update densities through rate equations:

$$\frac{\partial n_i}{\partial t} = \sum_j \sum_l k_{jl}(T) n_j n_l + \cancel{\sum_j I_j n_l}$$

Time Consuming!

Emulate using

Neural Nets



EPFL

Grackle

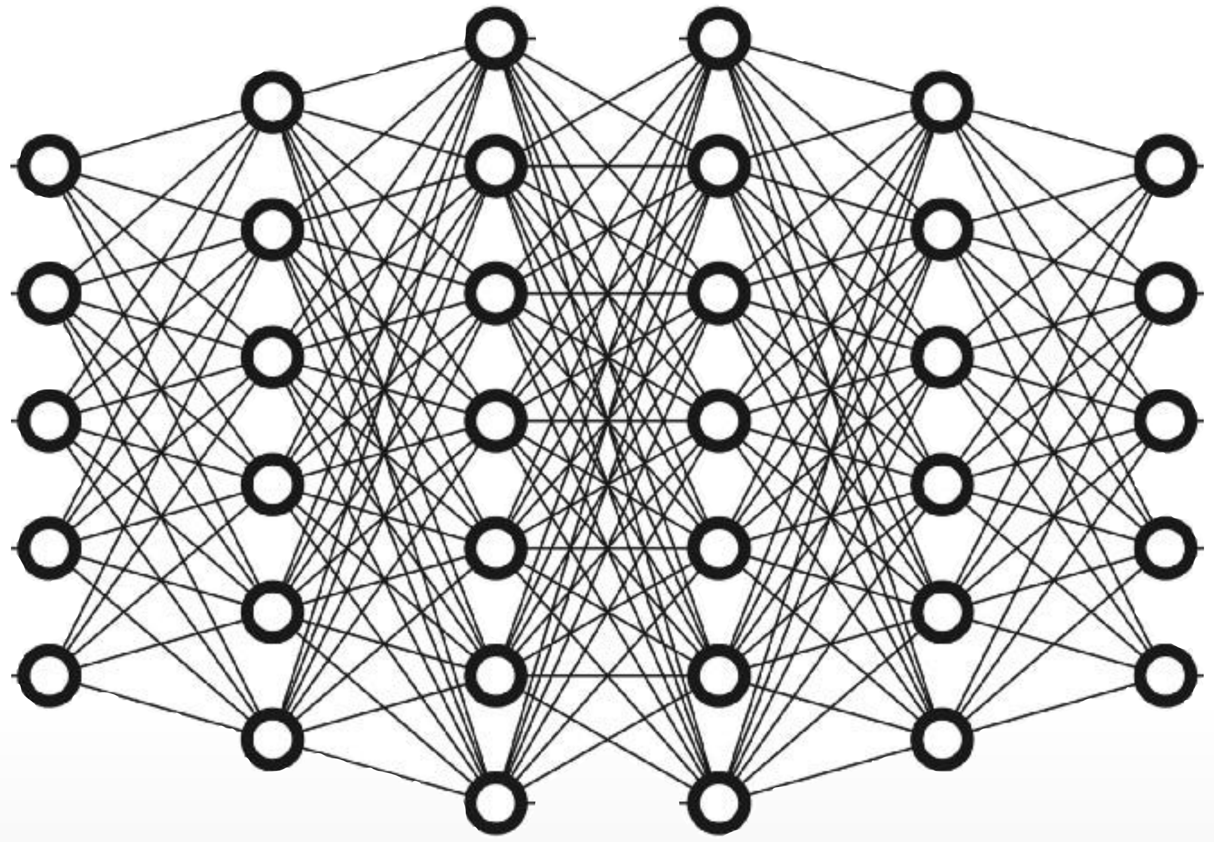
- Chemistry and cooling library
- Initial values for T, ρ_i
- Update T at each iteration
- Update densities through rate equations:

$$\frac{\partial n_i}{\partial t} = \sum_j \sum_l k_{jl}(T) n_j n_l + \cancel{\sum_j I_j n_l}$$

Time Consuming!

Emulate using

Neural Nets



- Simple architecture

Grackle

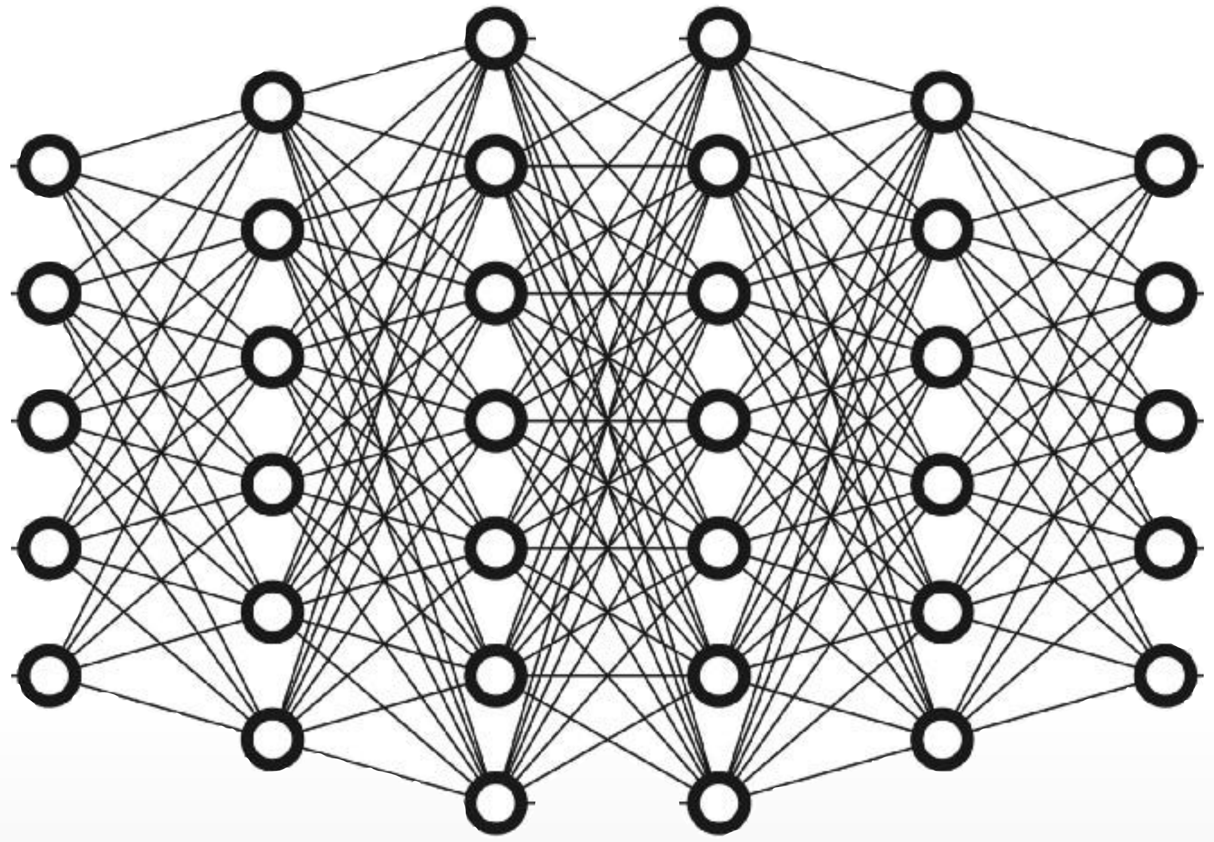
- Chemistry and cooling library
- Initial values for $T,\ \rho_i$
- Update T at each iteration
- Update densities through rate equations:

$$\frac{\partial n_i}{\partial t} = \sum_j \sum_l k_{jl}(T) n_j n_l + \cancel{\sum_j I_j n_l}$$

Time Consuming!

Emulate using

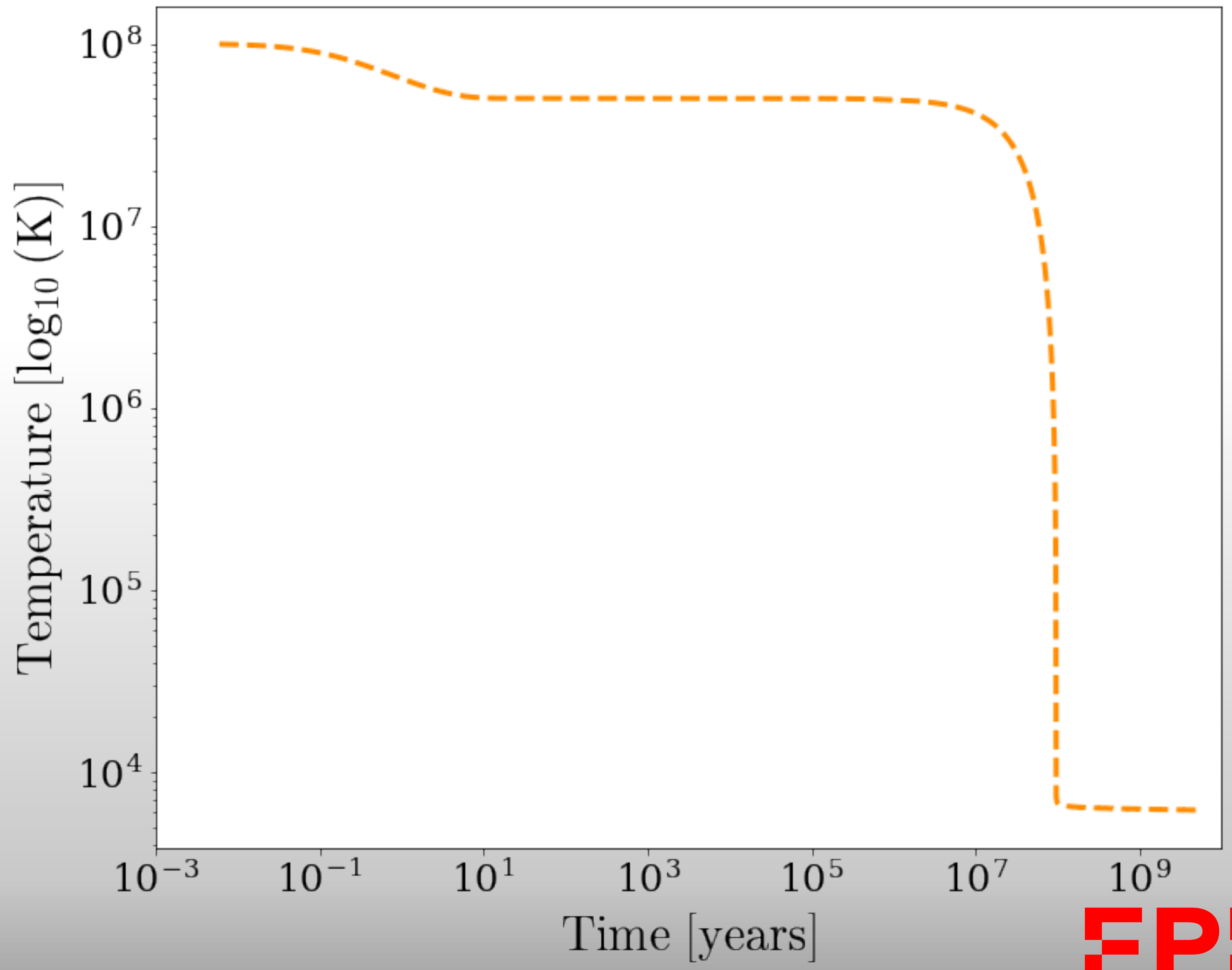
Neural Nets



- Simple architecture
- Train with data from Grackle

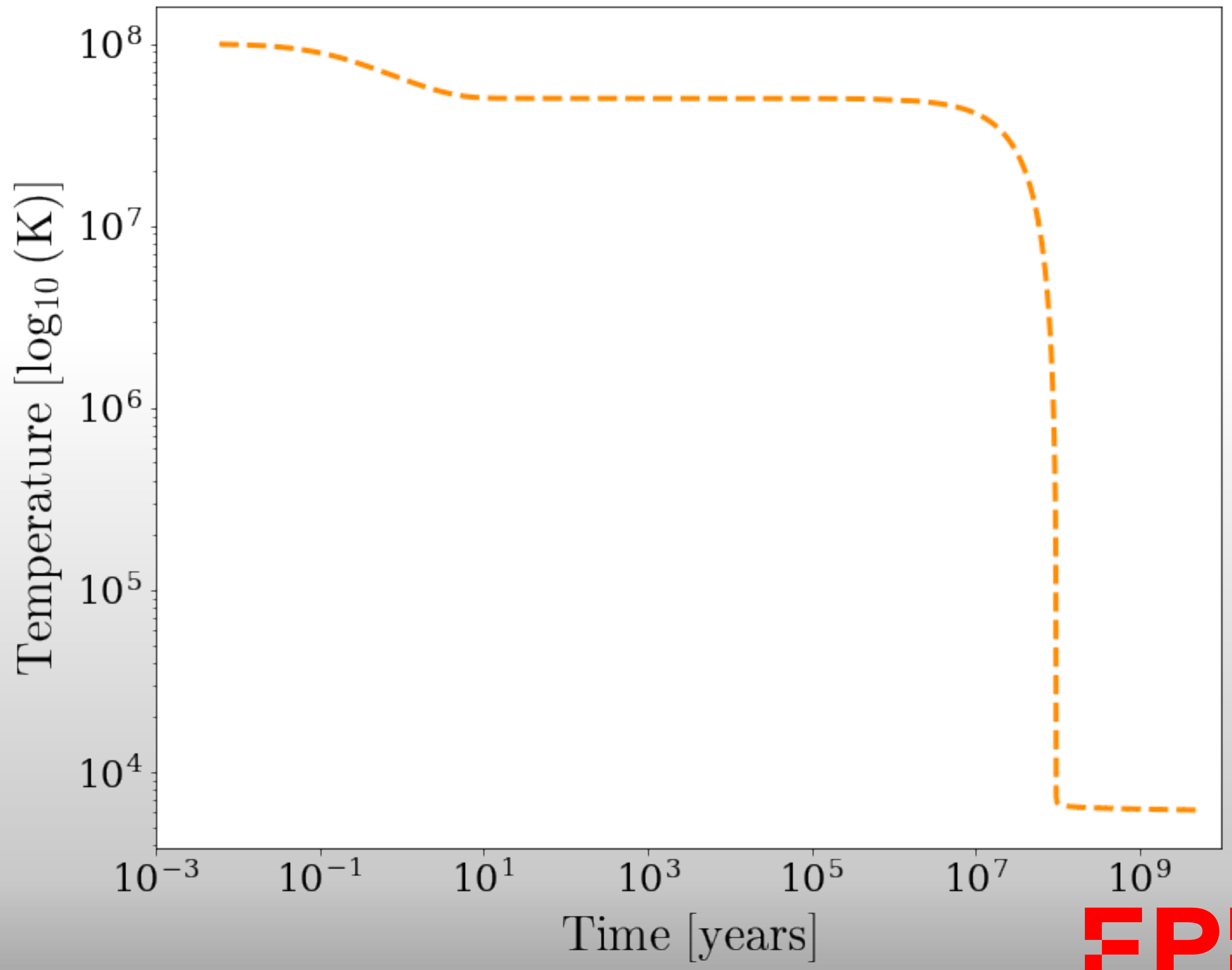
Data Generation & Training

Data Generation & Training



Data Generation & Training

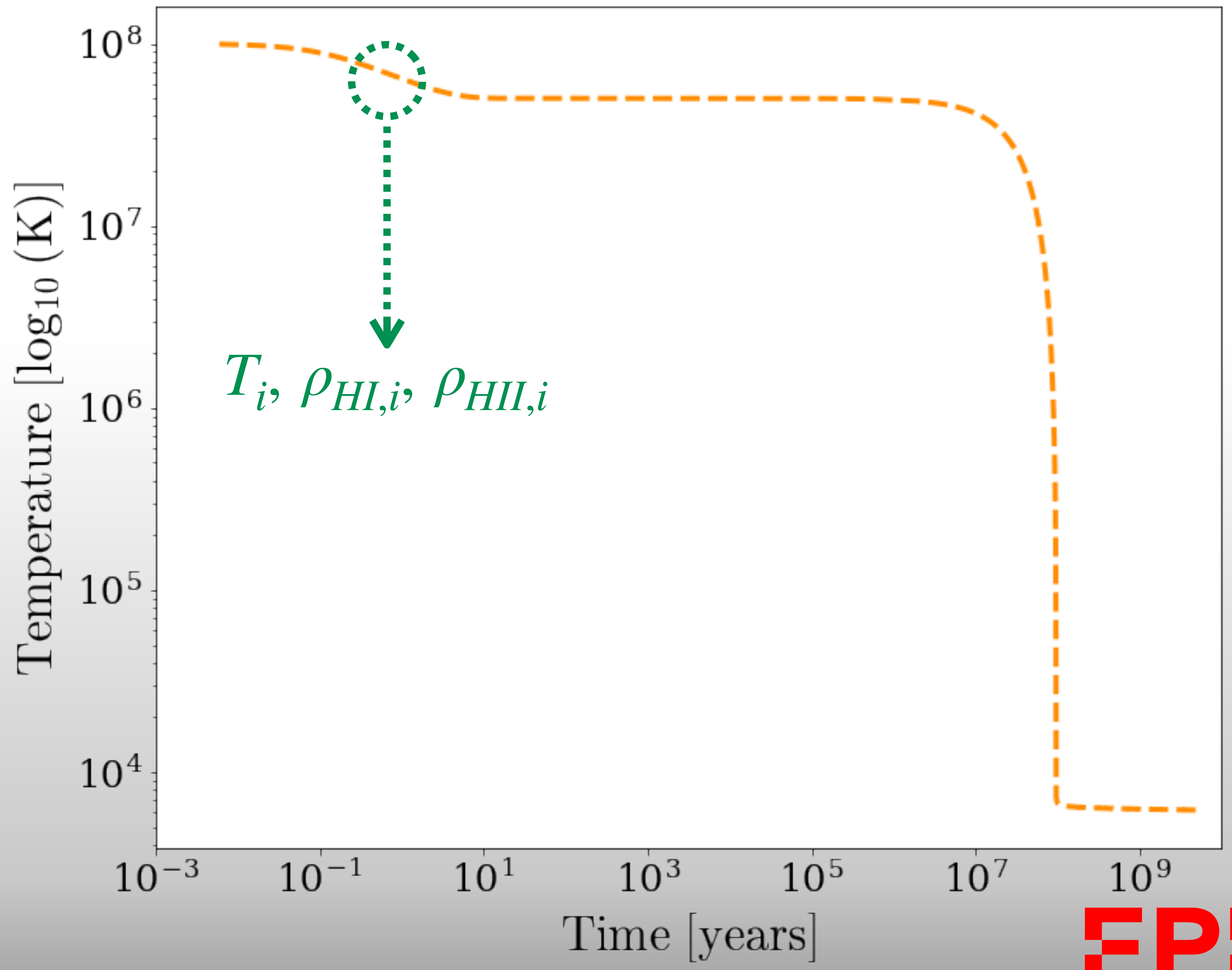
Sample:
T_i
$\rho_{HI,i}$
$\rho_{HII,i}$
dt_i



EPFL

Data Generation & Training

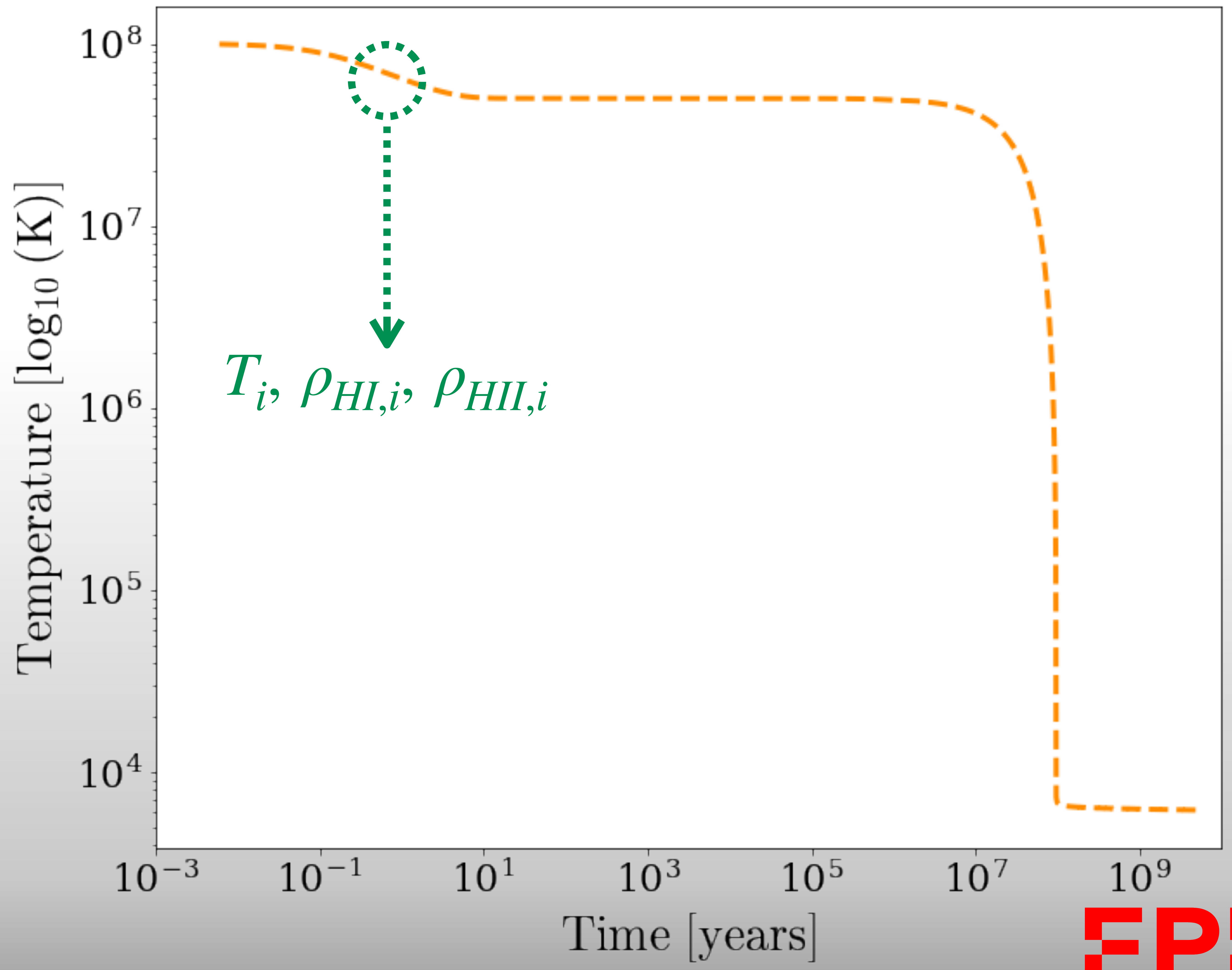
Sample:
T_i
$\rho_{HI,i}$
$\rho_{HII,i}$
dt_i



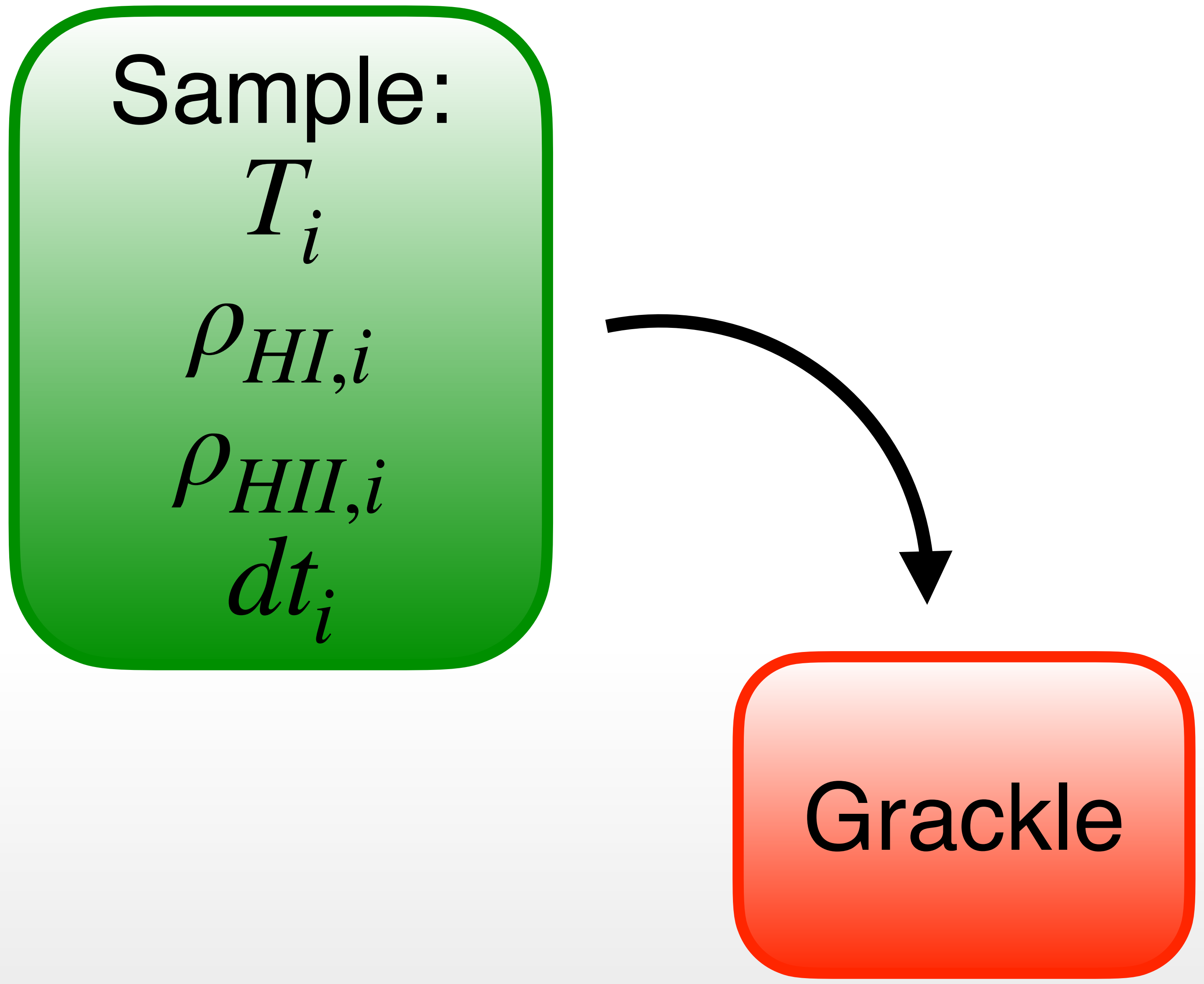
Data Generation & Training

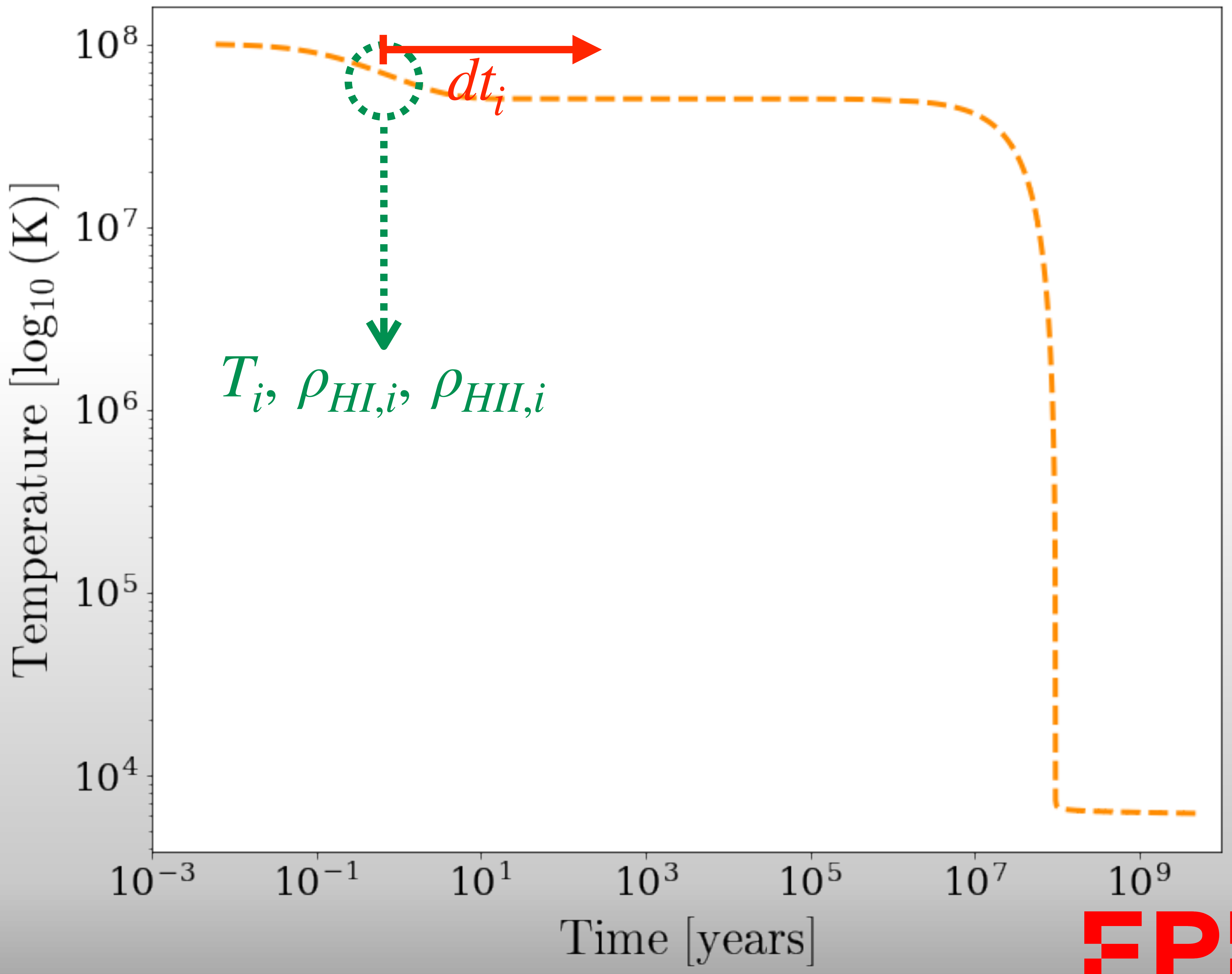
Sample:
T_i
$\rho_{HI,i}$
$\rho_{HII,i}$
dt_i

Grackle



Data Generation & Training





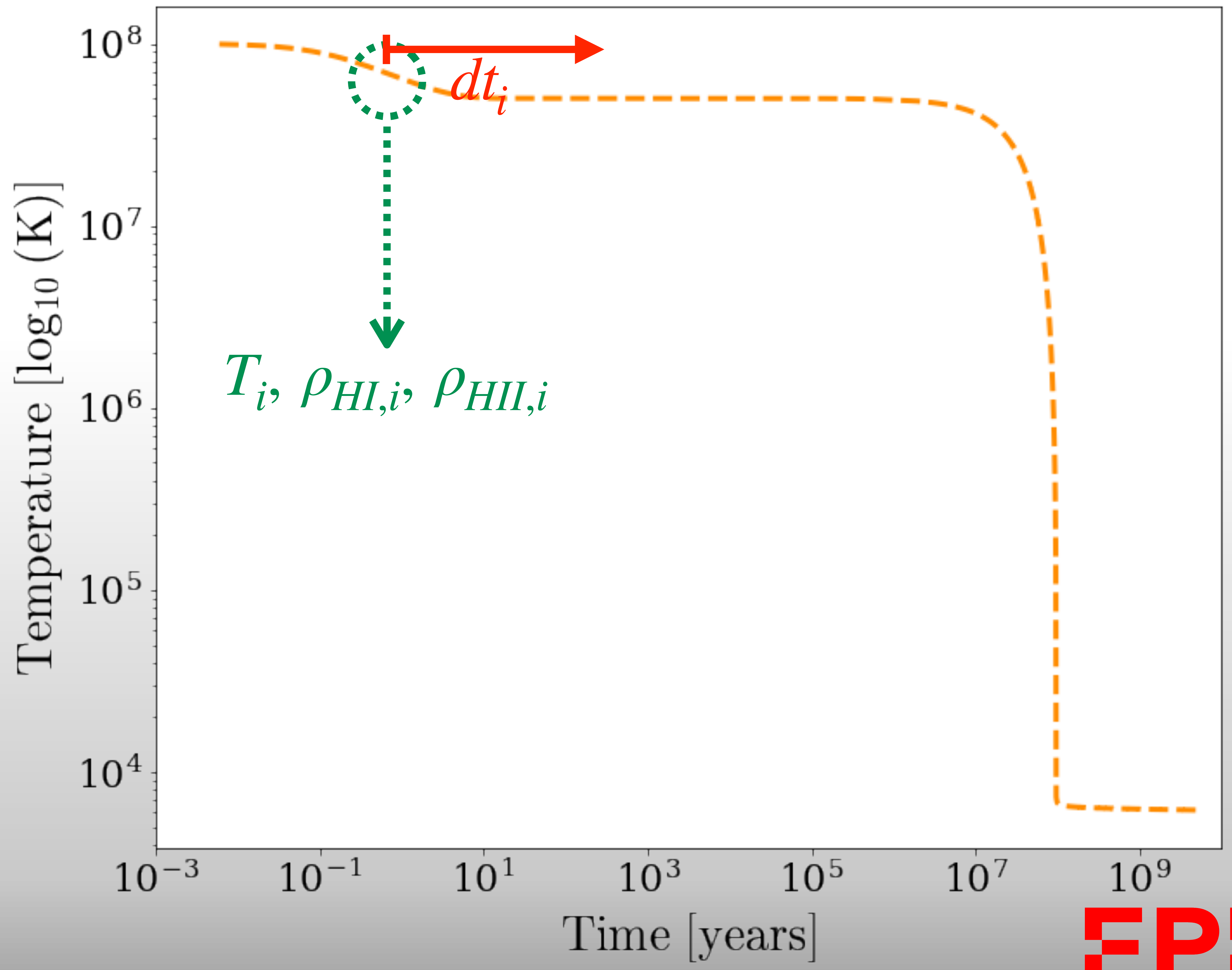
EPFL

Data Generation & Training

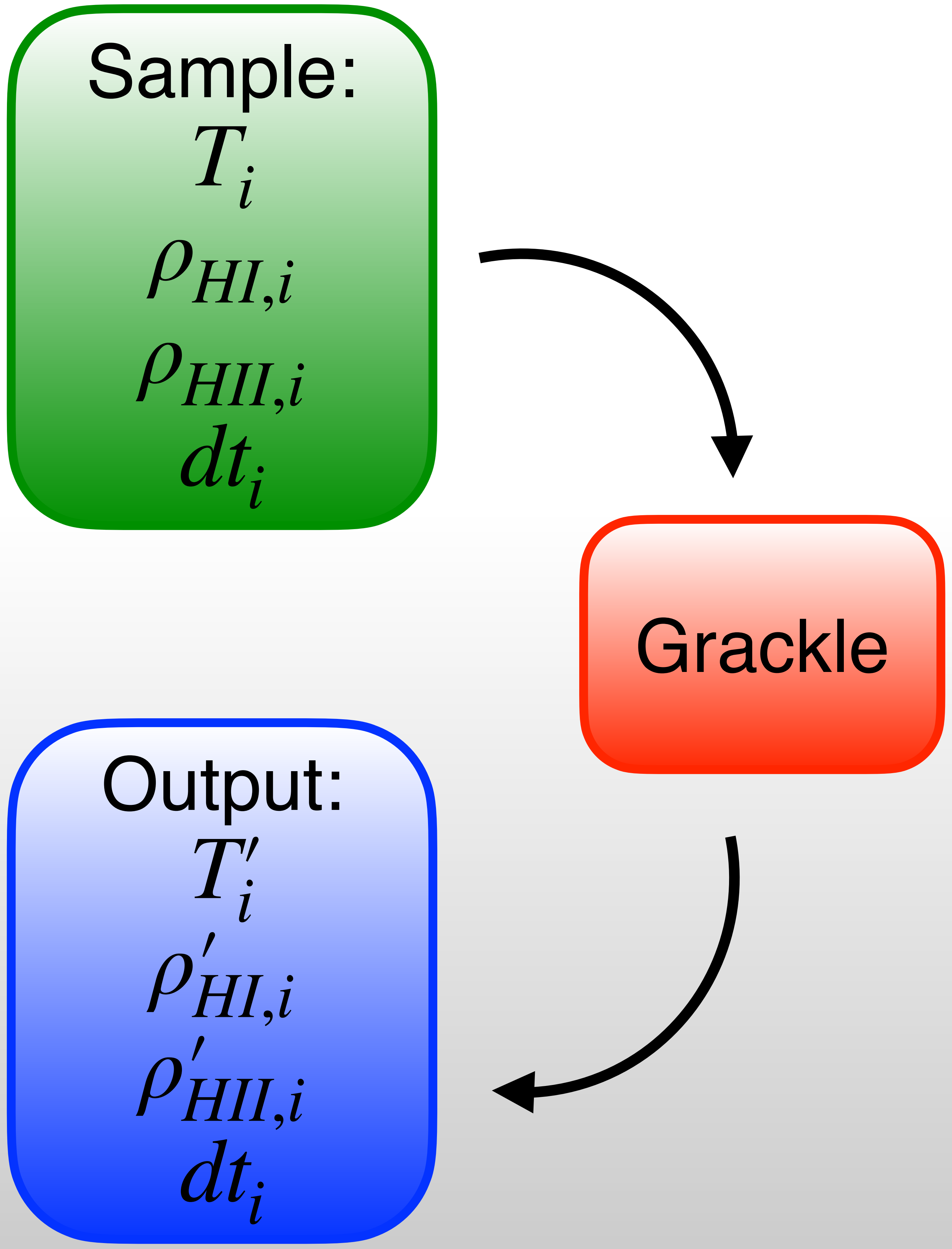
Sample:
T_i
$\rho_{HI,i}$
$\rho_{HII,i}$
dt_i

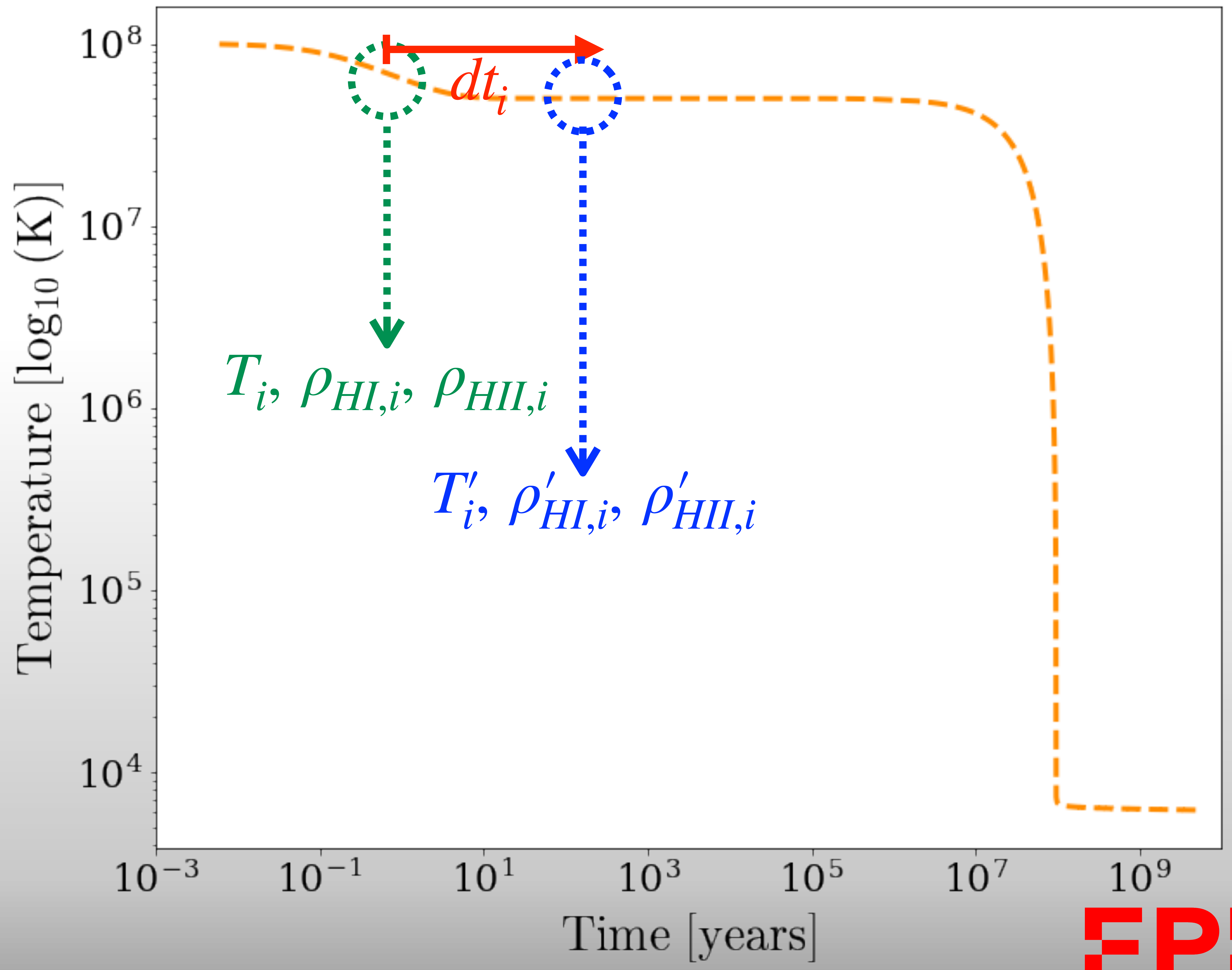
Grackle

Output:
T'_i
$\rho'_{HI,i}$
$\rho'_{HII,i}$
dt_i



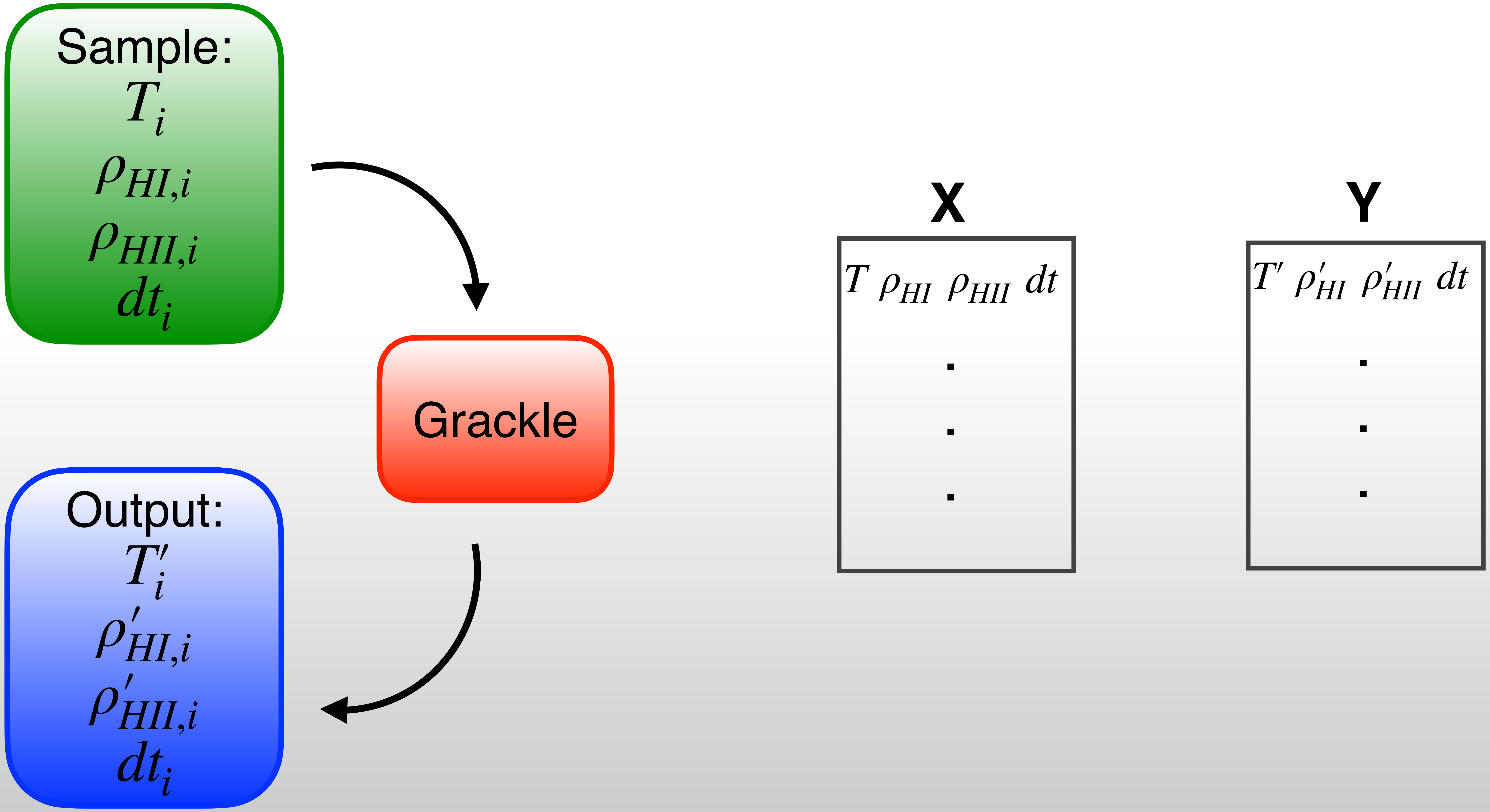
Data Generation & Training



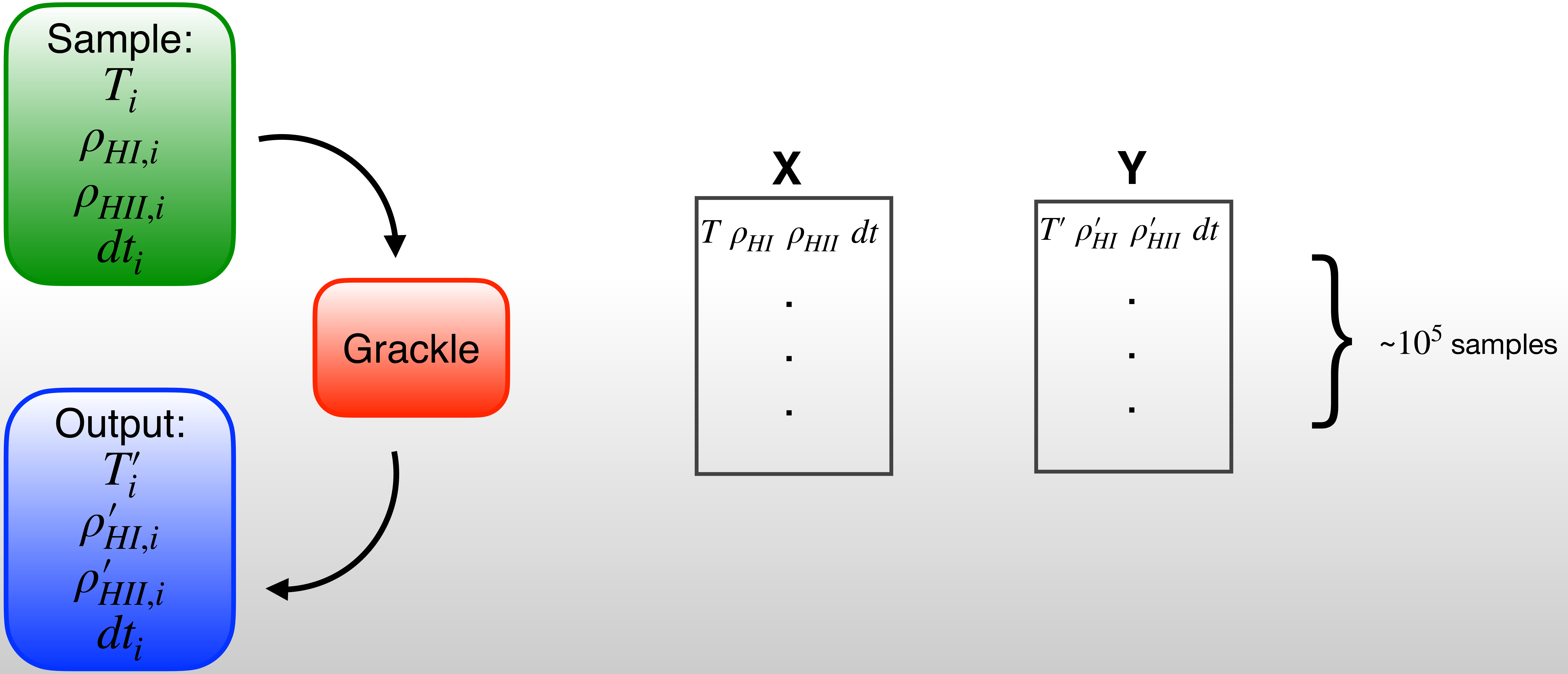


EPFL

Data Generation & Training



Data Generation & Training



EPFL

Data Generation & Training

X

$T \; \rho_{HI} \; \rho_{HII} \; dt$

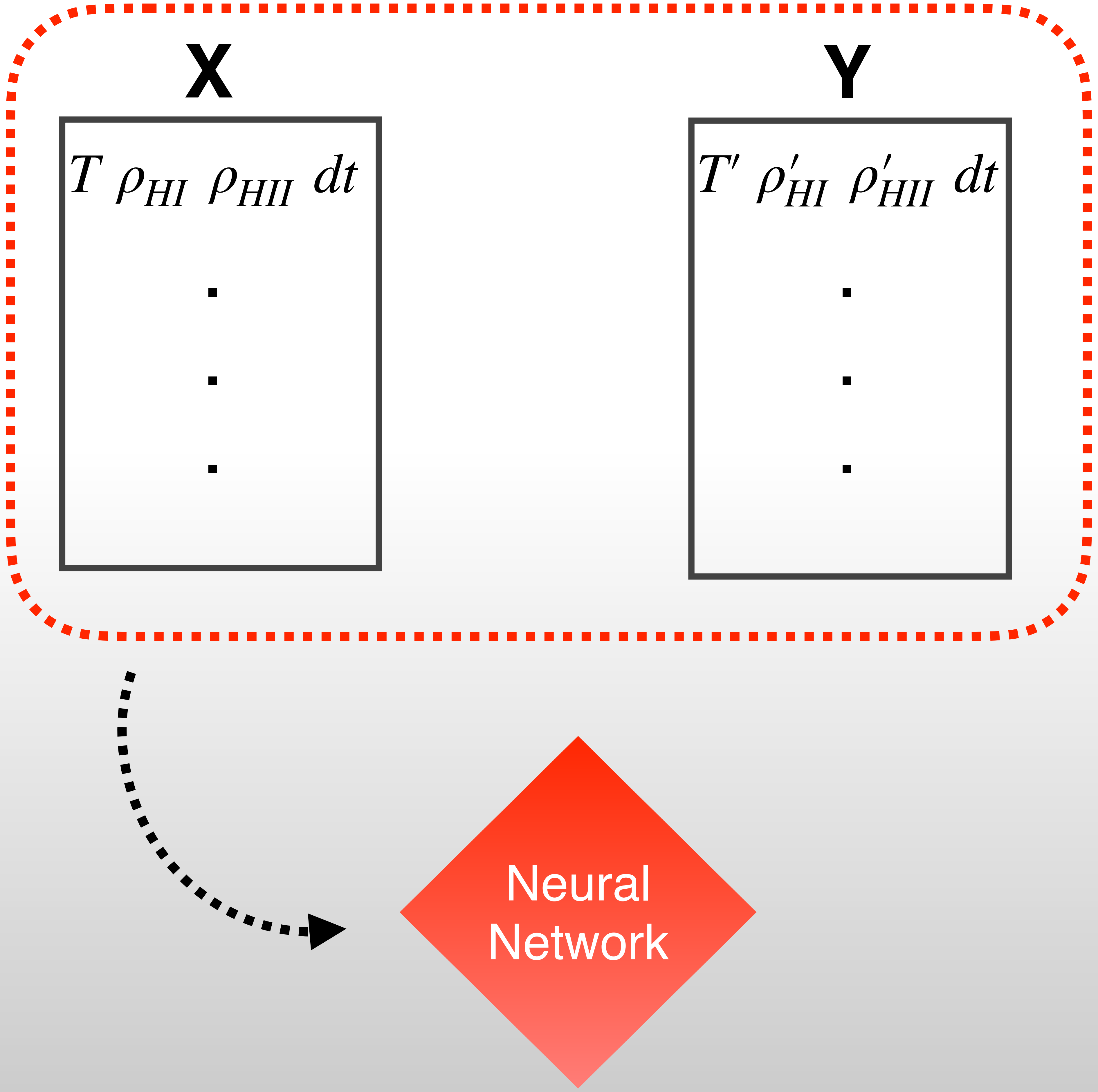
.
.
.

Y

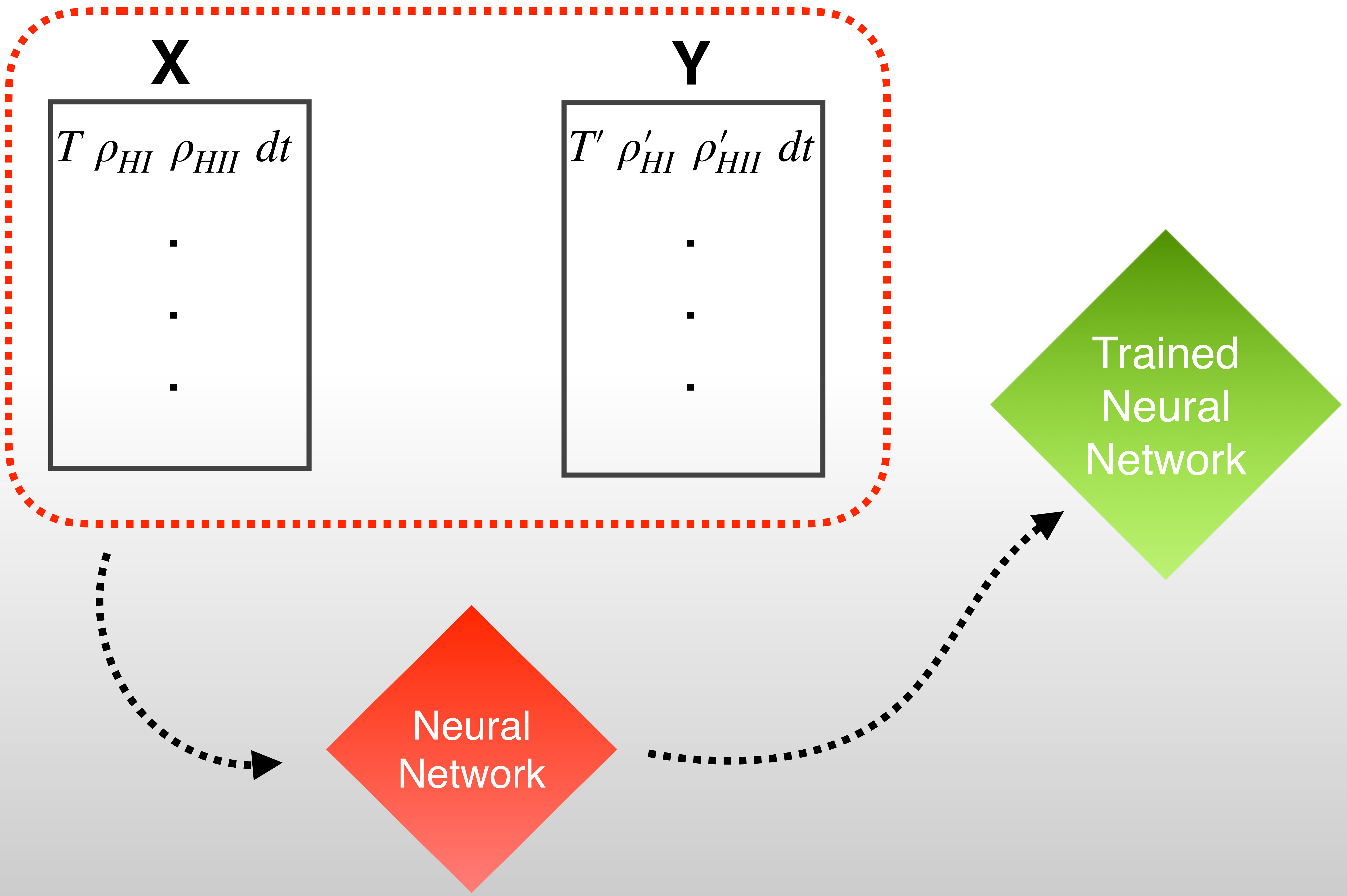
$T' \; \rho'_{HI} \; \rho'_{HII} \; dt$

.
.
.

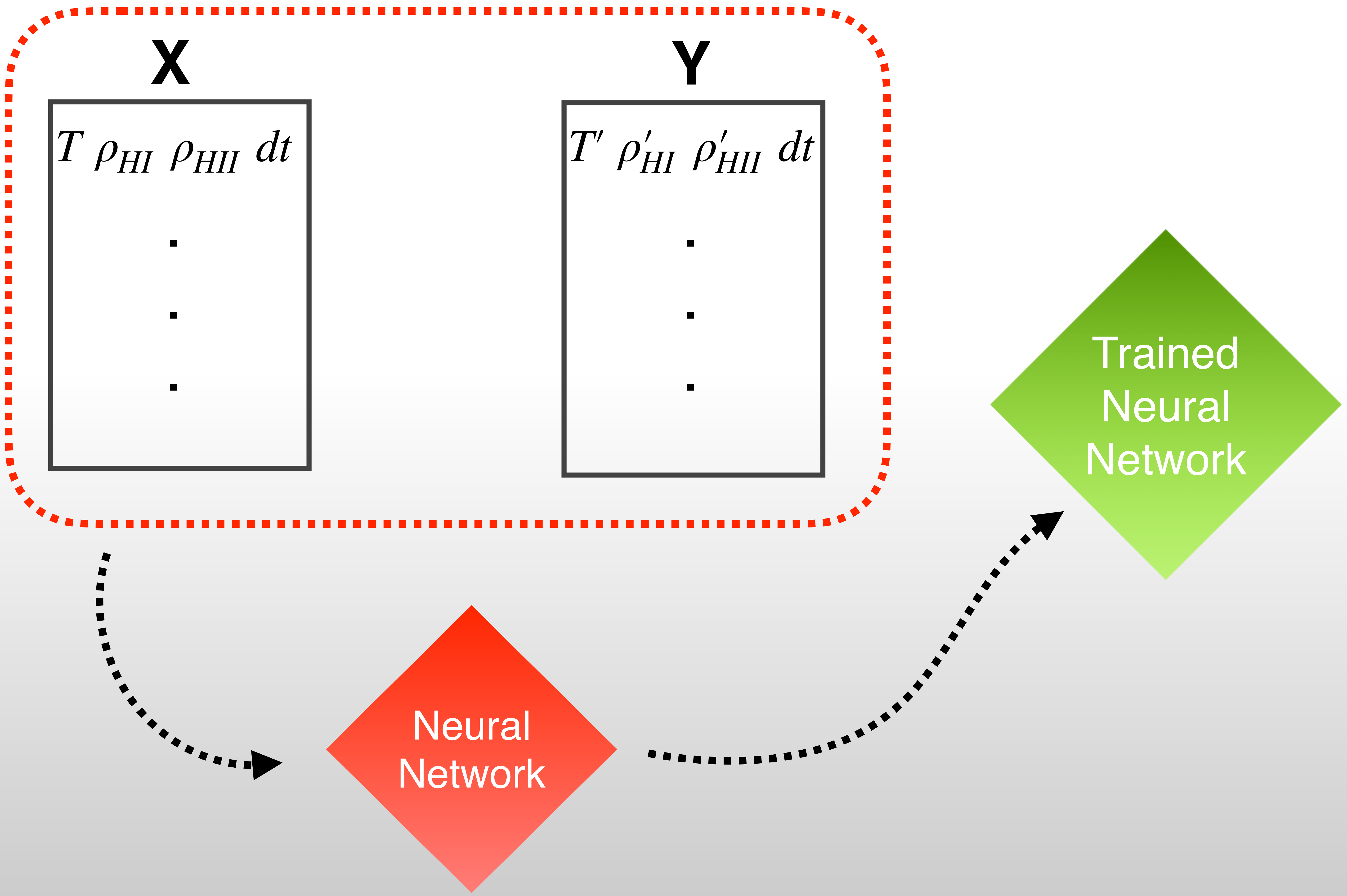
Data Generation & Training

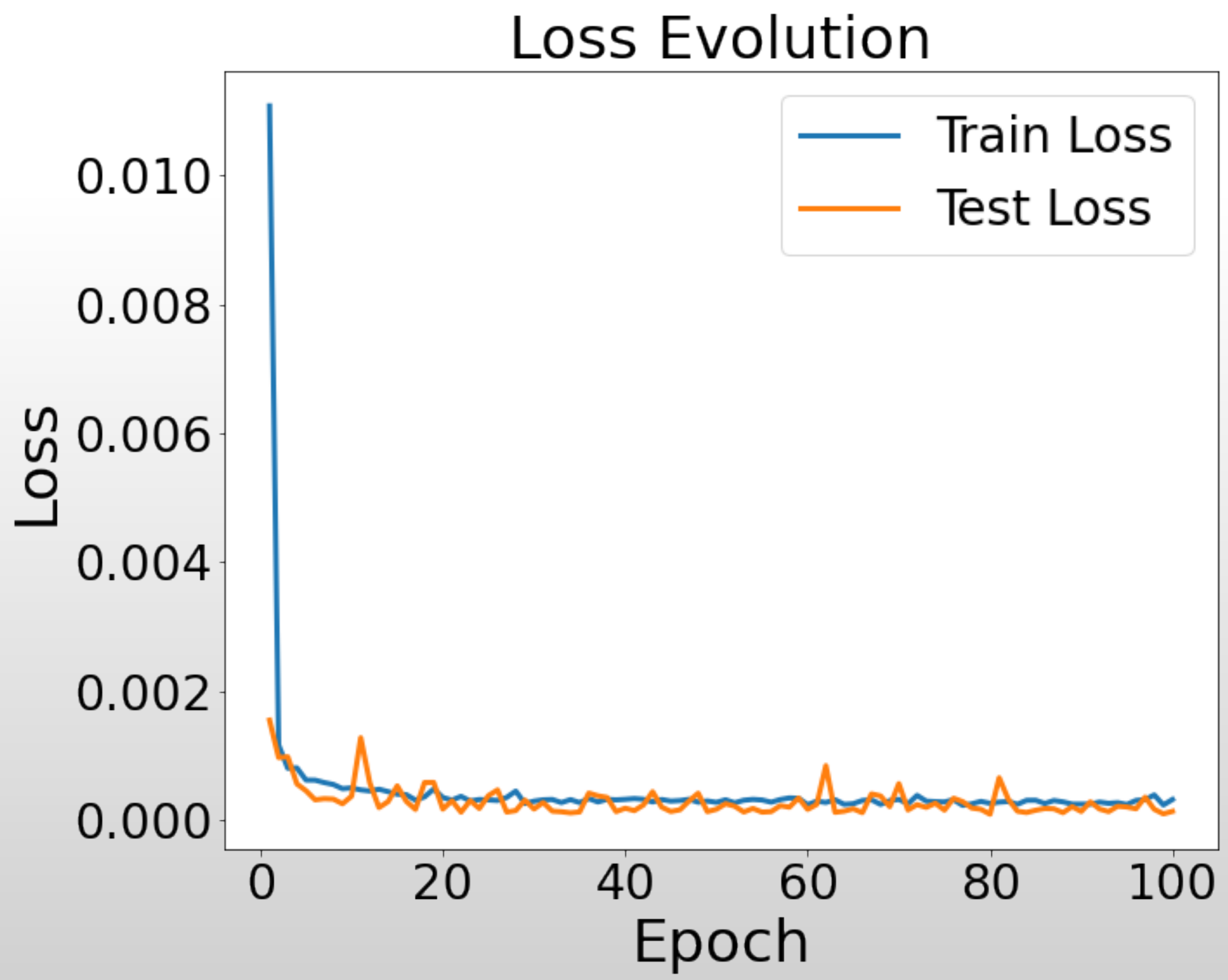


Data Generation & Training



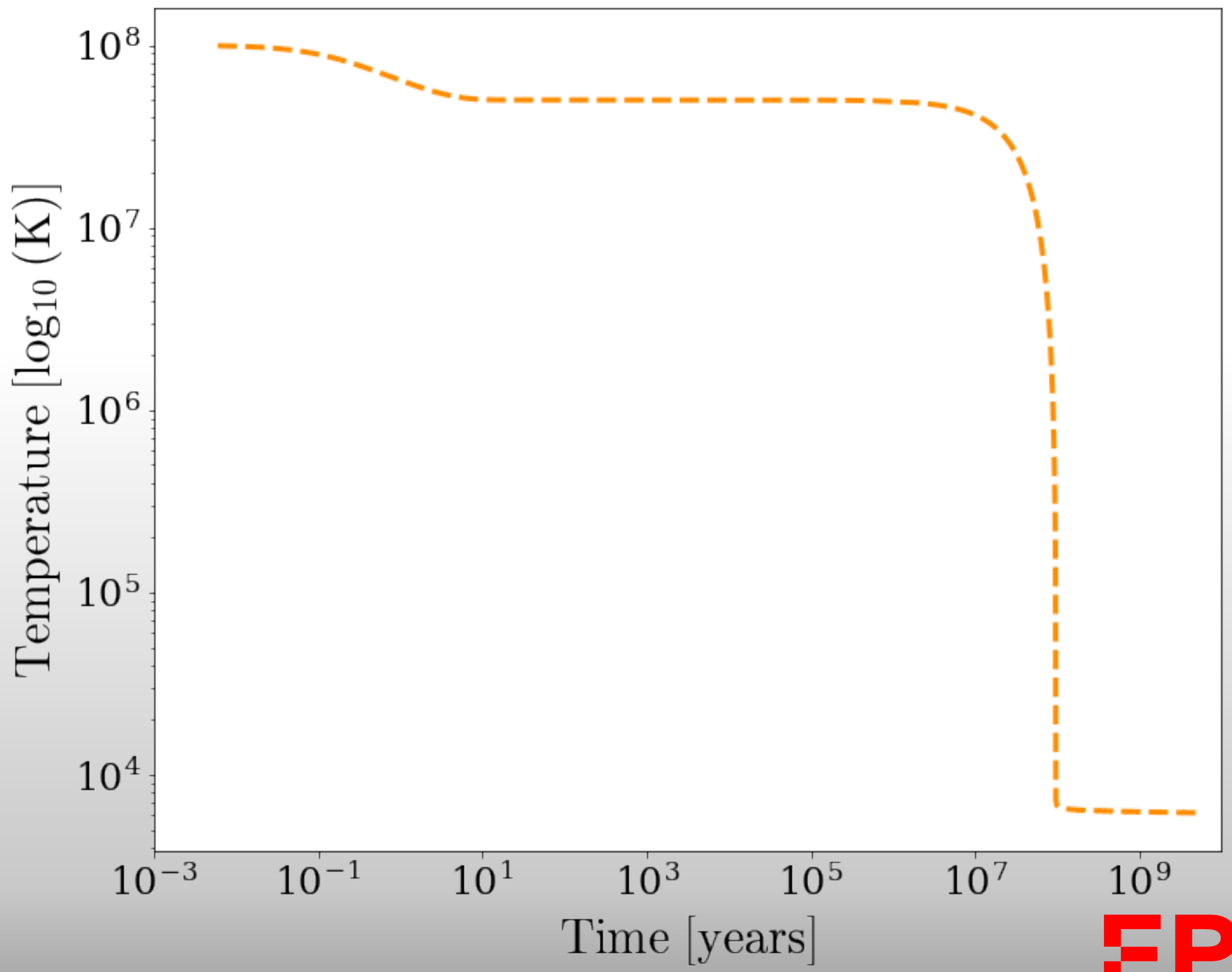
Data Generation & Training





Predicting the Time Evolution of a Cell

Predicting the Time Evolution of a Cell

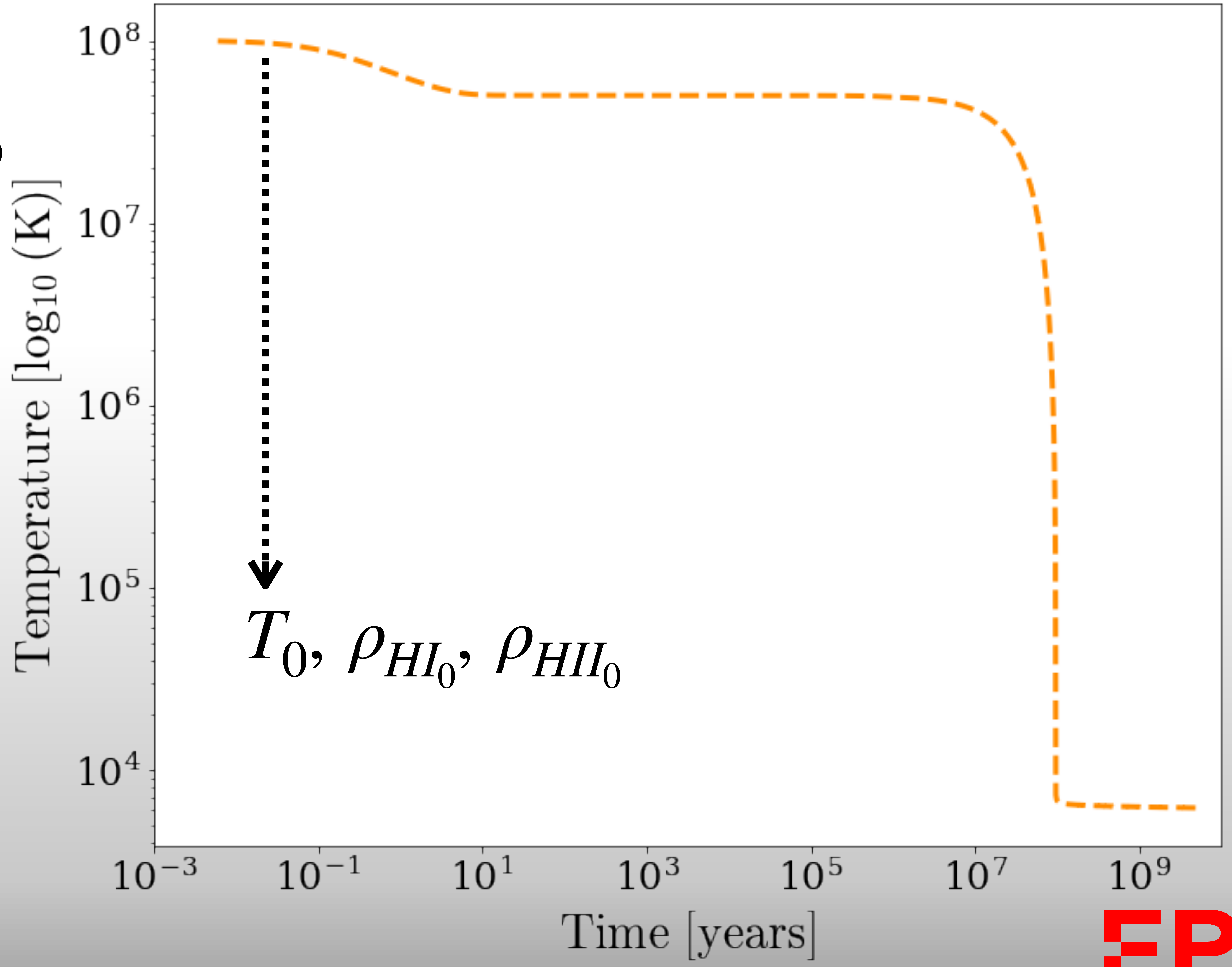


EPFL

Predicting the Time Evolution of a Cell

- Initial values: $T_0, \rho_{HI_0}, \rho_{HII_0}$

T_0 ρ_{HI_0} ρ_{HII_0} T_0 ρ_{HI_0} ρ_{HII_0} $\cdots$ T_0 ρ_{HI_0} ρ_{HII_0}



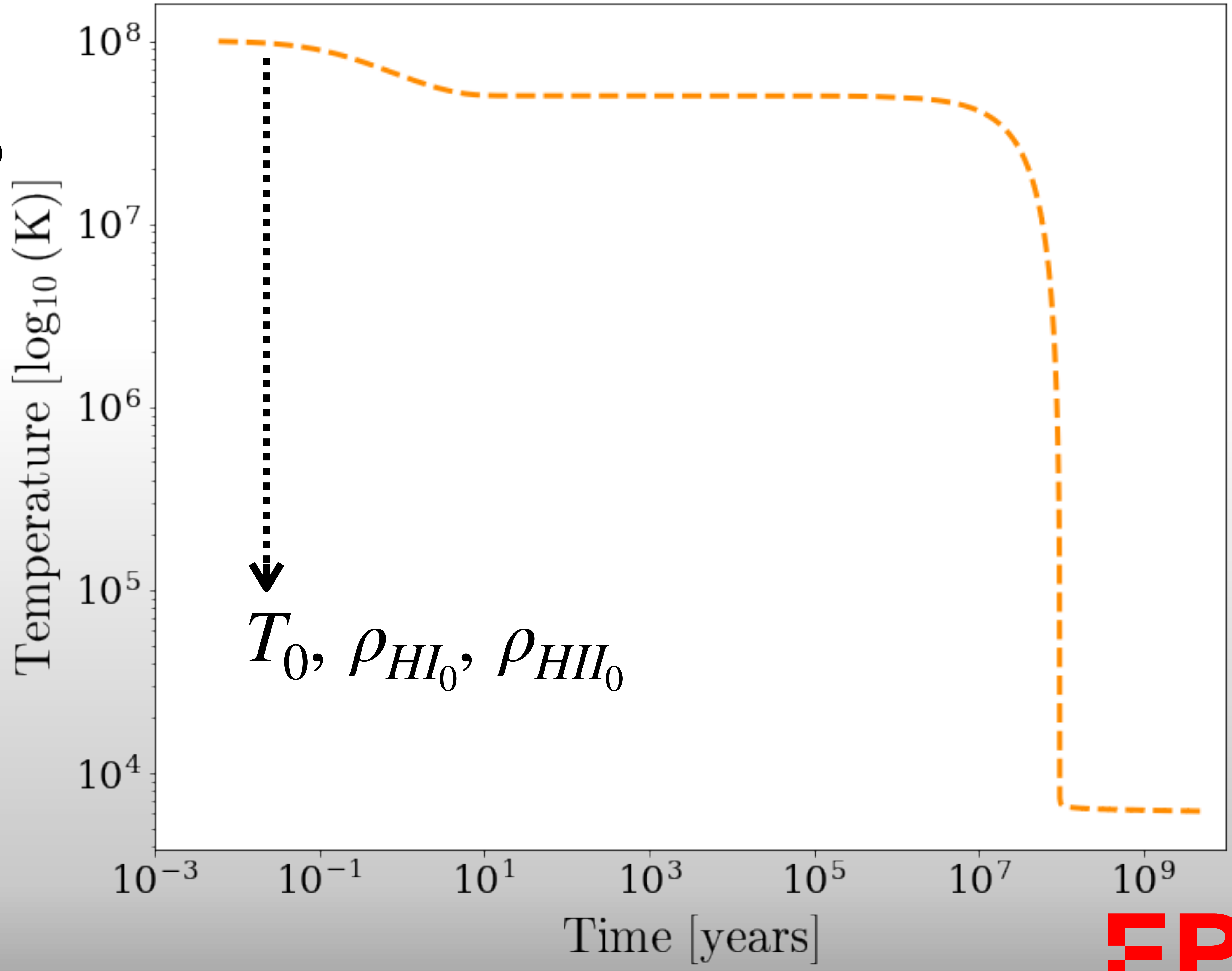
EPFL

Predicting the Time Evolution of a Cell

- Initial values: T_0, ρ_{HI_0}, ρ_{HII_0}
- $dt \in \left[10^{-2}, 10^{9}\right]$ yr

$$dt_i$$

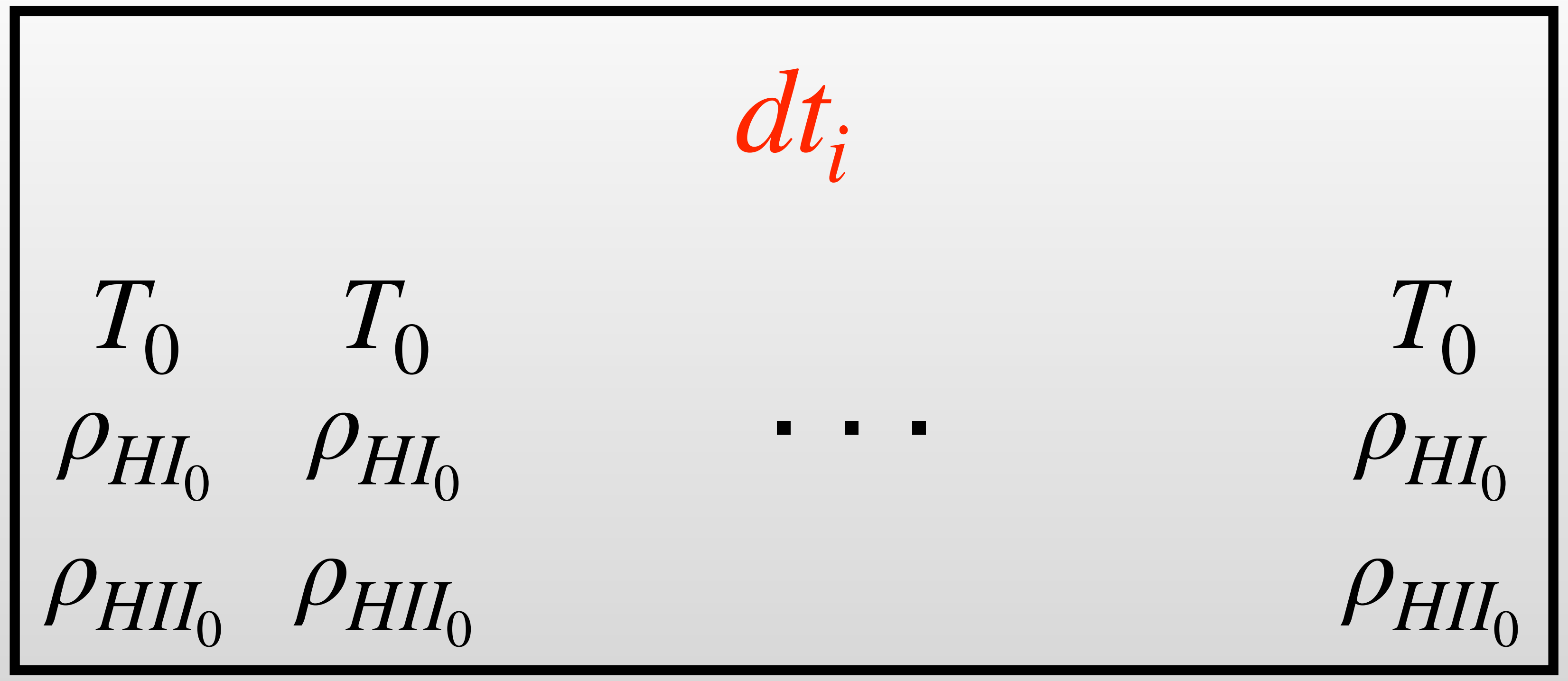
T_0	T_0		T_0
ρ_{HI_0}	ρ_{HI_0}	$\cdots$	ρ_{HI_0}
ρ_{HII_0}	ρ_{HII_0}		ρ_{HII_0}

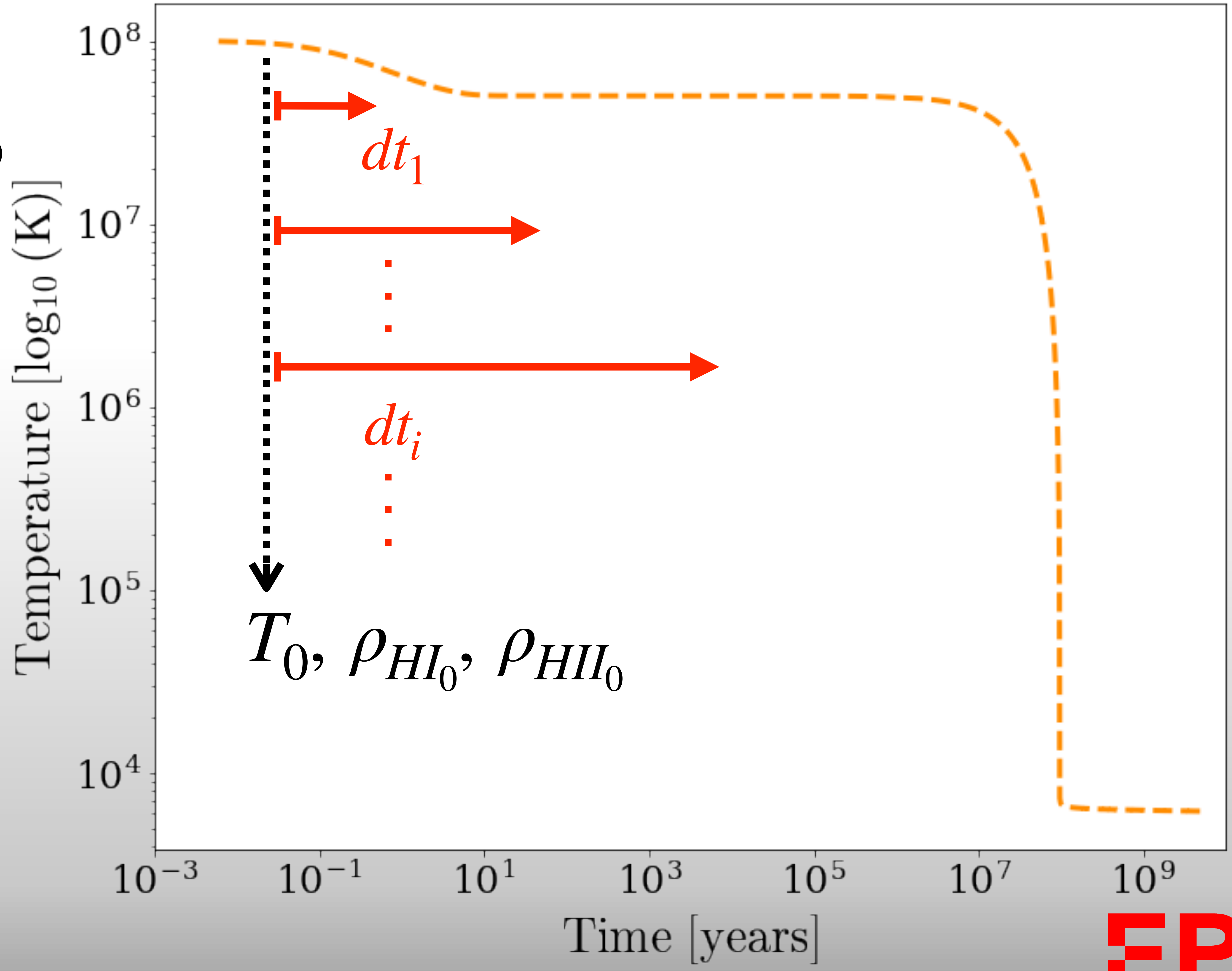


EPFL

Predicting the Time Evolution of a Cell

- Initial values: $T_0,\ \rho_{HI_0},\ \rho_{HII_0}$
- $dt \in \left[10^{-2}, 10^{9}\right]$ yr



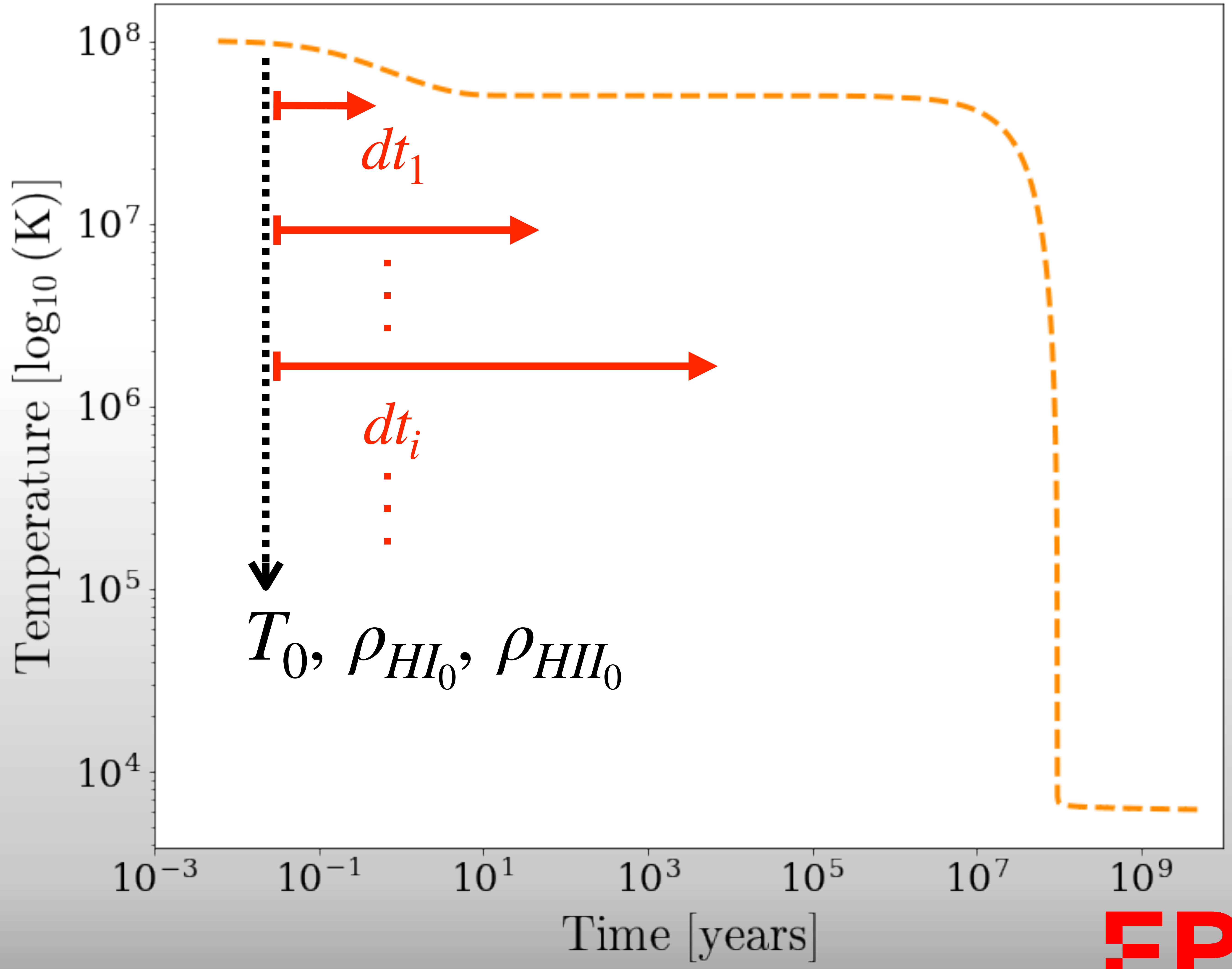


EPFL

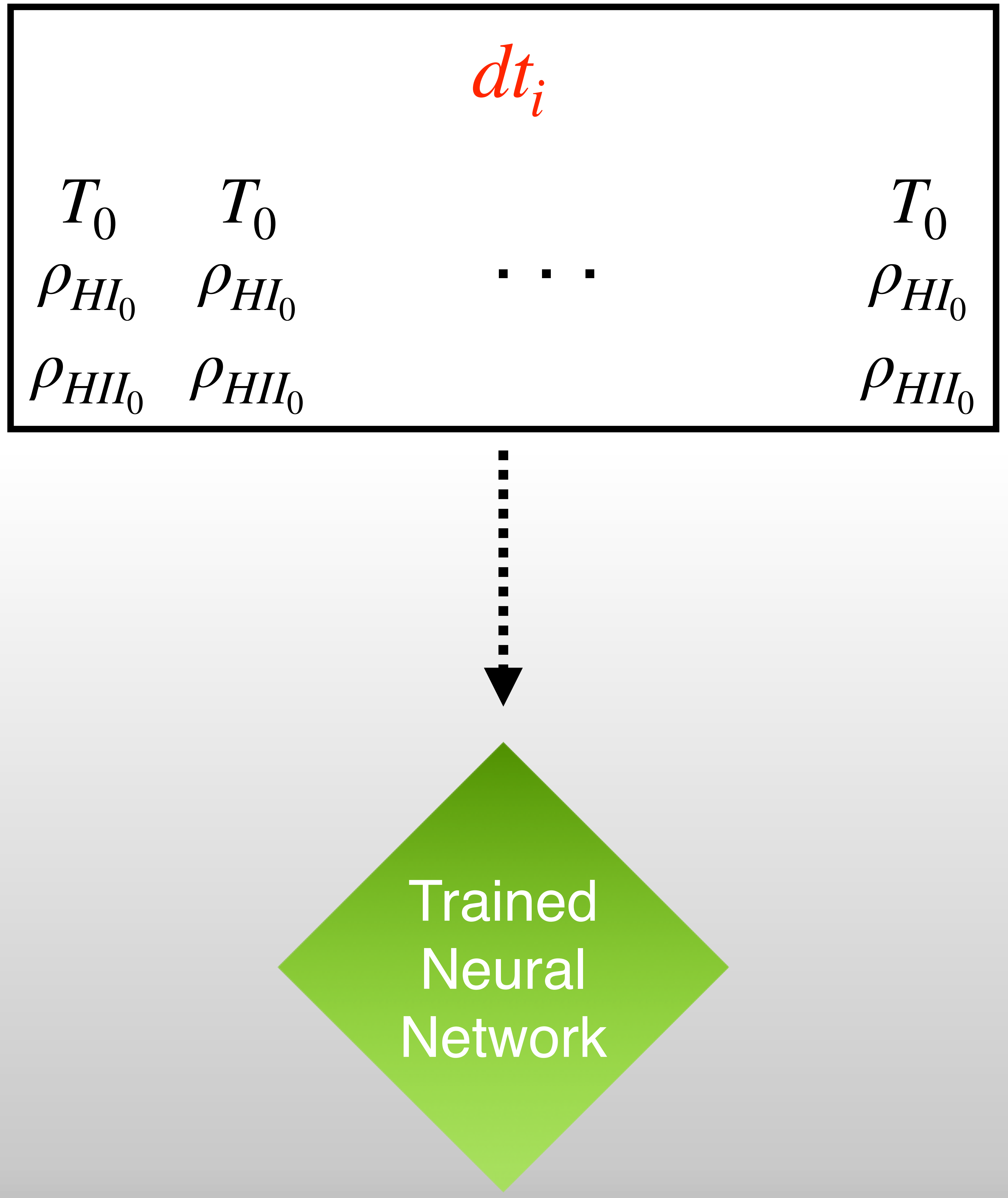
Predicting the Time Evolution of a Cell

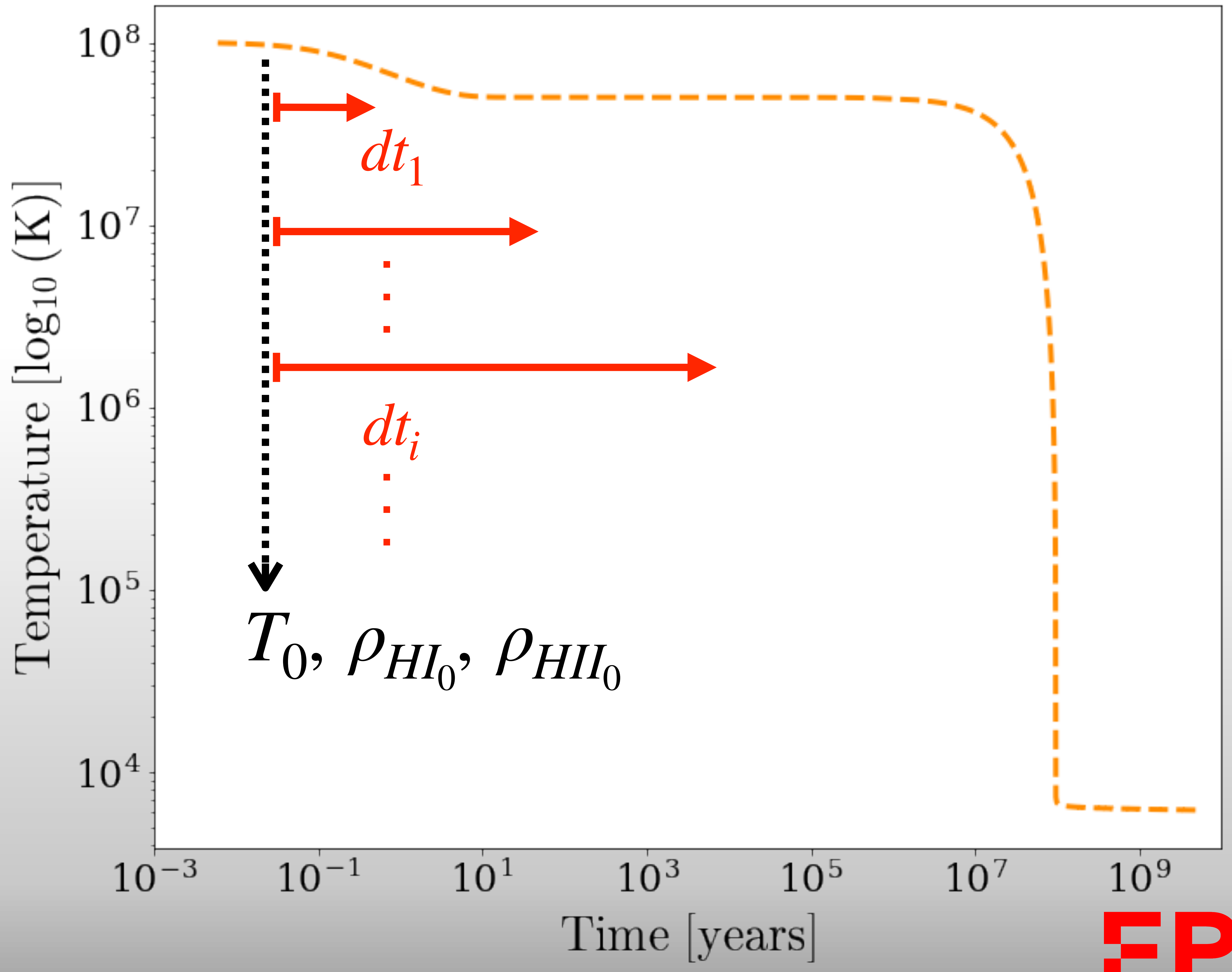
$$dt_i$$

T_0	T_0		T_0
ρ_{HI_0}	ρ_{HI_0}	$\cdots$	ρ_{HI_0}
ρ_{HII_0}	ρ_{HII_0}		ρ_{HII_0}



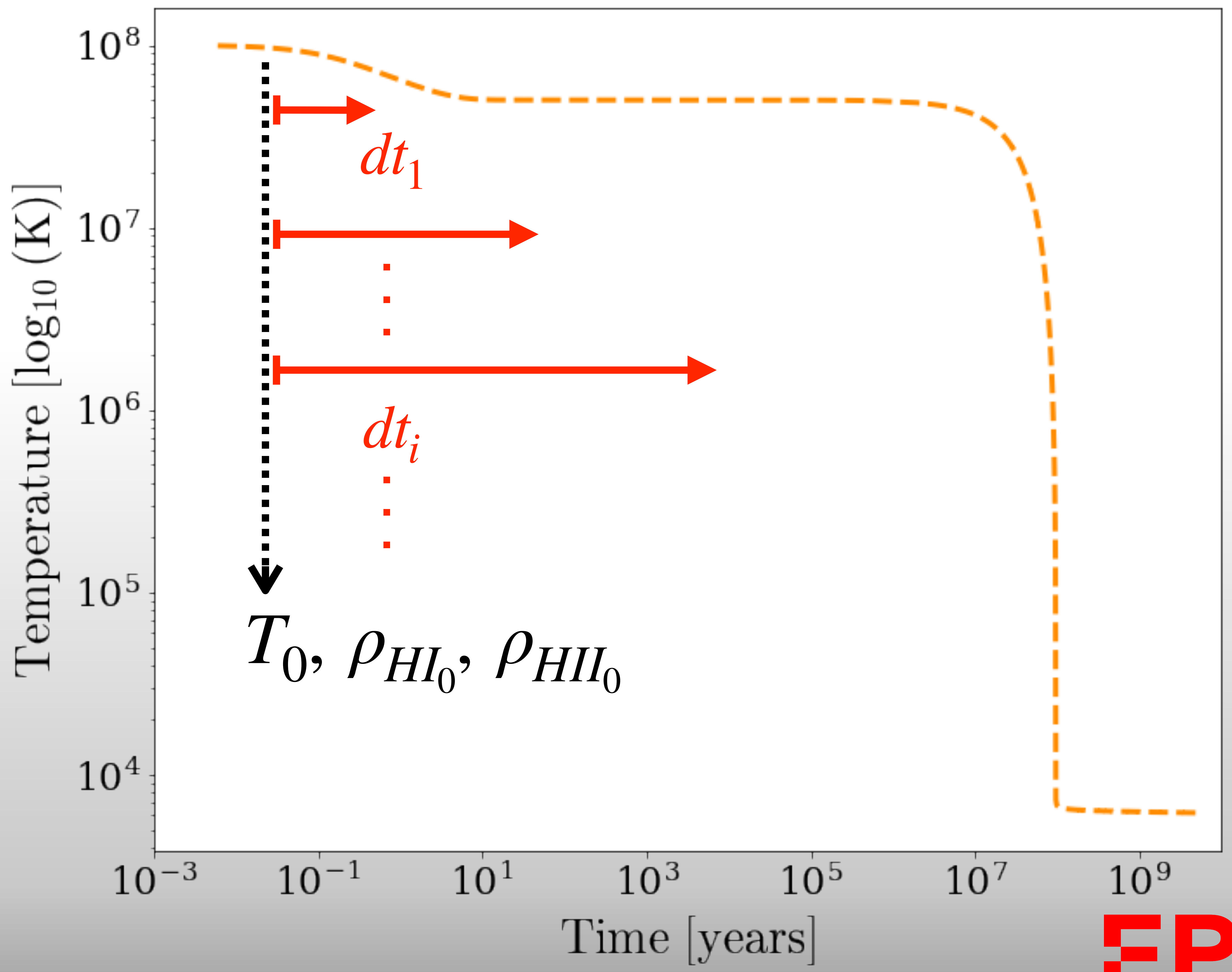
Predicting the Time Evolution of a Cell



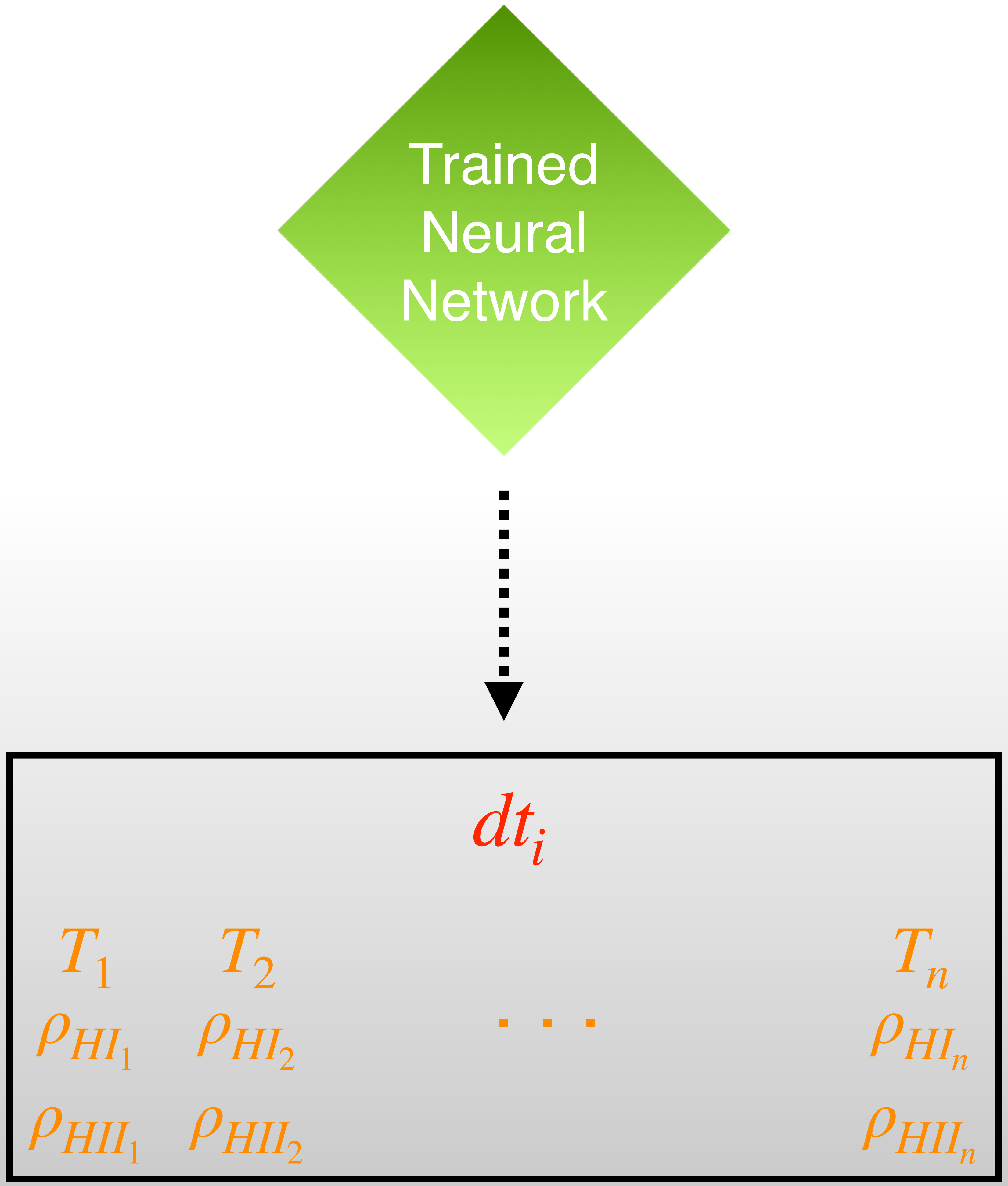


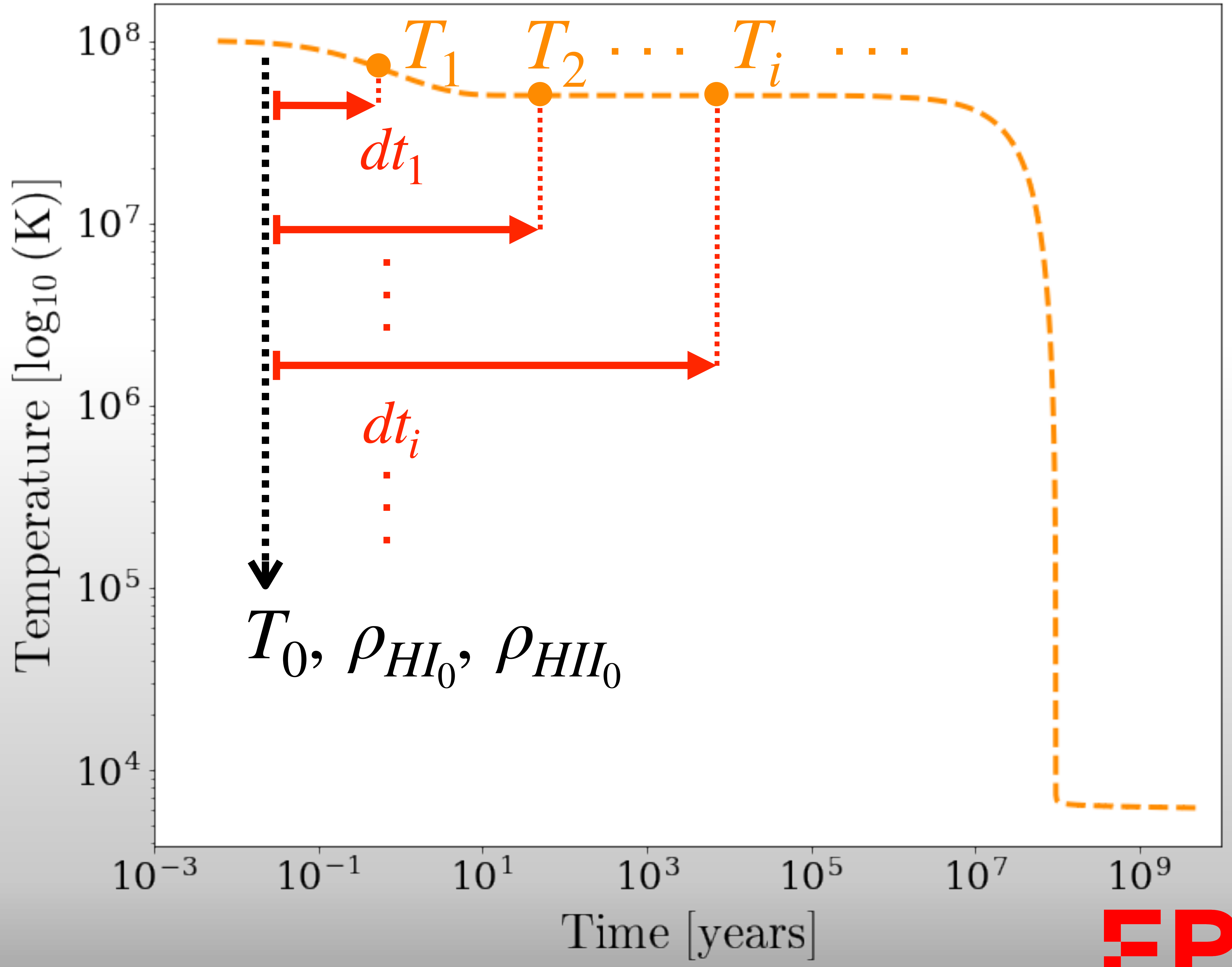
Predicting the Time Evolution of a Cell

Trained Neural Network



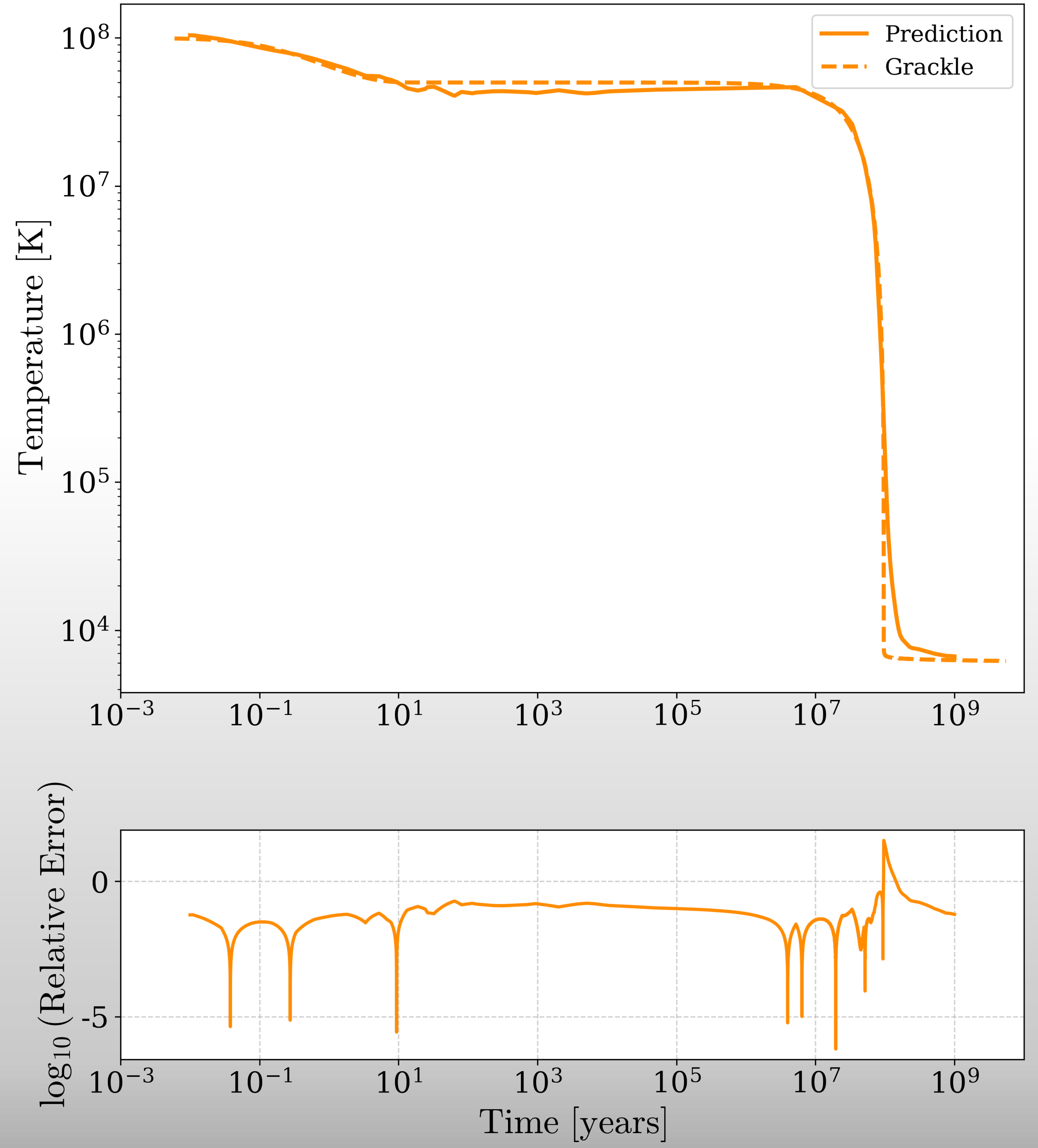
Predicting the Time Evolution of a Cell





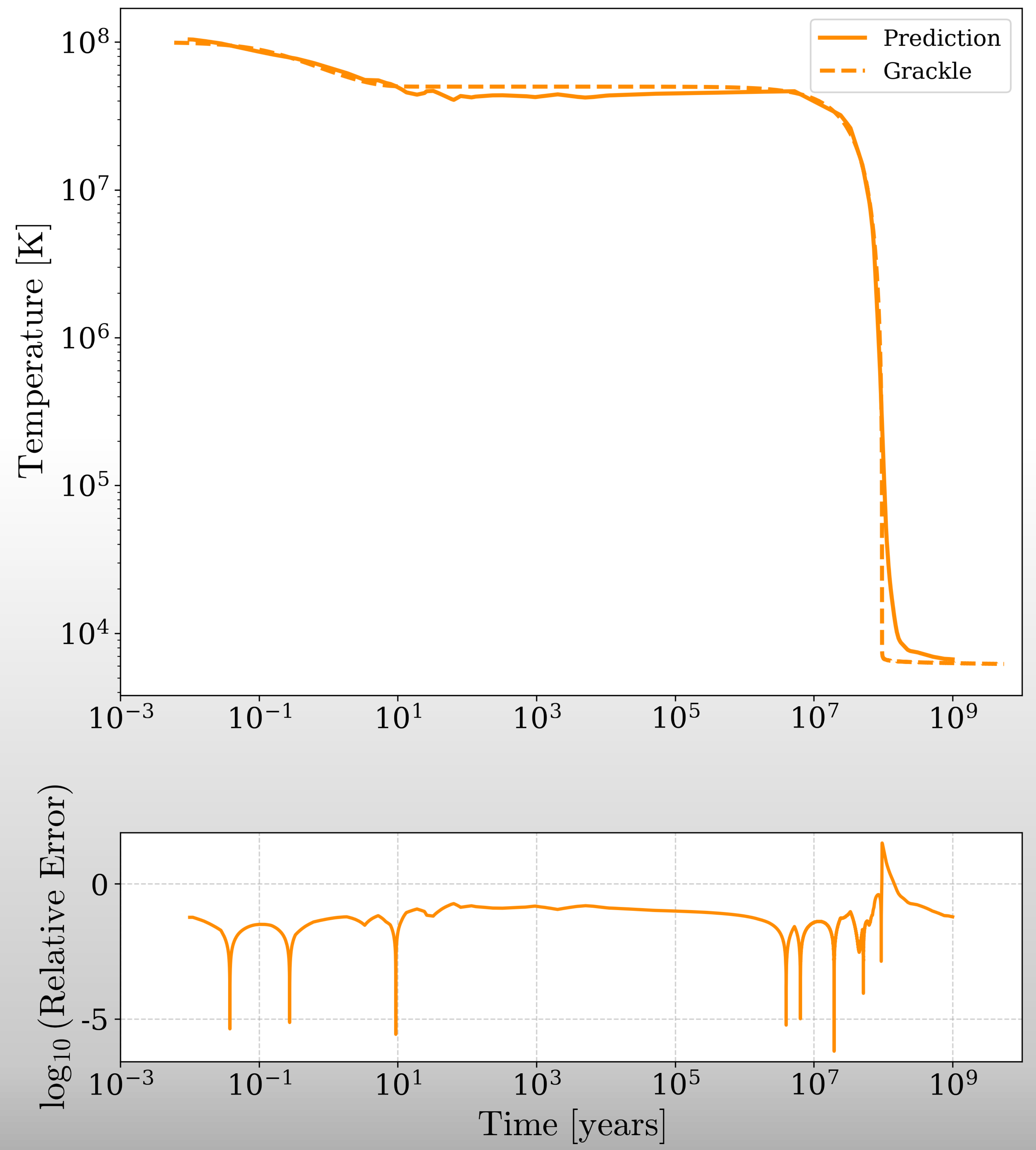
Predicting the Time Evolution of a Cell

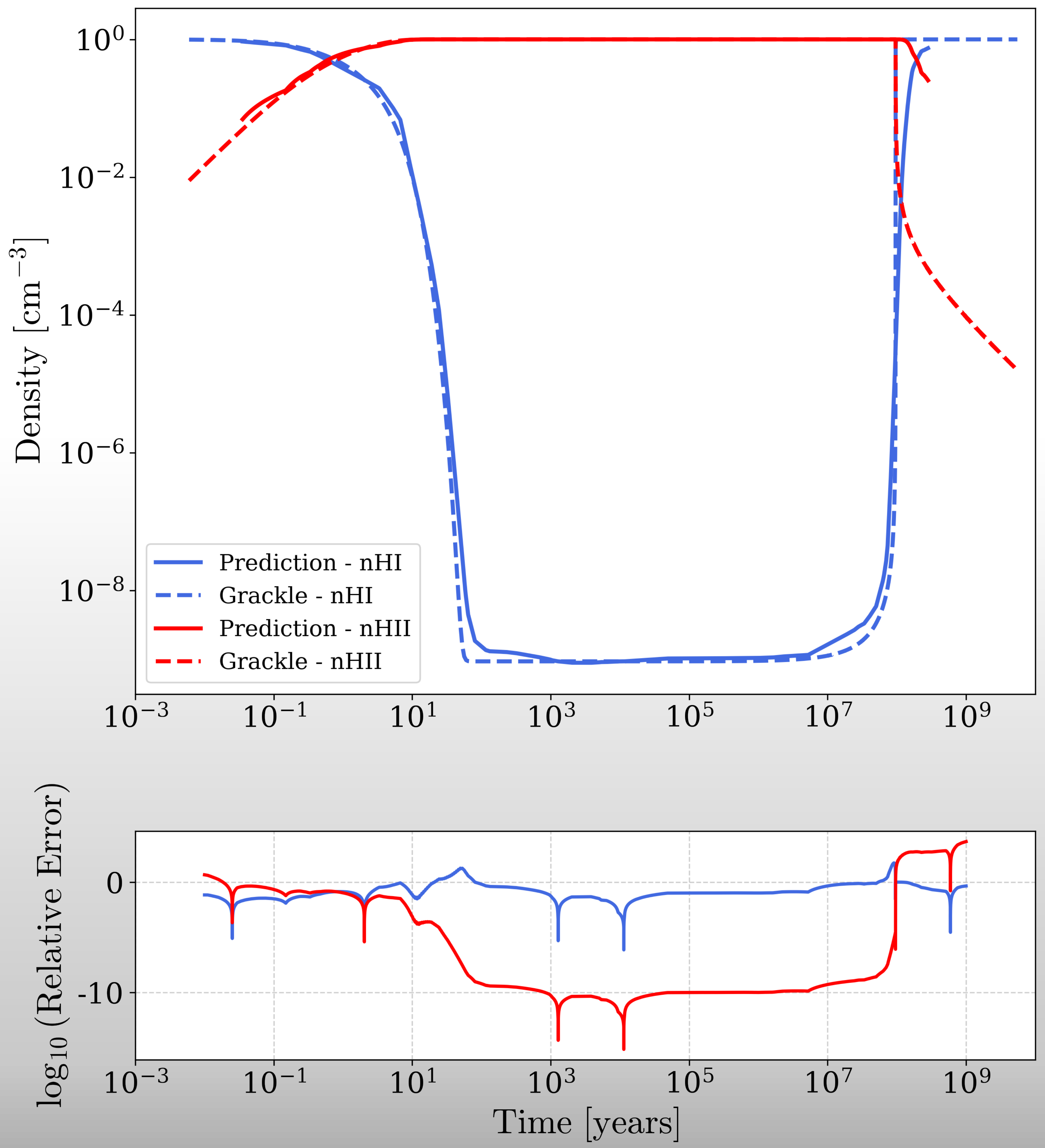
Predicting the Time Evolution of a Cell



EPFL

Predicting the Time Evolution of a Cell





Next Steps

Next Steps

Next Steps

- Enforcing physical constraints at training phase

Next Steps

- Enforcing physical constraints at training phase
- Increasing the number of species: $\rho_{H_2}, \rho_{He}, \ldots$

EPFL

Next Steps

- Enforcing physical constraints at training phase
- Increasing the number of species: $\rho_{H_2}, \rho_{He}, \ldots$
- Adding radiation effects like UV background

Conclusion

Conclusion

Neural networks may offer a promising approach to reducing computational bottlenecks in cosmological simulations, even with relatively simple architectures.