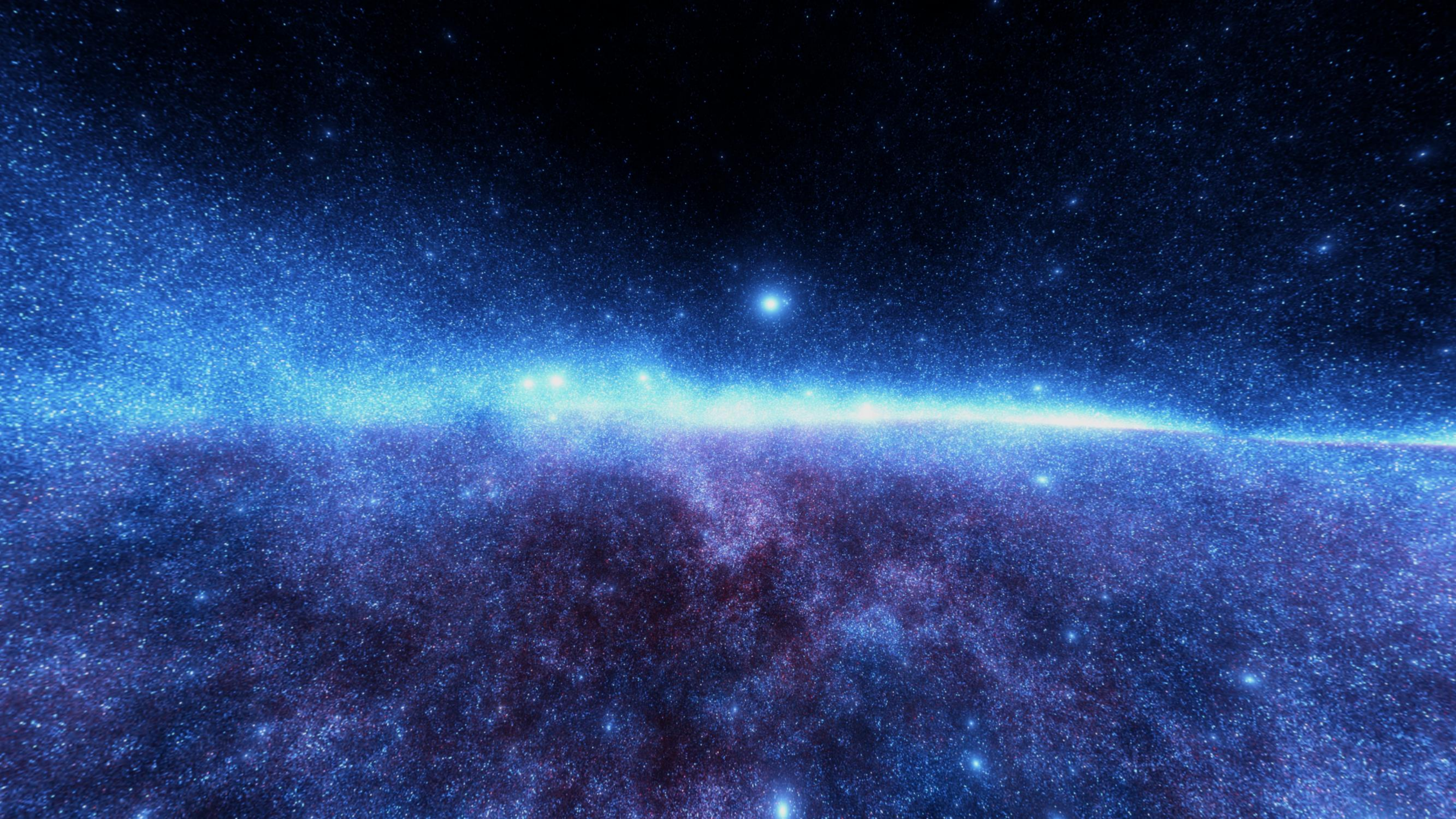
Feasibility study of a neural network approach for mapping the distribution of interstellar medium

GRESSIER Maxime
LOLIVIER Antonin

Feasibility study of a neural network approach for mapping the distribution of interstellar medium

I- Introduction



II- Physical theory

Volterra's equation of the first kind

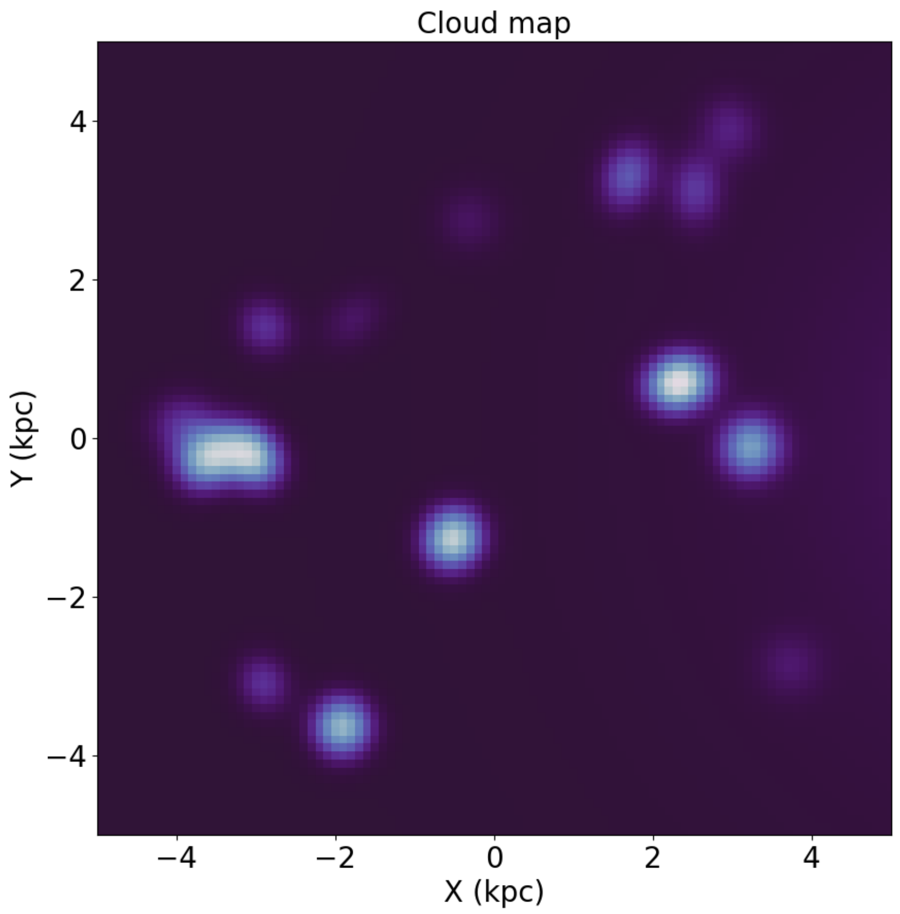
$$A(x) = \int_0^x d(x')dx'$$

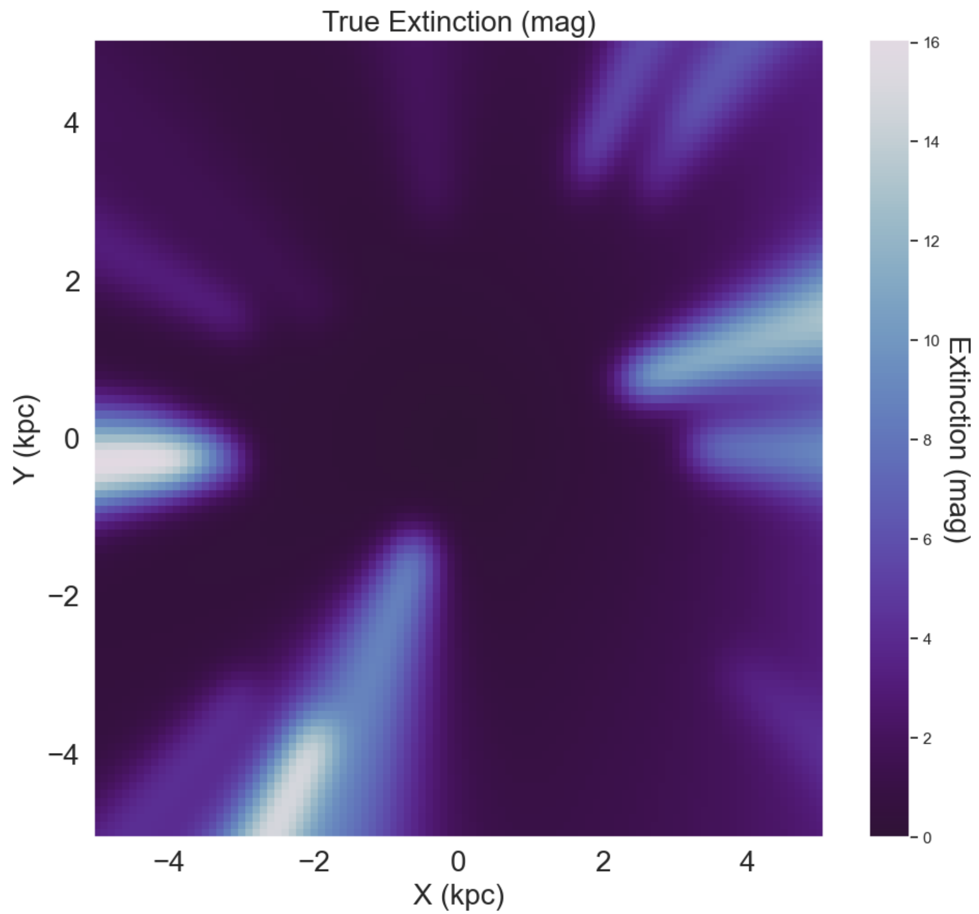
$A(x) \rightarrow Absorbance$

$d(x') \rightarrow Extinction\ density$

II- Physical theory

Simulated datas

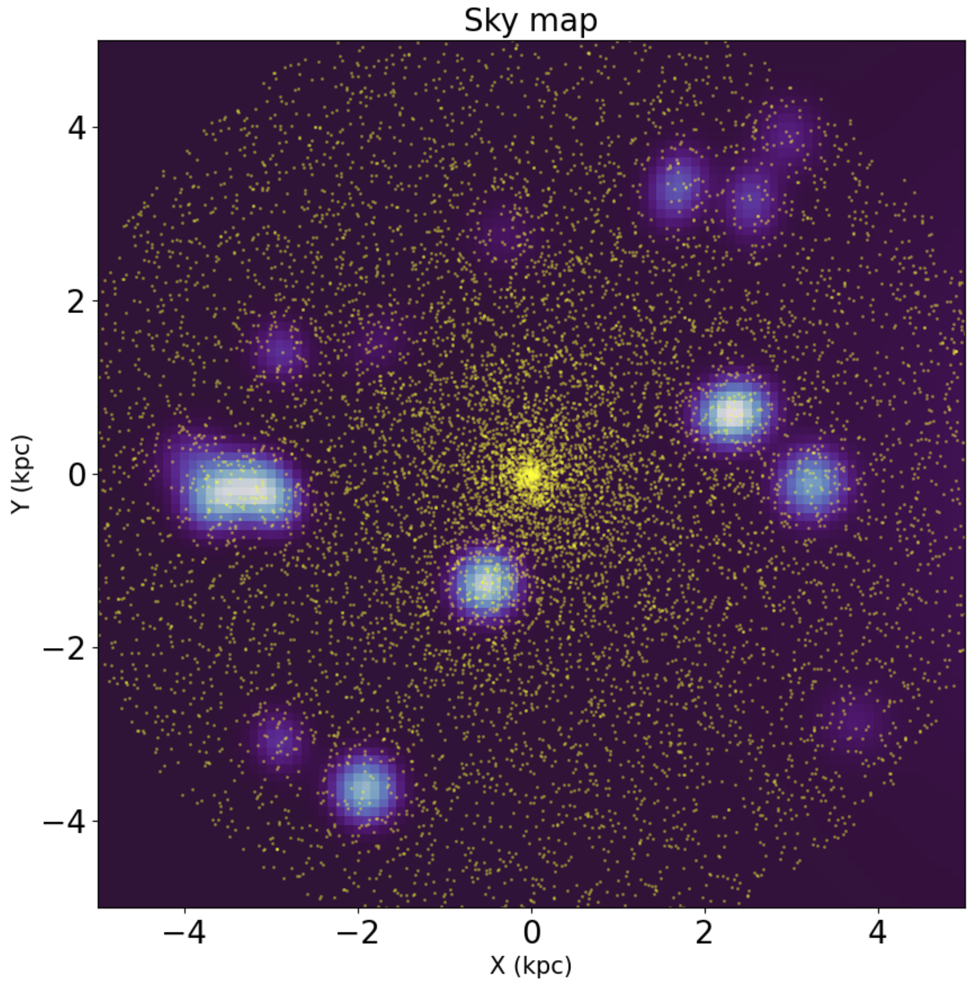




II- Physical theory

Simulated datas

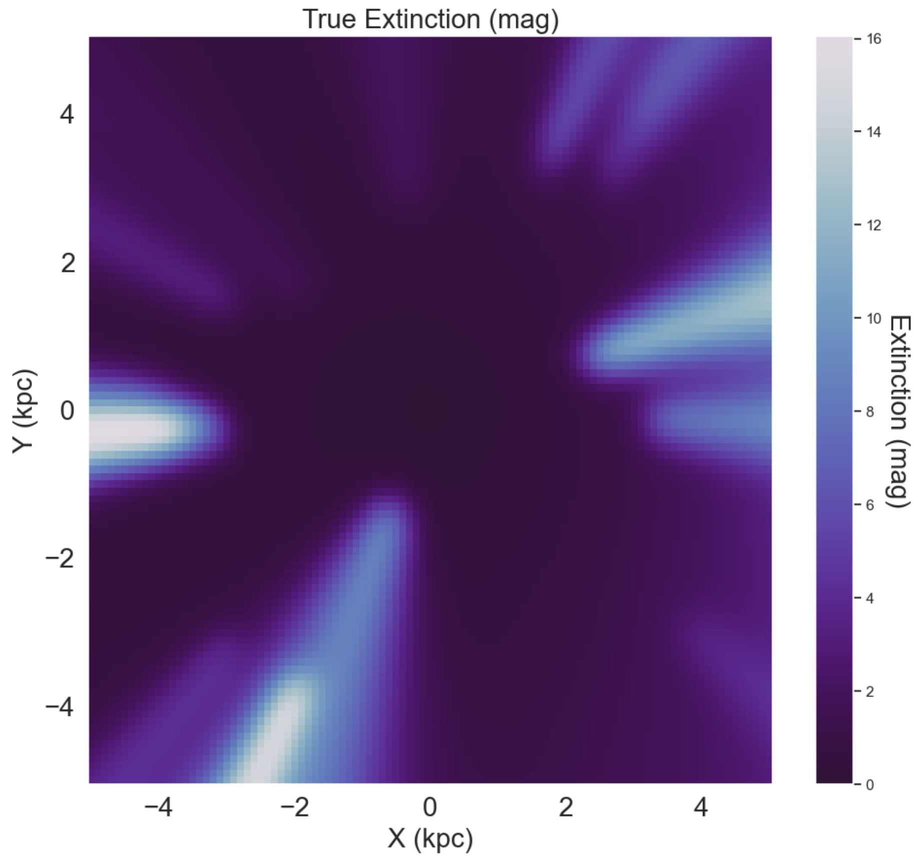
Star repartition



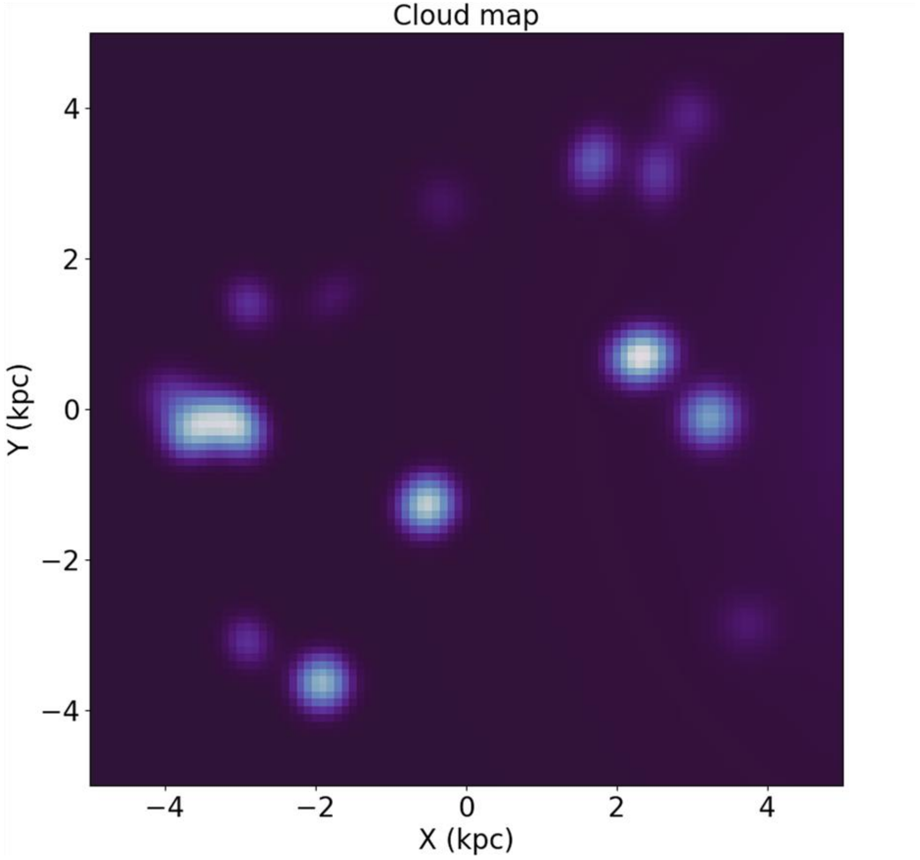
II- Physical theory

Goals

Start from integral



To density



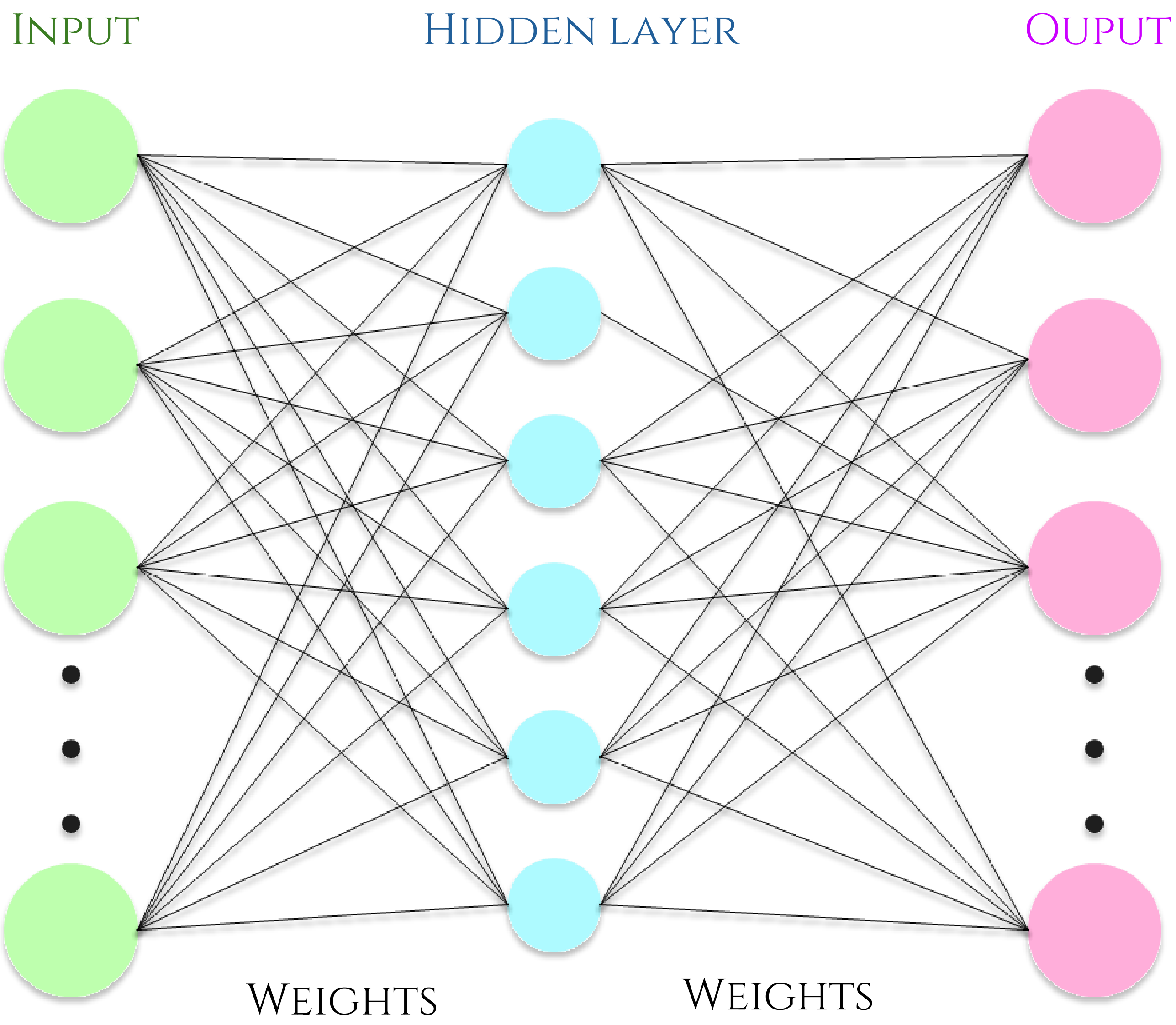
II- Physical theory

Goals

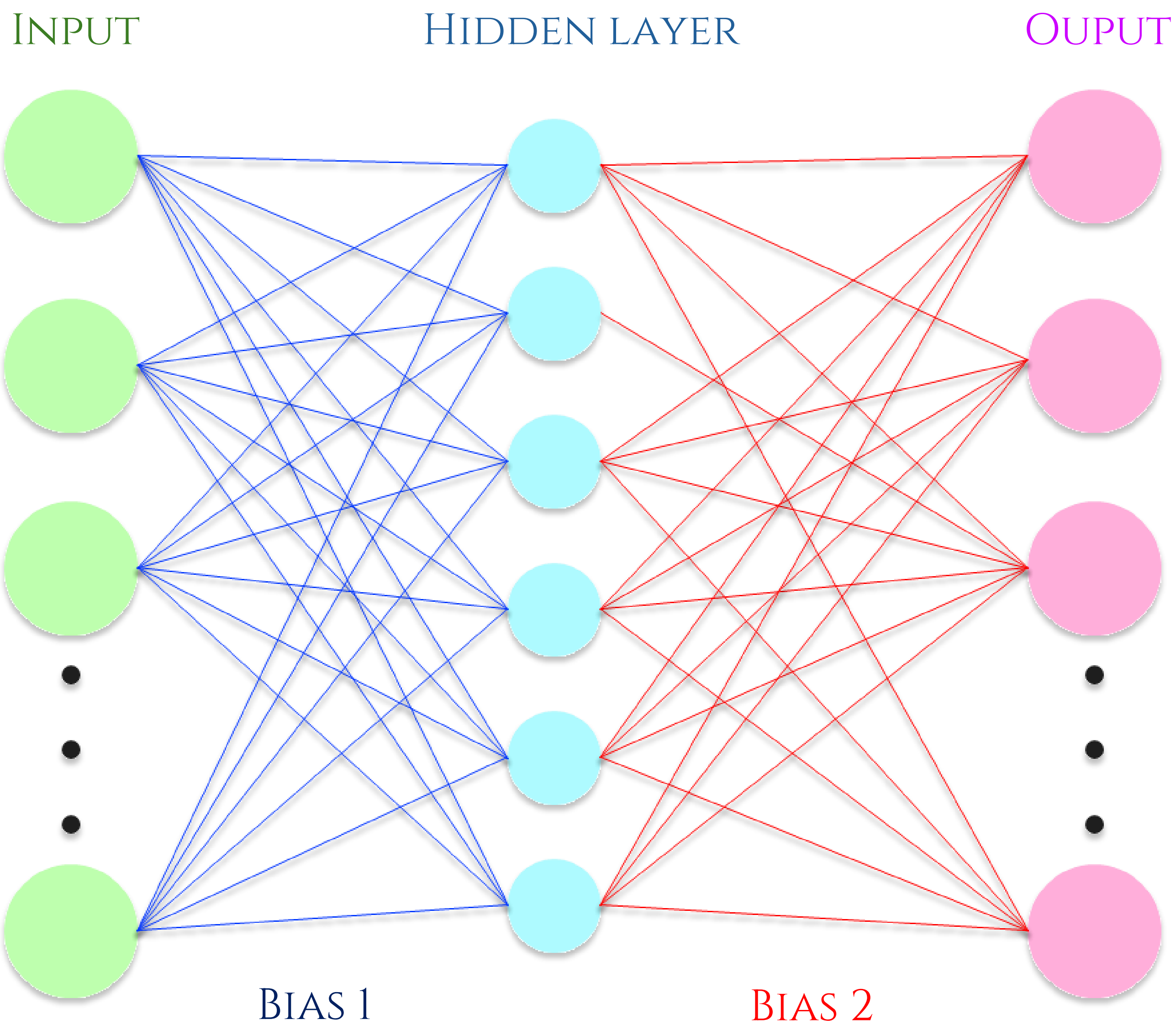
Three ways to achieve this

	Pros	Cons
• Analytically :	- Precision - Understanding	- Can be difficult or impossible to find - Can be very specific
• Numerically :	- Tunnable precision - Flexibility	- Lack of precision - Size sensitve
• Neural network :	- Adaptability - Speed	- No extrapolation - One dataset = One network

III- Neural network

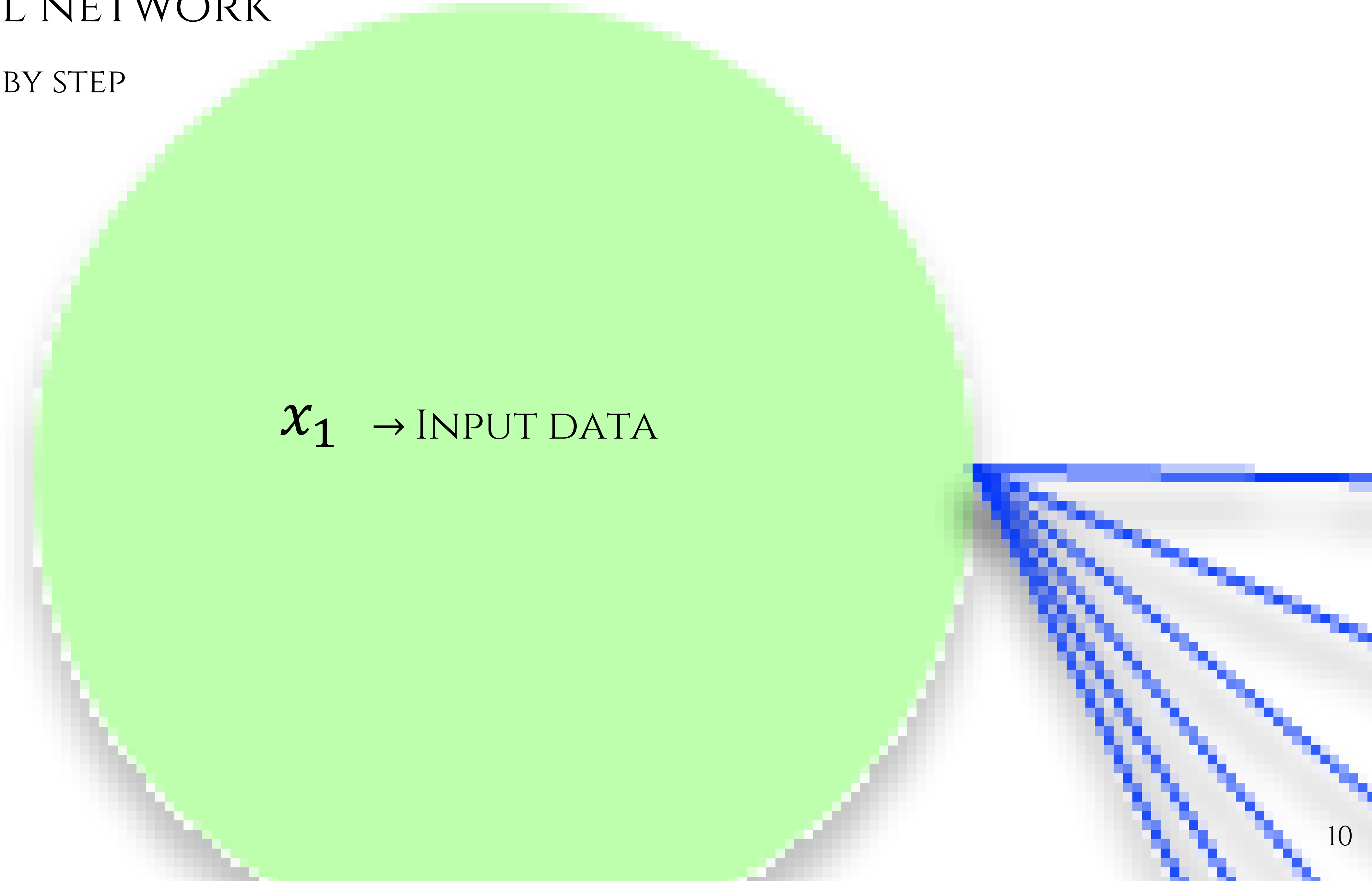


III- Neural network



III- Neural network

Epoch step by step



III- Neural network

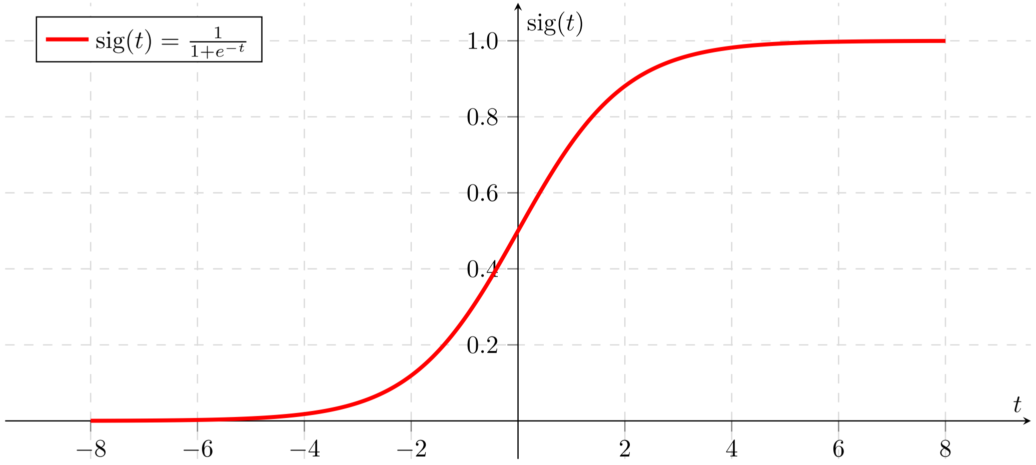
Epoch step by step

$$w_{11}(t)$$

(Where w_{ij} is the weight that connects the i^{th} input of the j^{th} neuron)

III- Neural network

Epoch step by step



Activation function

$$n_1 = sig\left(\sum_i w_{i1}(t)x_i + bias_1(t)\right)$$

III- Neural network

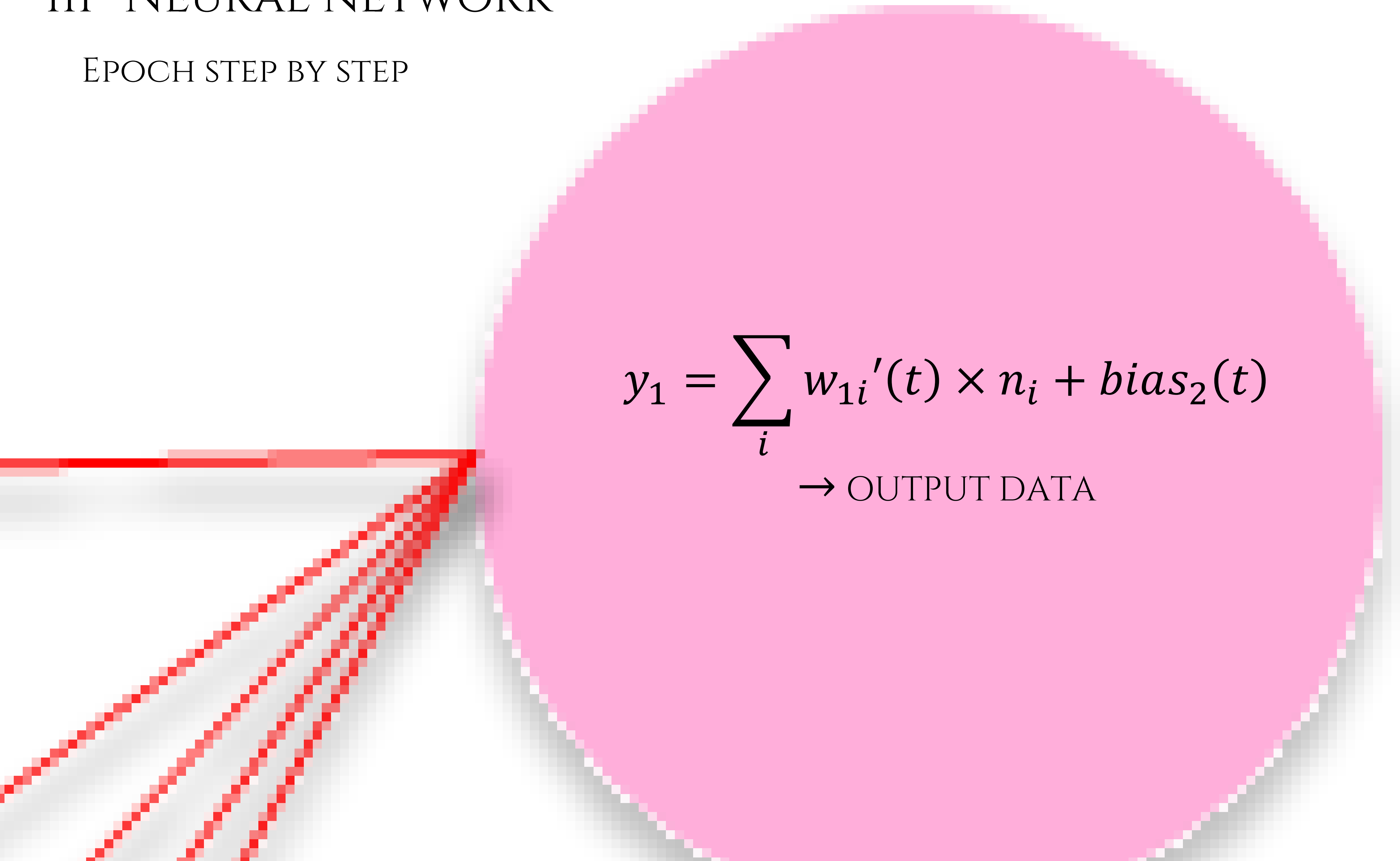
Epoch step by step

$$w'_{11}(t)$$

(Where w'_{ij} is the weight that connects the i^{th} output of the j^{th} neuron)

III- Neural network

Epoch step by step



$$y_1 = \sum_i w_{1i}'(t) \times n_i + bias_2(t)$$

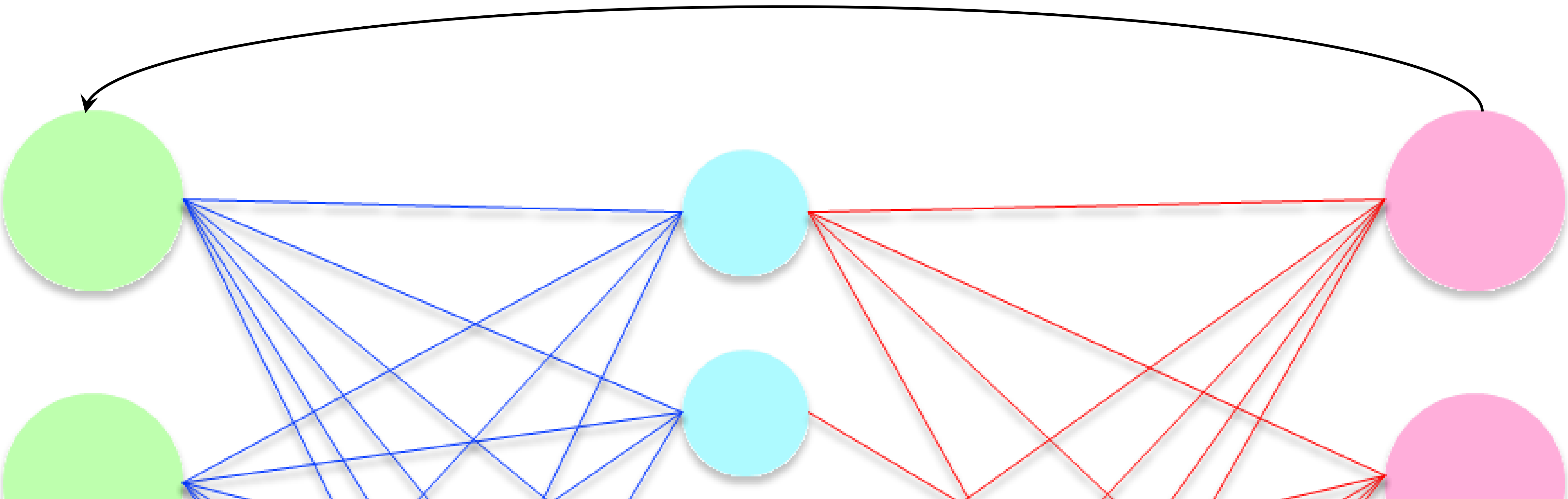
→ OUTPUT DATA

III- Neural network

Epoch step by step

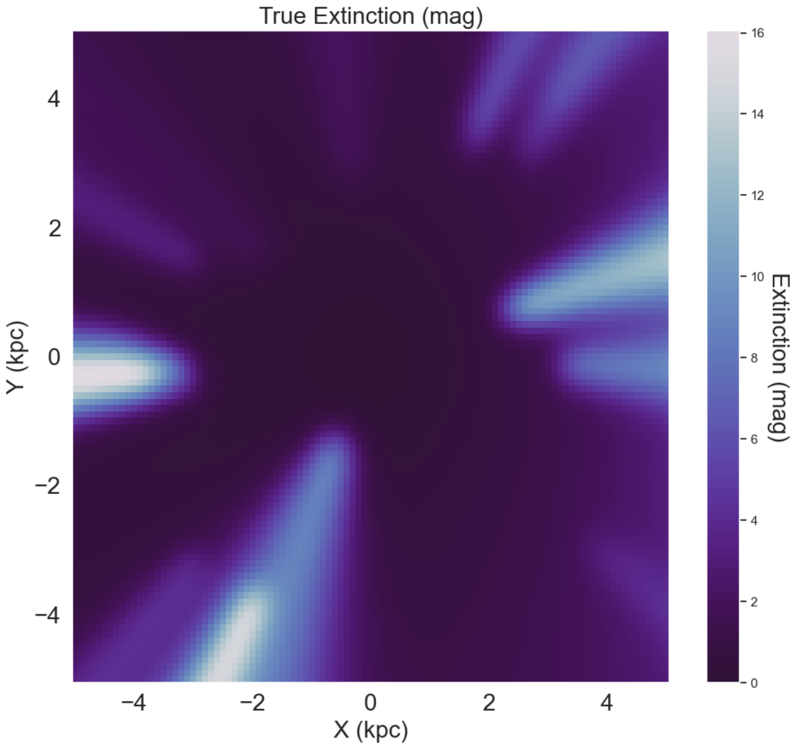
From the output and the target data,
we can find the loss function

Weights updating according to loss function

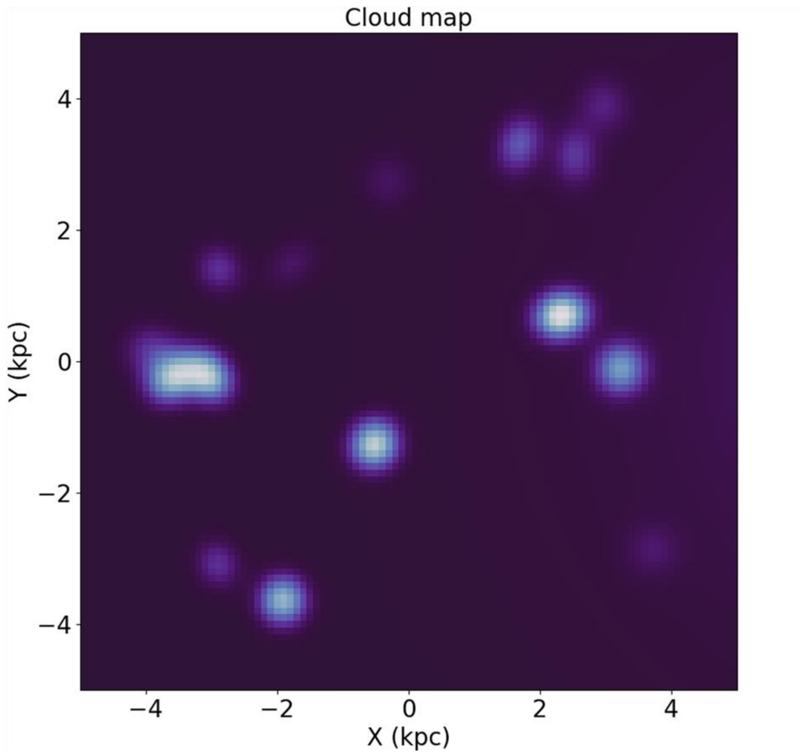


IV- Galaxy mapping problem

Input :



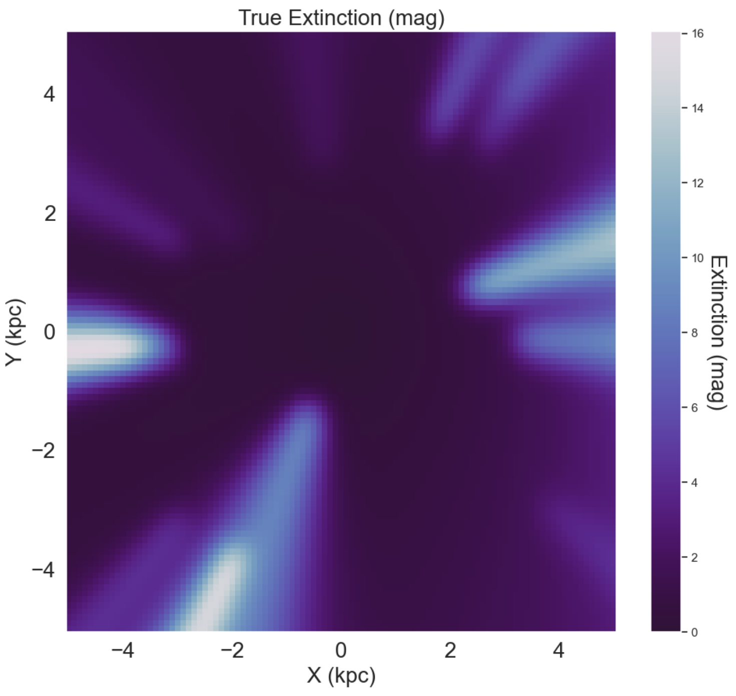
Output (ideal) :



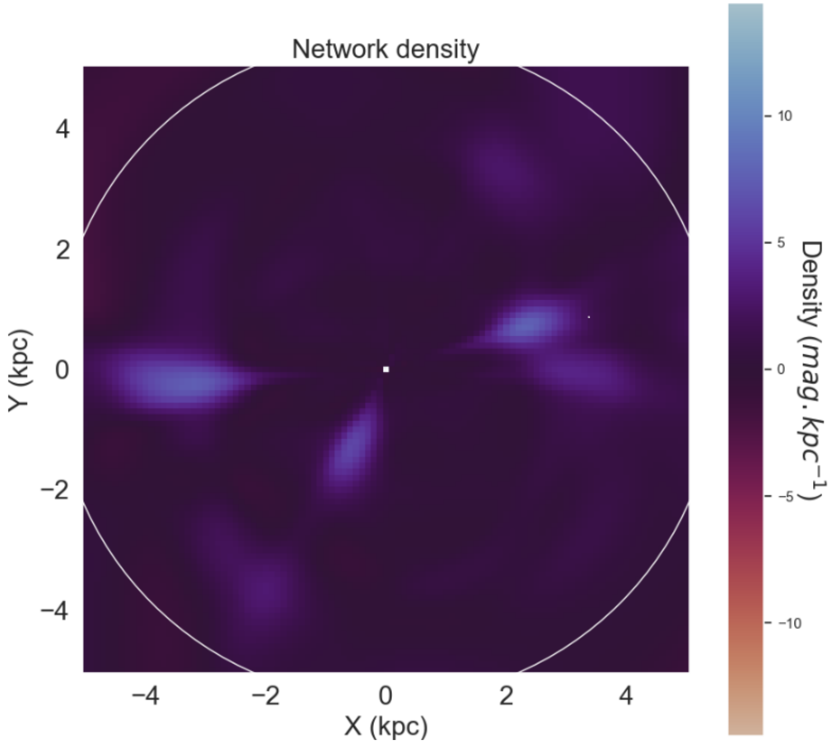
IV- Galaxy mapping problem

Learning process

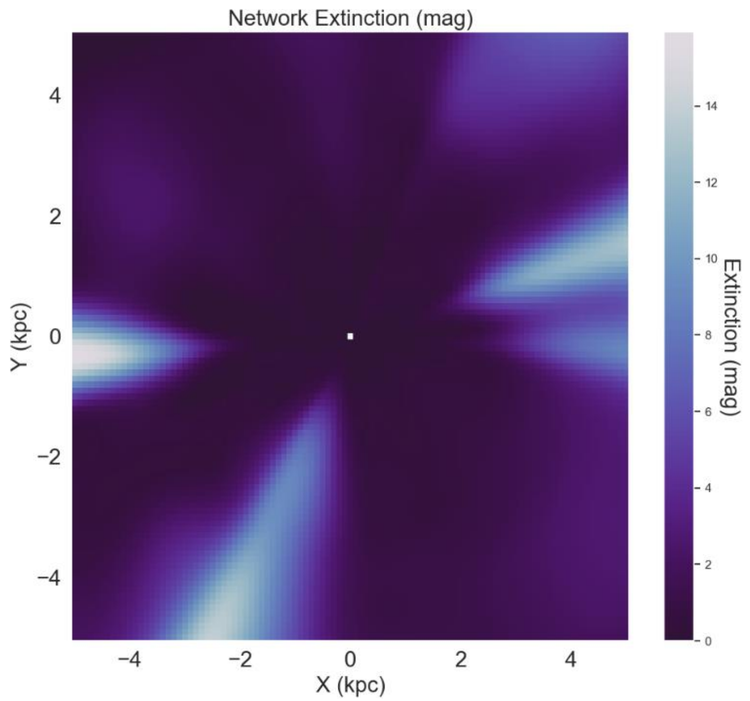
Input :



Output :



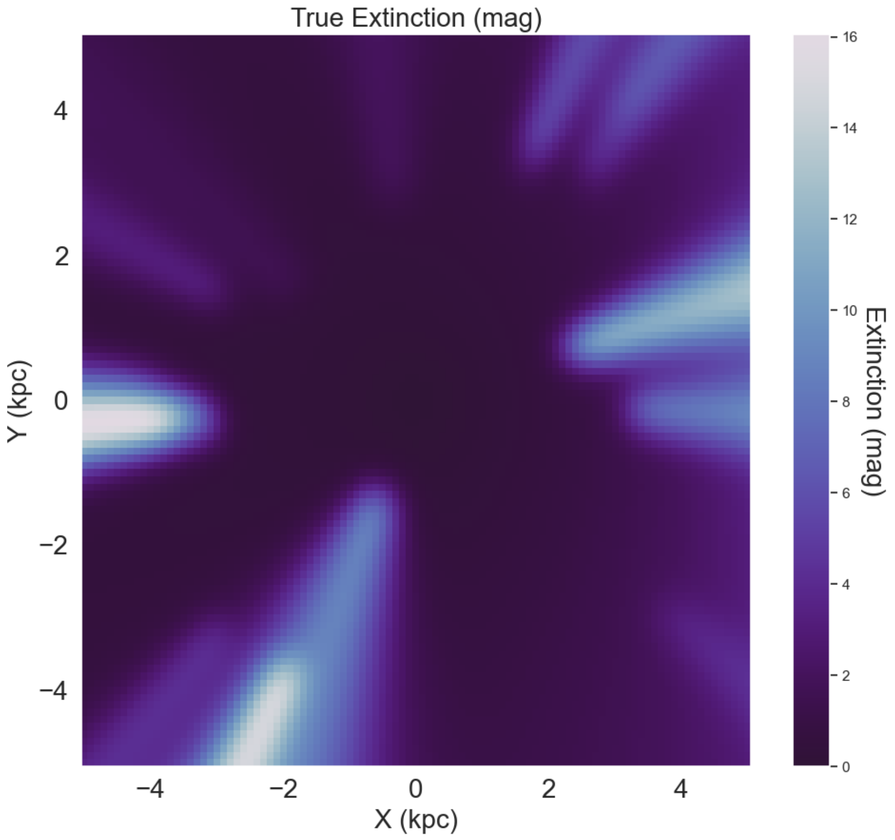
Integrated output :



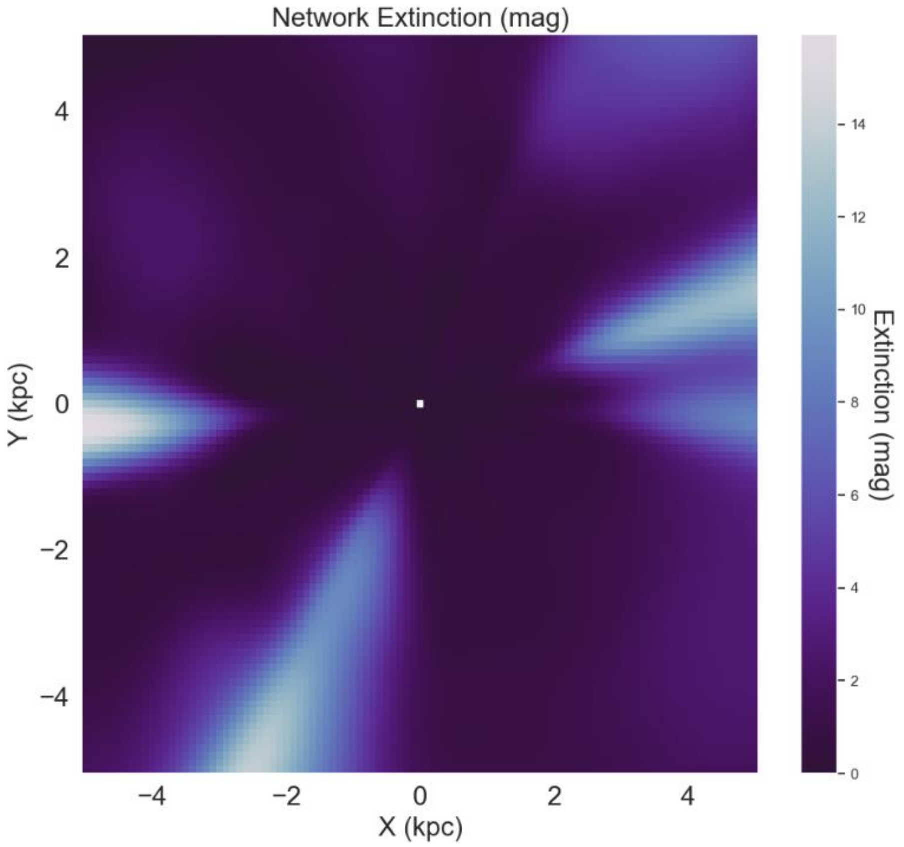
IV- Galaxy mapping problem

Learning process

Input :



Integrated output :



IV- Galaxy mapping problem

Parameters

Loss Function

Several loss functions :

$$f_{mse} = (y_{target} - y)^2$$

$$f_{\log likelihood} = \left(\frac{y_{target} - y}{\sigma}\right)^2$$

Reduction method

$$F_{mse,\, sum} = \sum f_{mse}$$

$$F_{mse,\, mean} = \langle f_{mse} \rangle$$

} Tends to 0

$$F_{\log likelihood,\, sum} = \sum f_{\log likelihood}$$

→ Tends to star number

$$F_{\log likelihood,\, mean} = \langle f_{\log likelihood} \rangle$$

→ Tends to 1

IV- Galaxy mapping problem

Parameters

Loss Function (Mean squared error, Log likelihood)

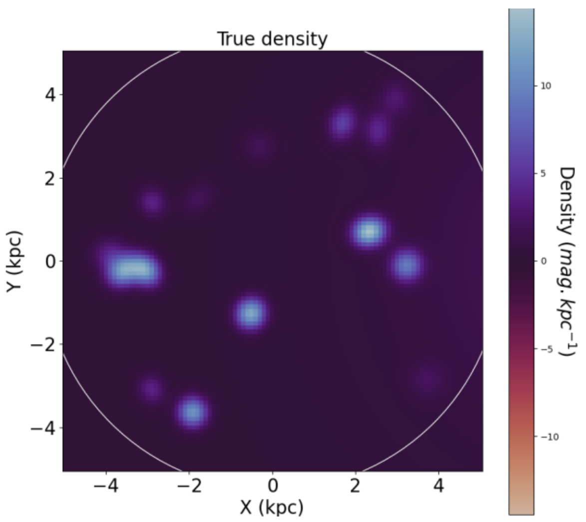
Reduction method (mean, sum)

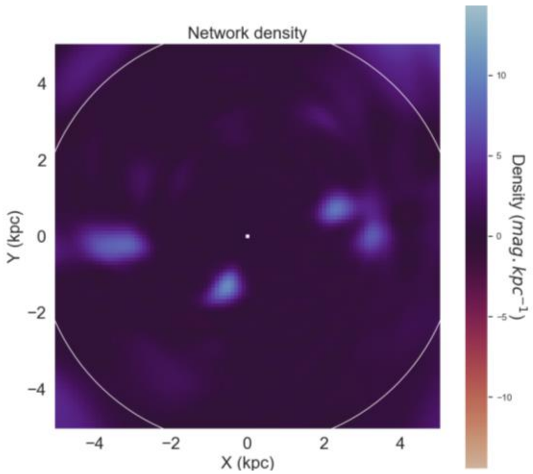
Epoch number

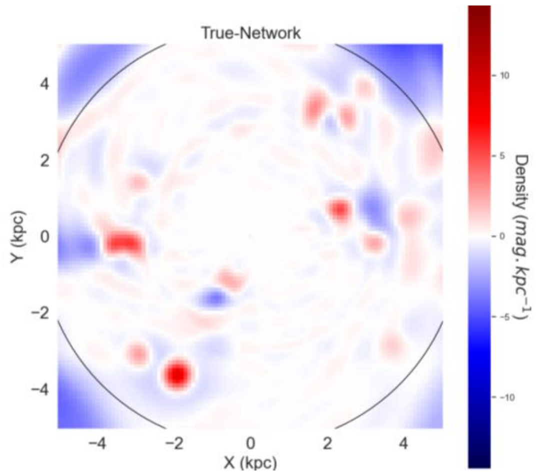
V- Results

Loss function

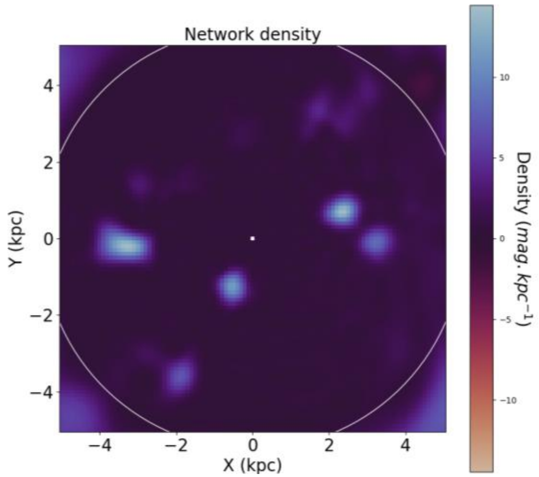
Density

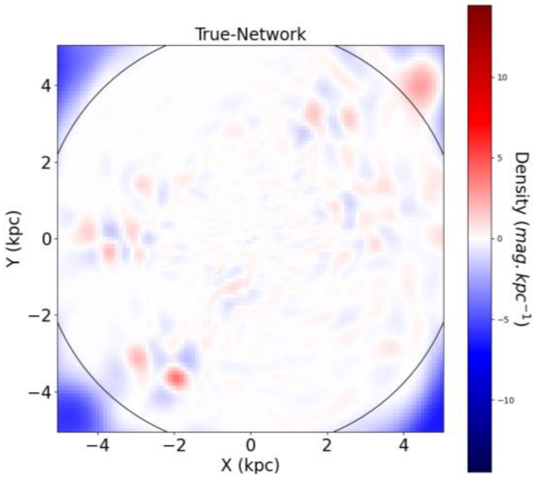






Log Likelihood



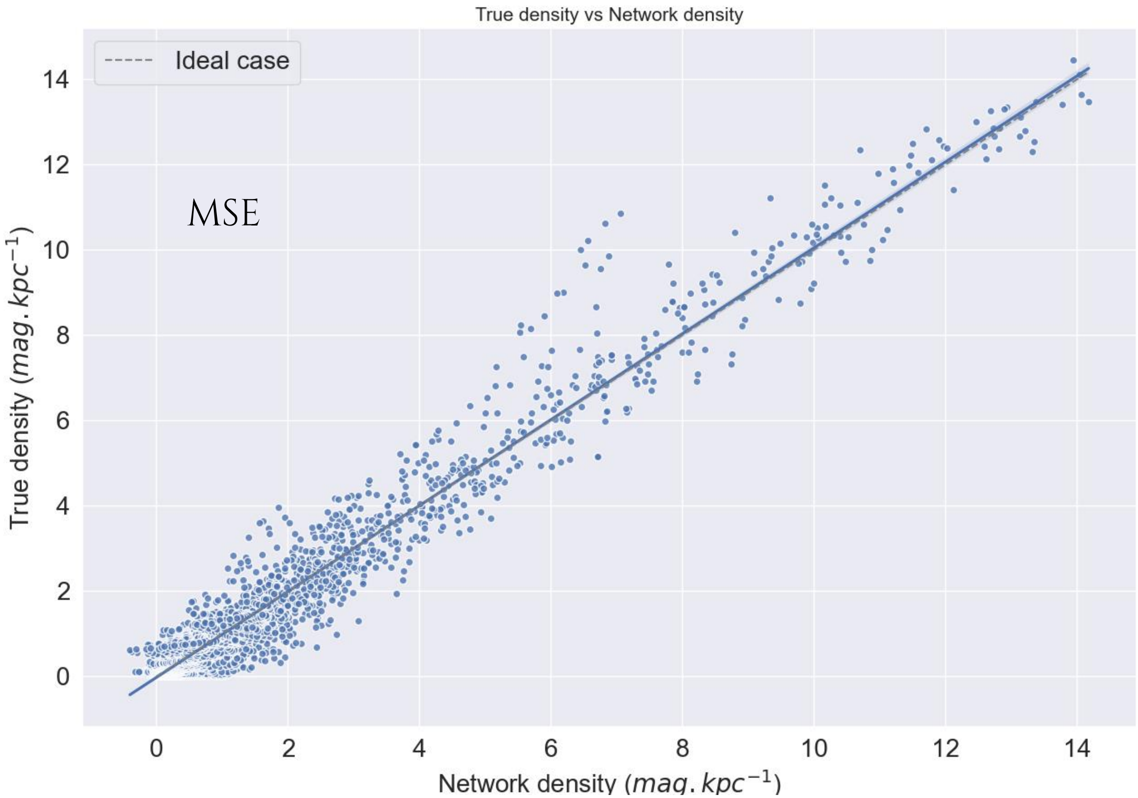


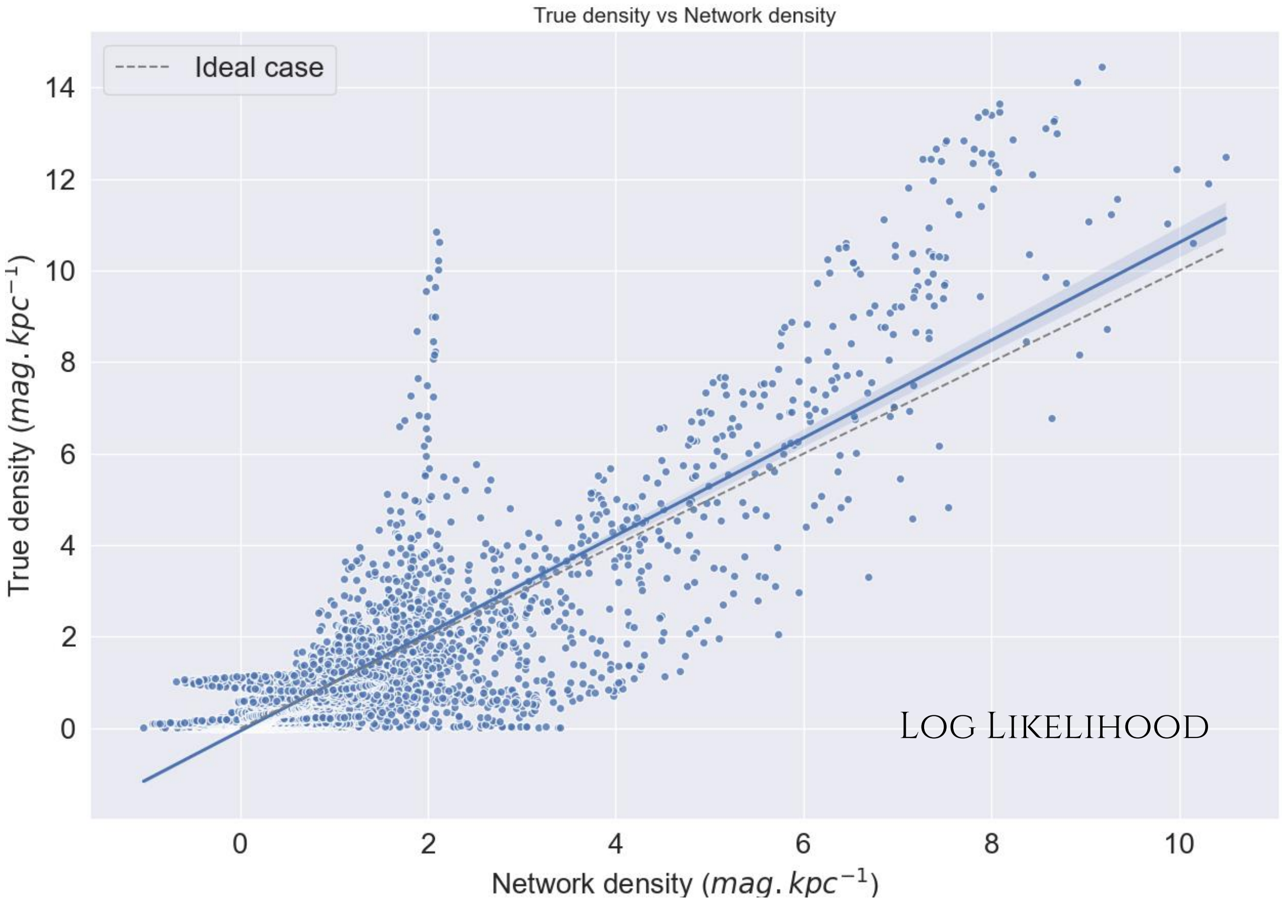
MSE

V- Results

Loss function

Dispersion diagram

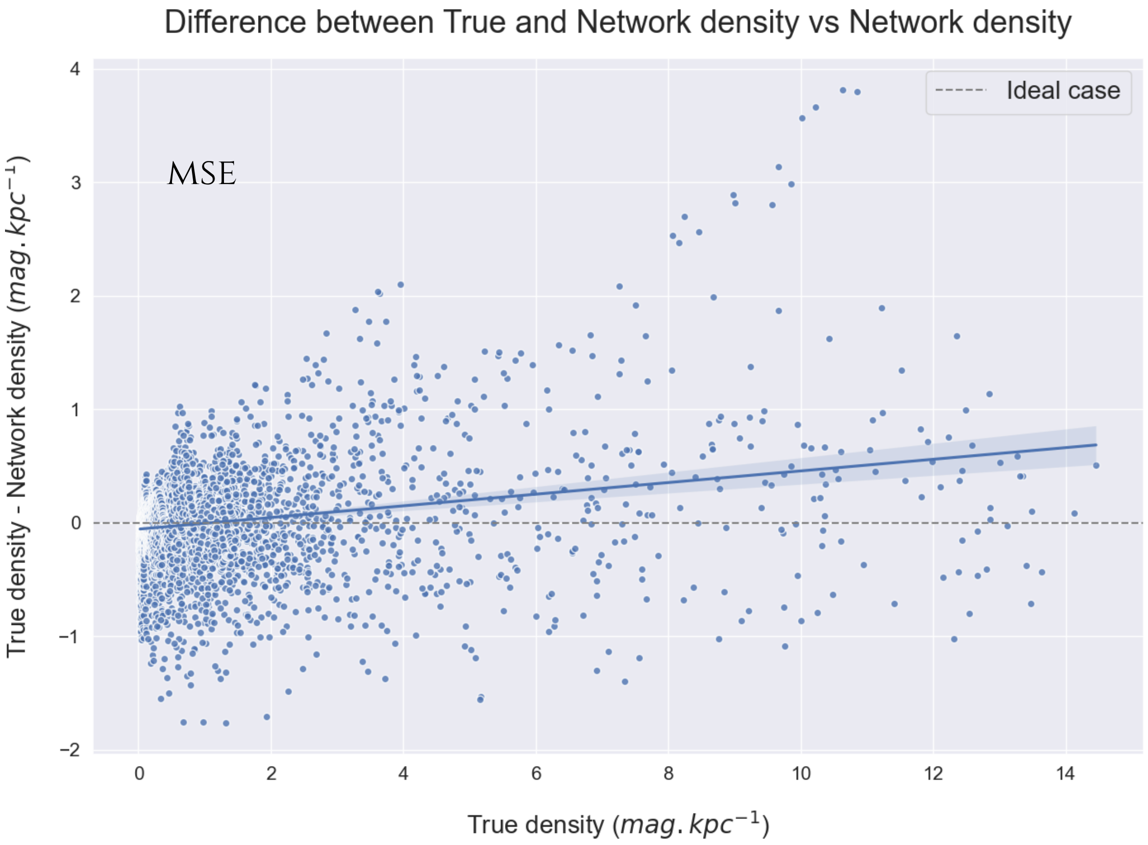


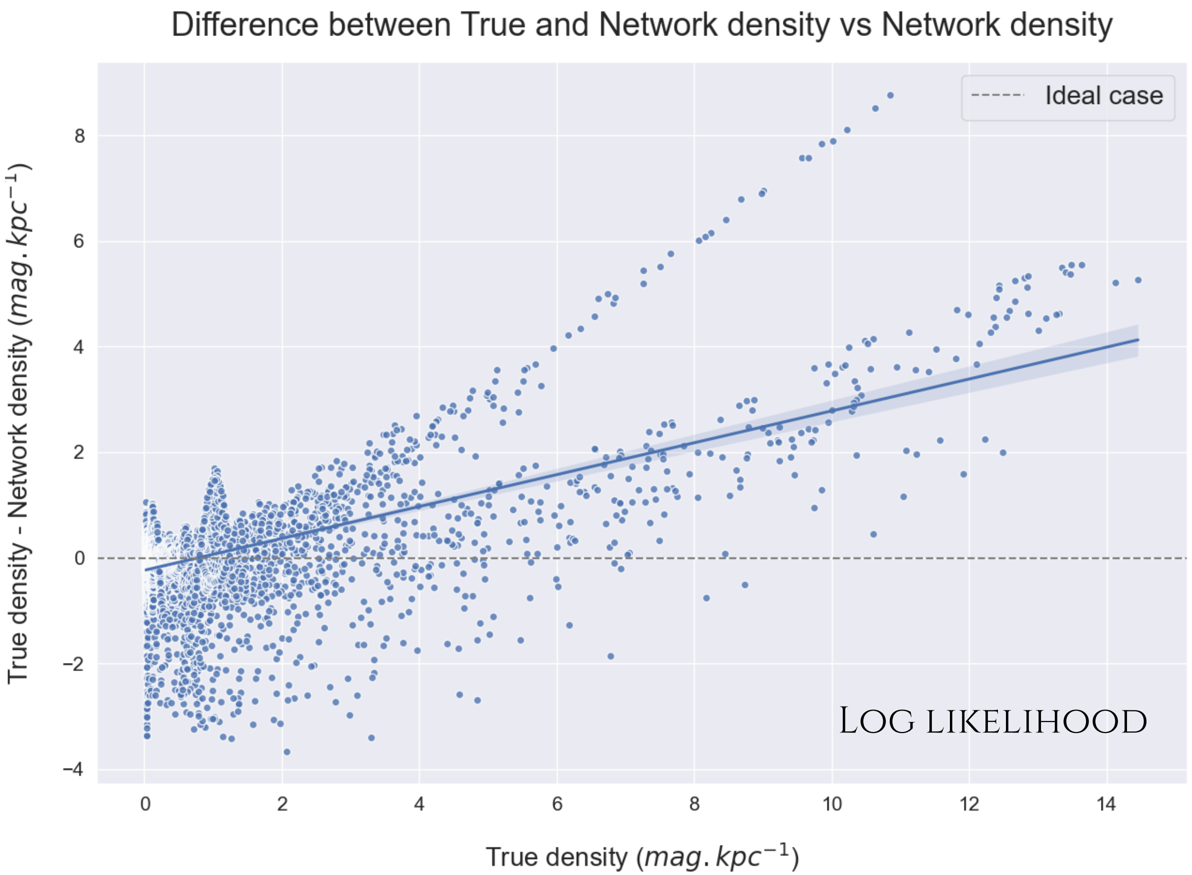


V- Results

Loss function

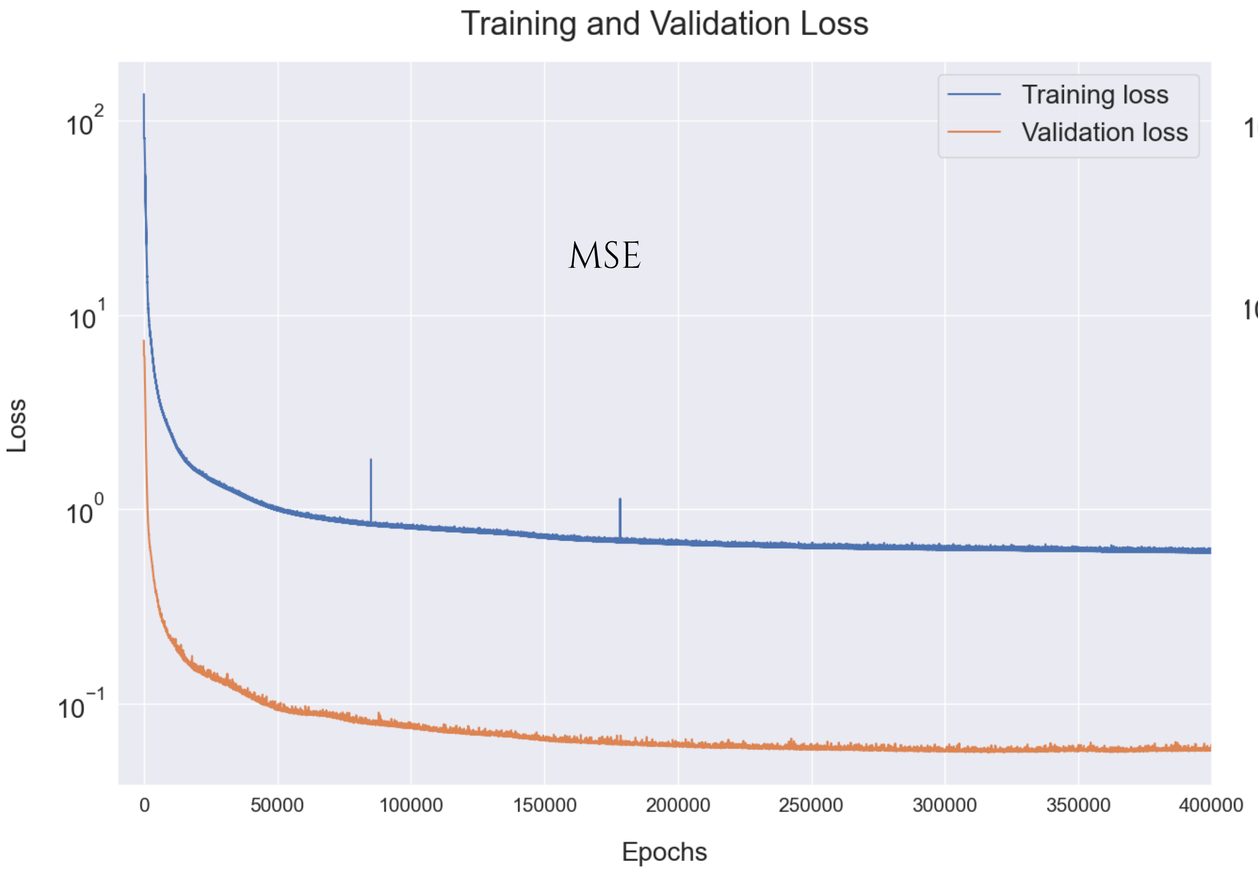
Residue analysis

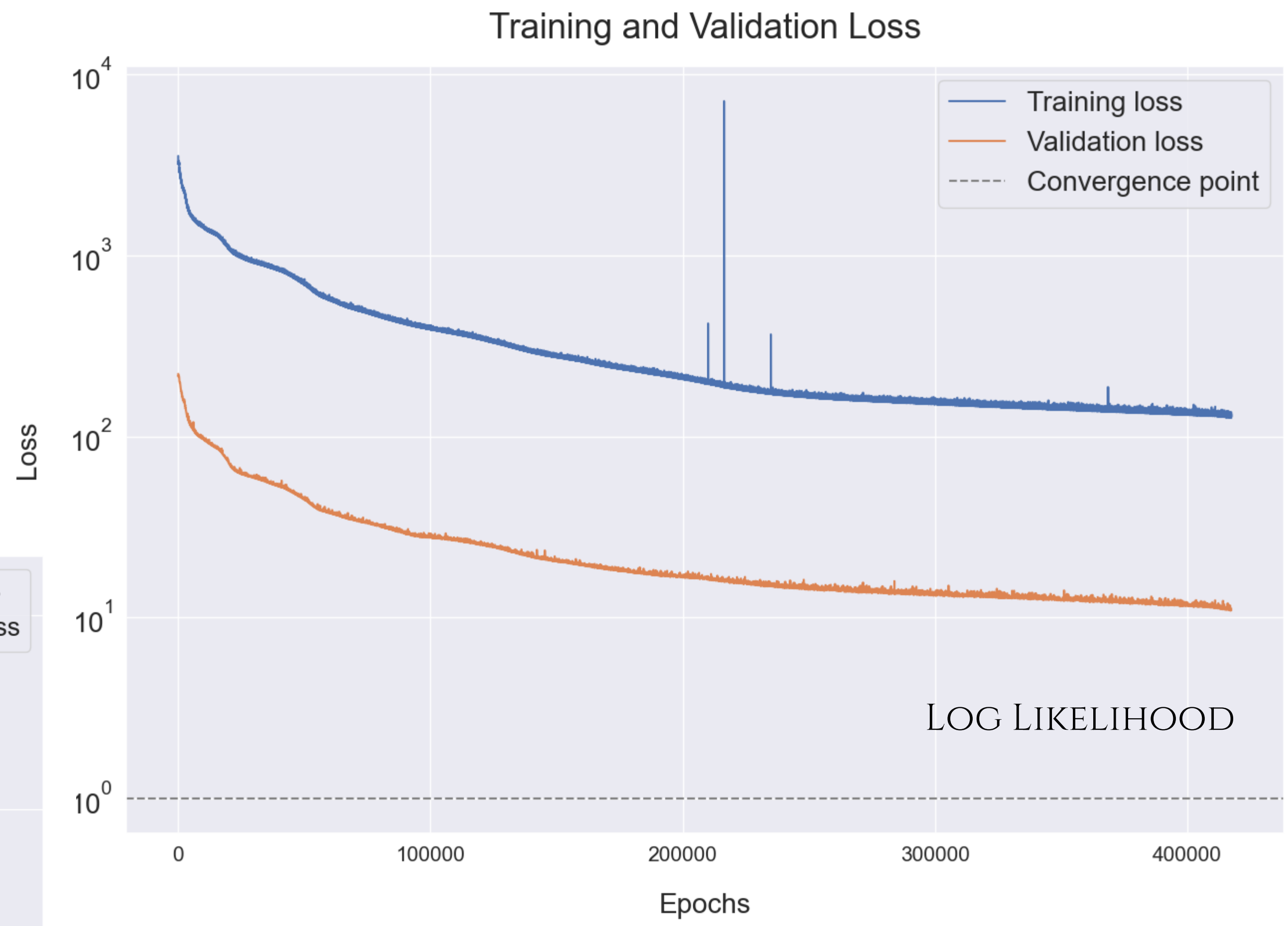




V- Results

Loss function

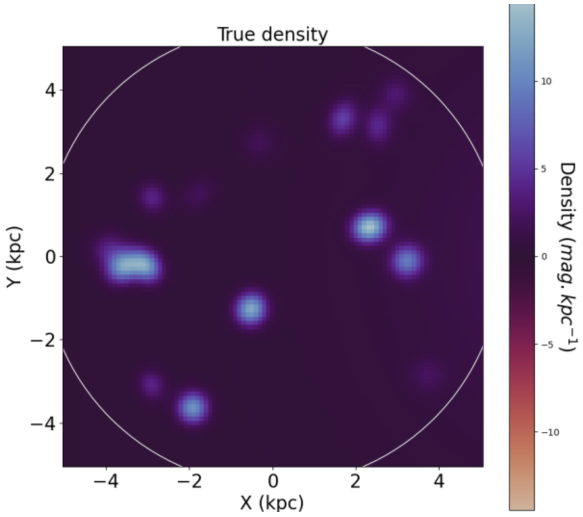


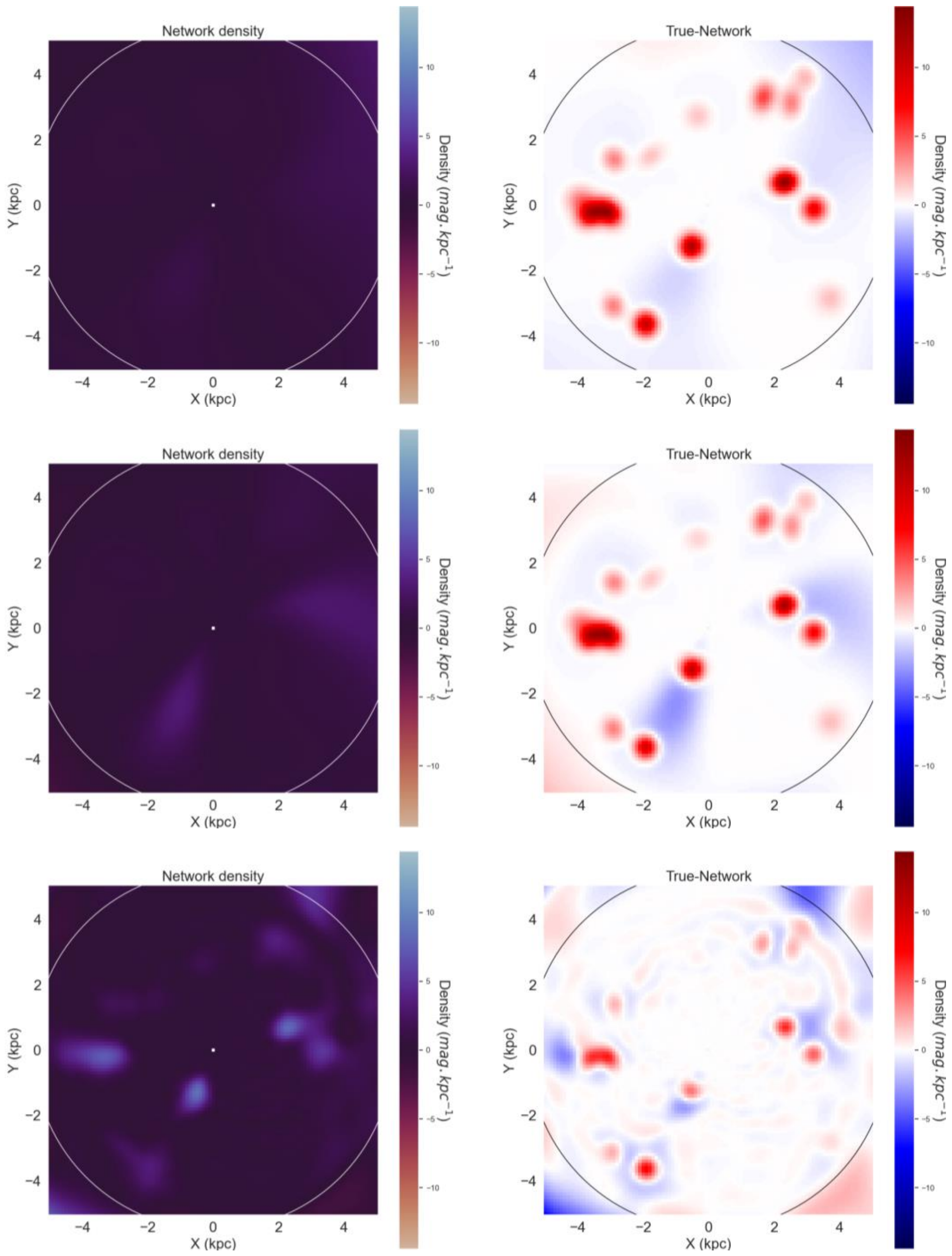


V- Results

Epoch

Density





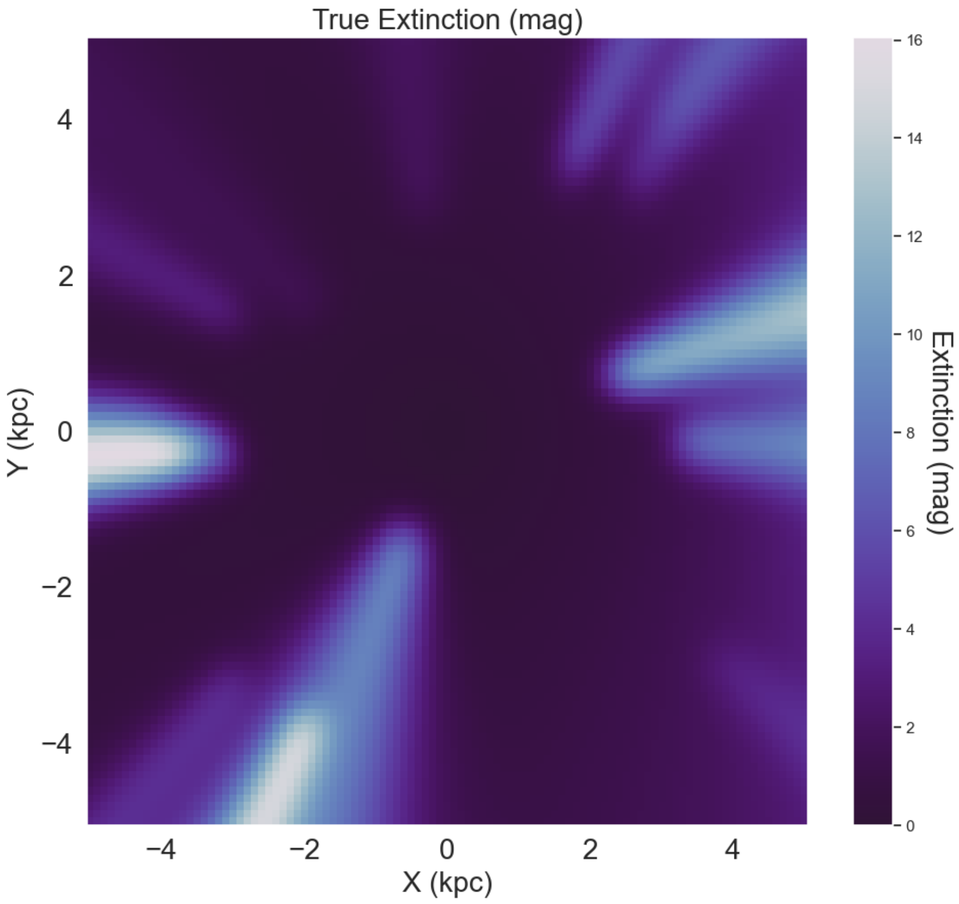
8 000 epochs

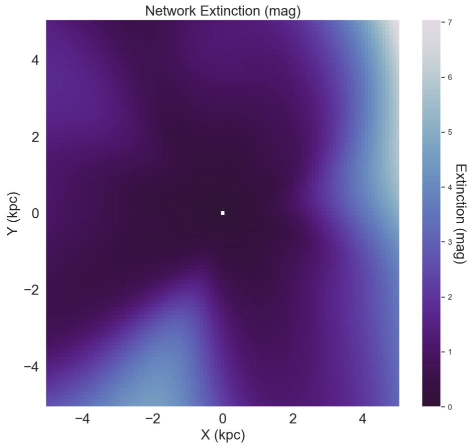
16 000 epochs

420 000 epochs

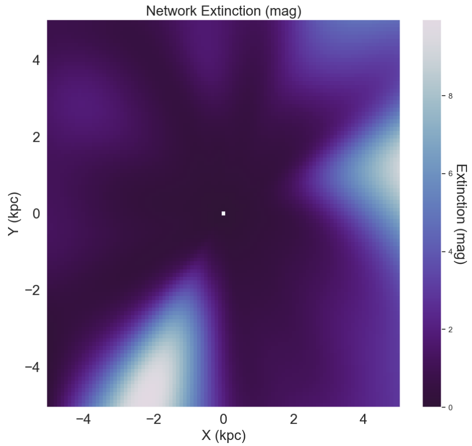
V- Results

Epoch

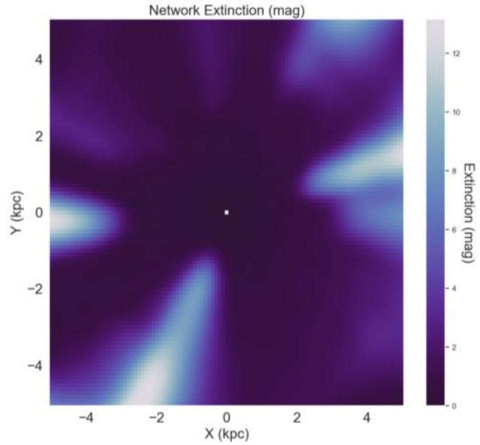




8 000 epochs



16 000 epochs

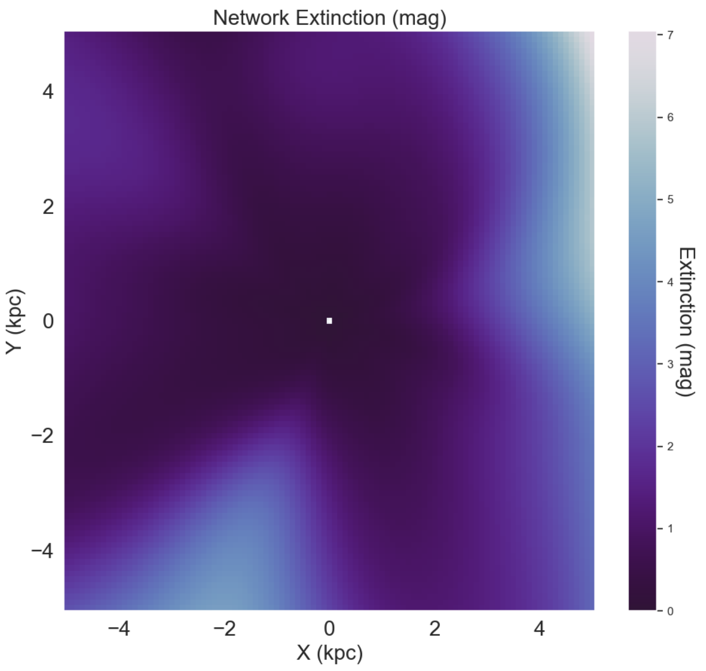


420 000 epochs

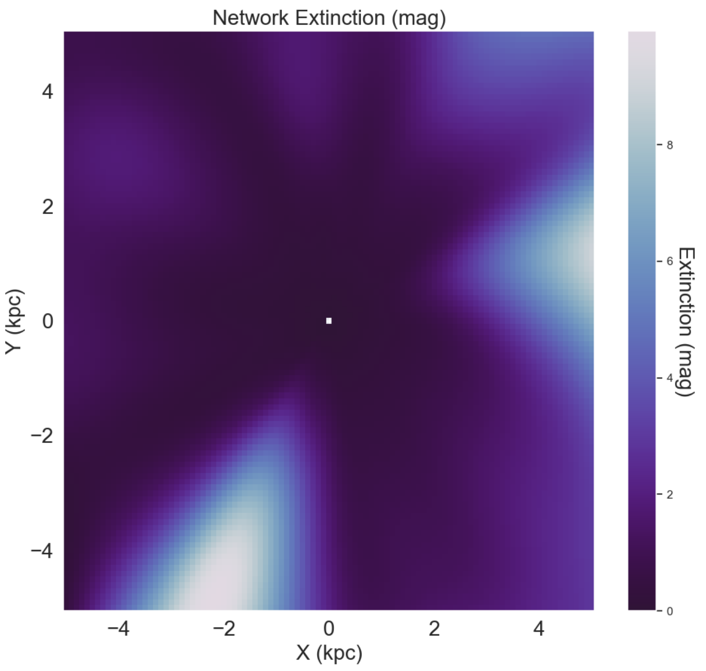
V- Results

Epoch

Extinction



8 000 epochs

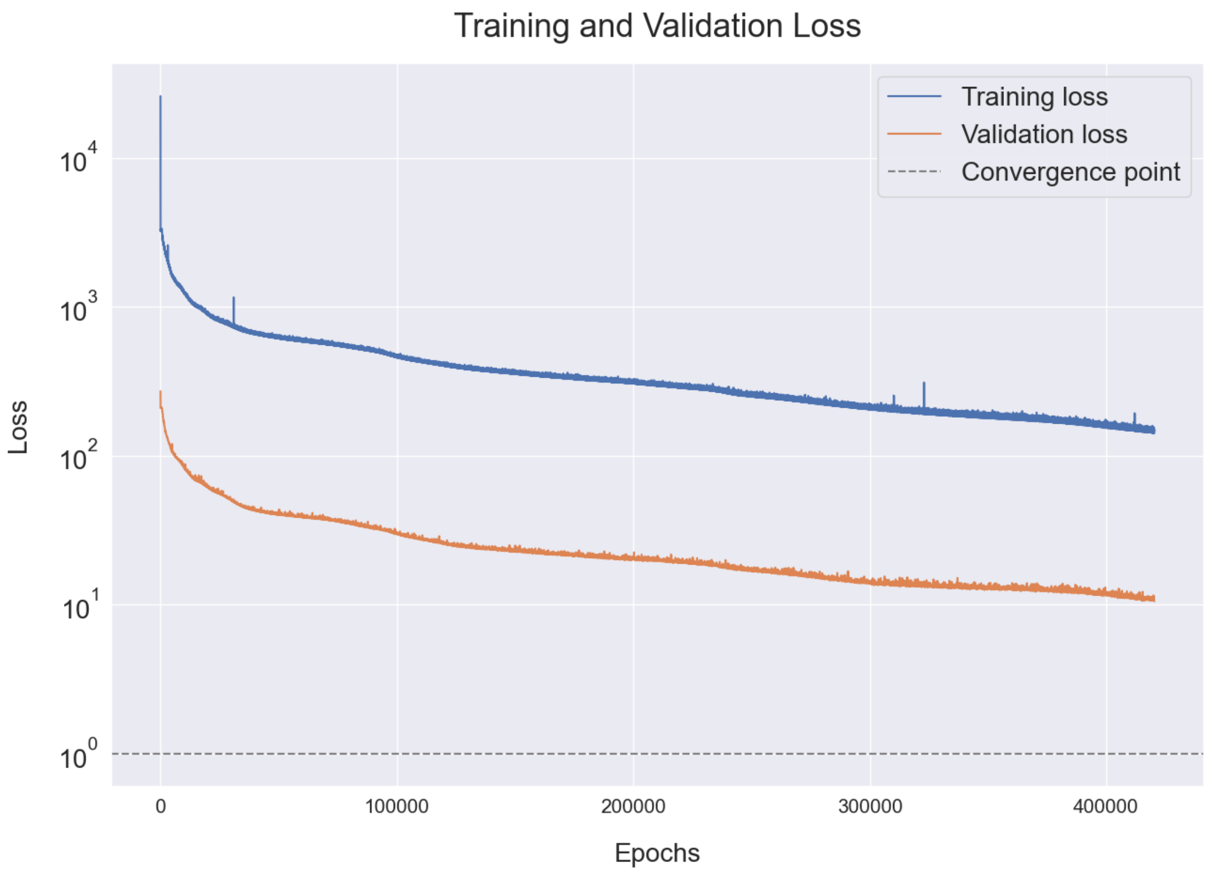


16 000 epochs

V- Results

Epoch

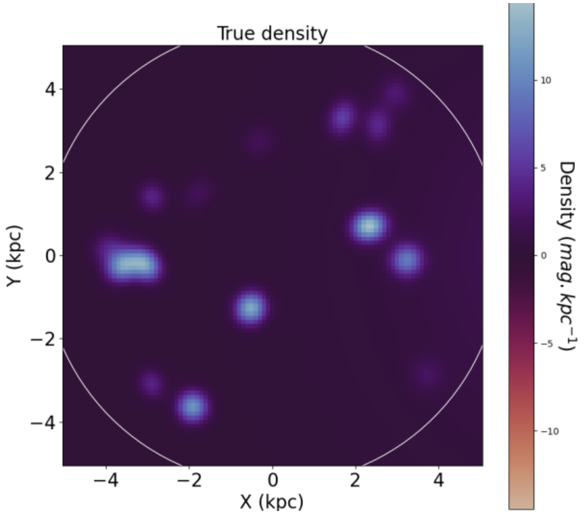
Extinction

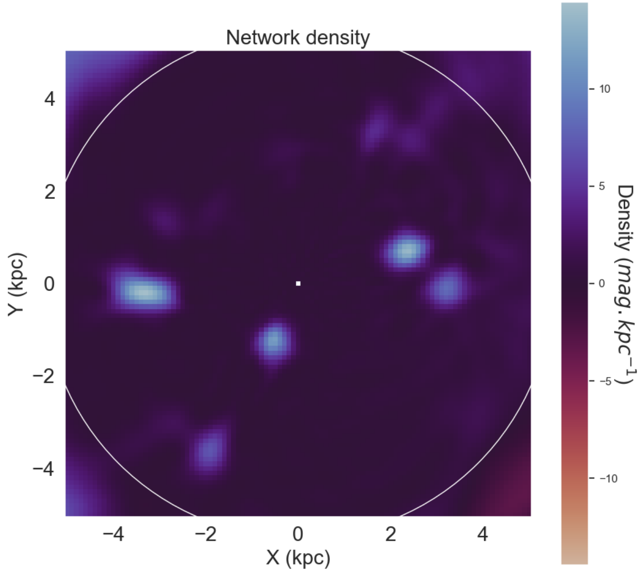


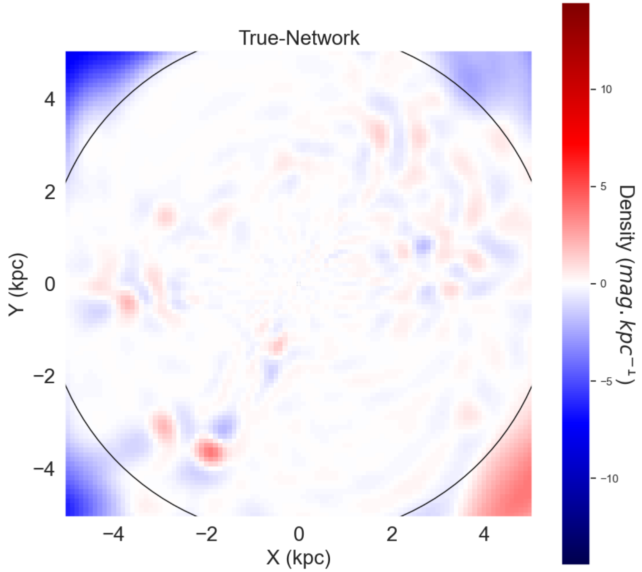
V- Results

Reduction method

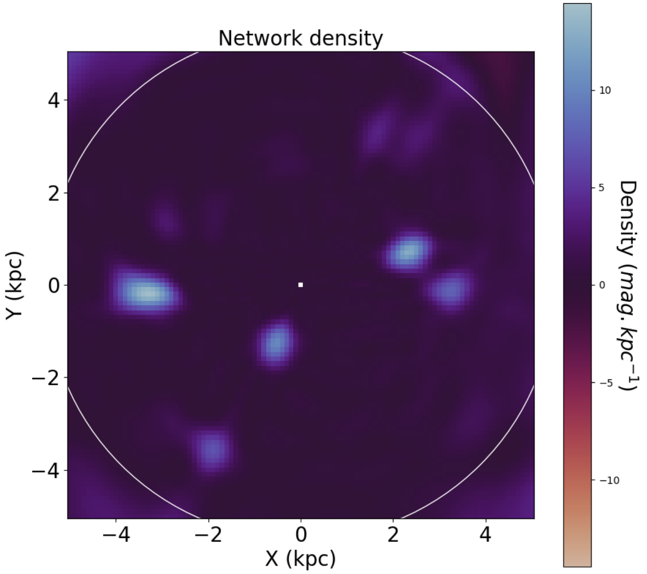
Density

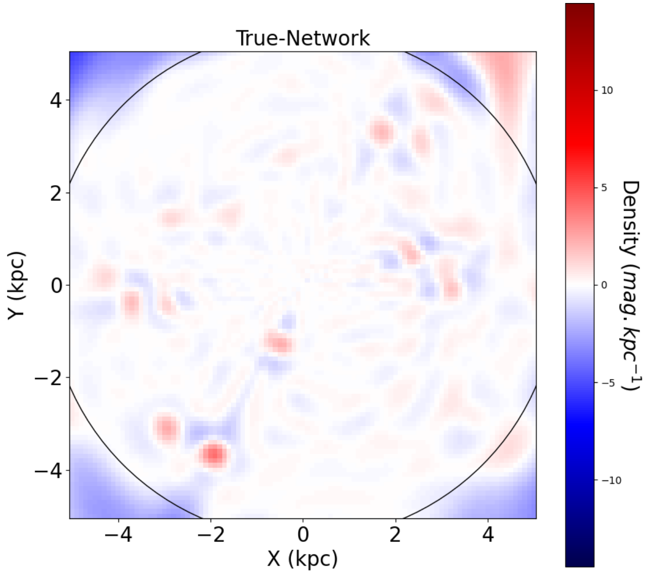






Sum & sum



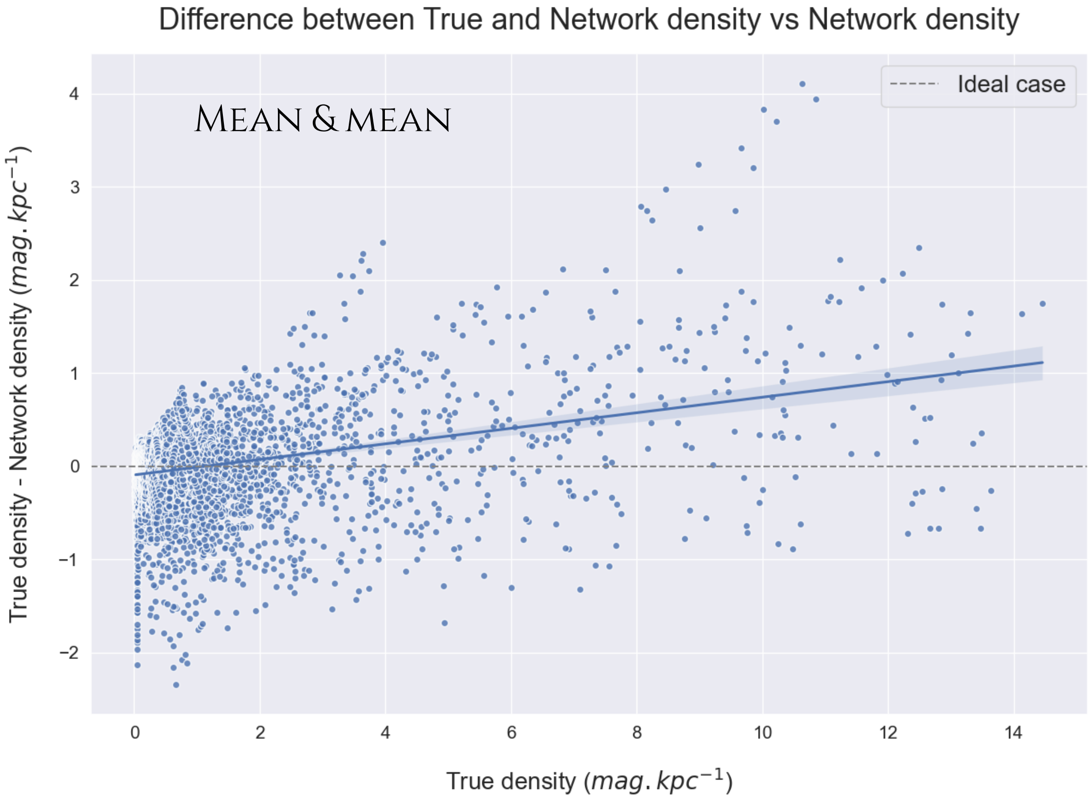


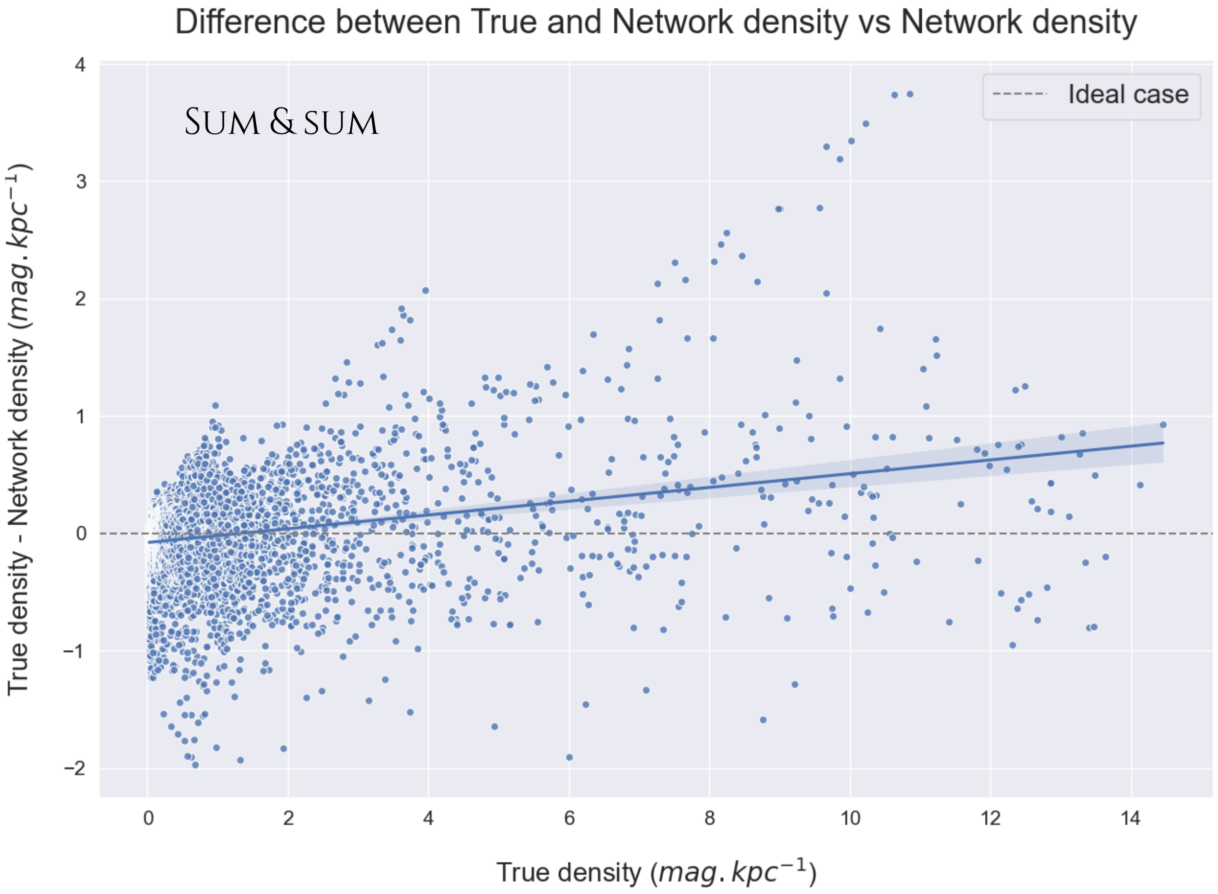
Mean & mean

V- Results

Reduction method

Residue analysis





VI- Conclusion

- Neural networks are timeworth way to solve some diffcult problems

- Some parameters are way more influential on the learning way of the NN

- Can be a significative computation time saving tool

VI- Conclusion

What next ?

- Extand the model to more complex neural networks

- Add astrophysical constraints

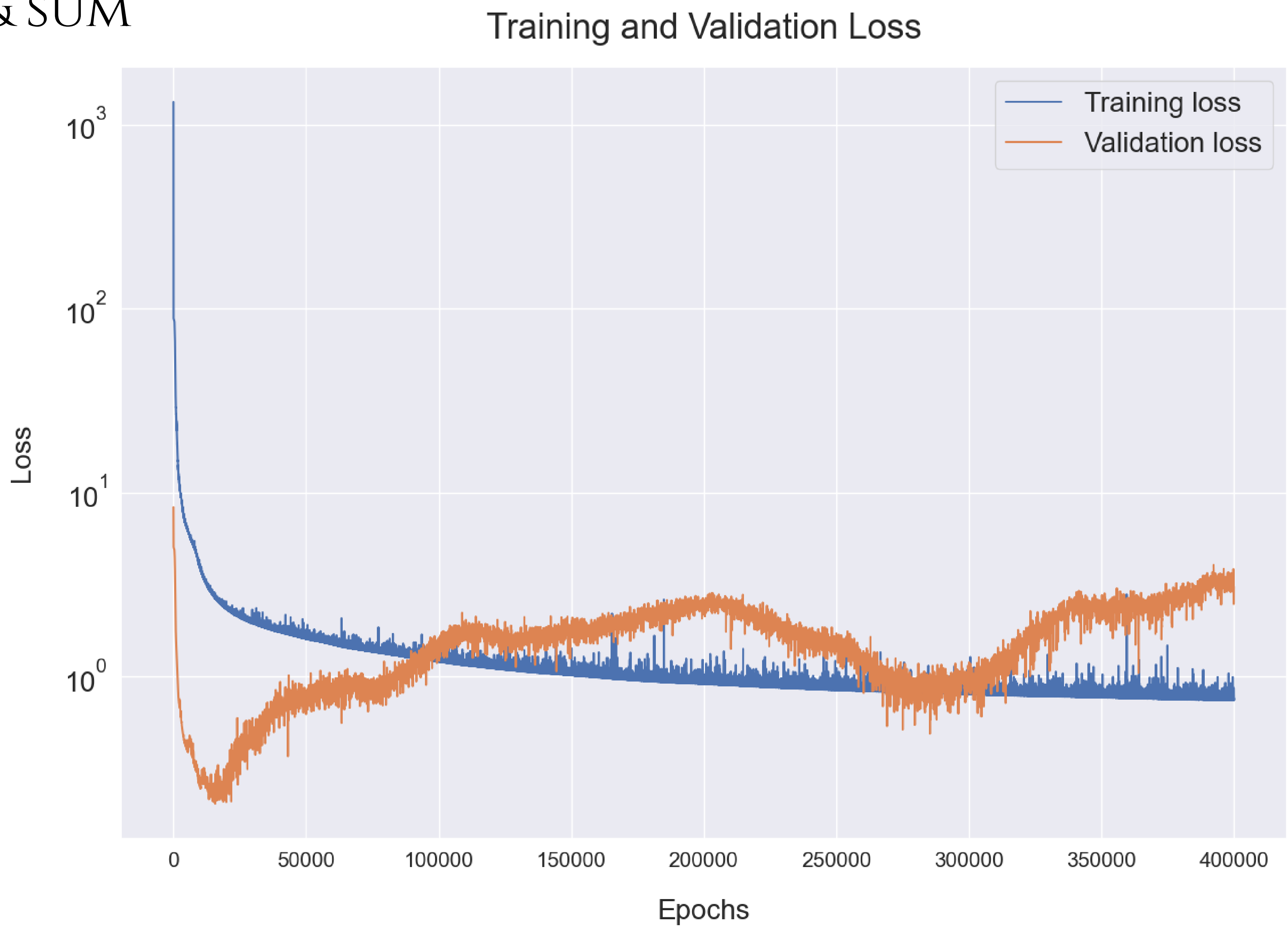
- Generalizee to 3D and Apply the method to real datas from GAÏA telescope

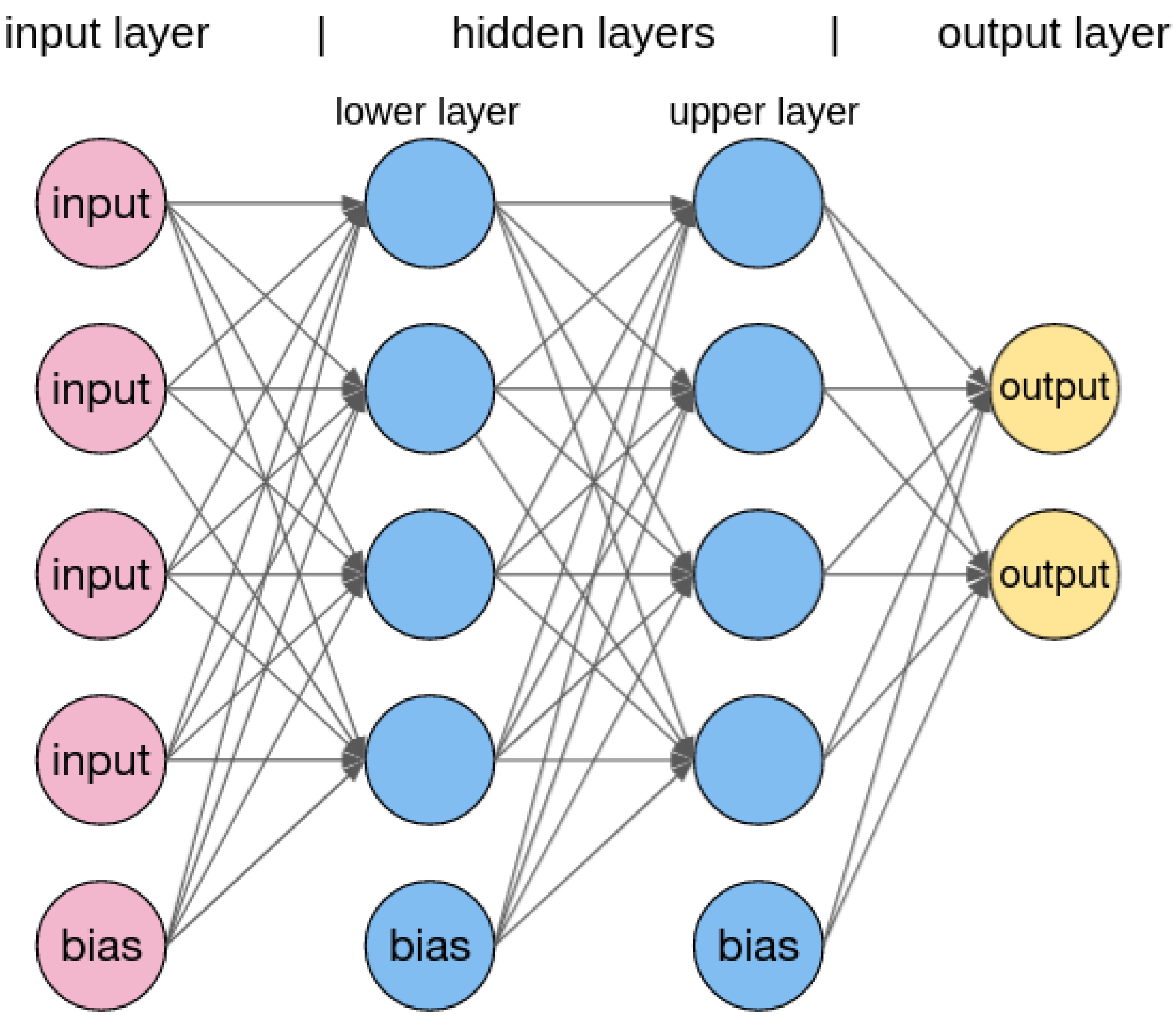
Bibliogrpahy

- S. Lloyd, R. A. Irani, M. Ahmadi, 2020, *Using Neural Networks for Fast Numerical Integration and Optimization*

- Thanks to A. Siebert for the code

MEAN & SUM


Training and Validation Loss
Training loss
Validation loss
Loss
10^3
10^2
10^1
10^0
0
50000
100000
150000
200000
250000
300000
350000
400000
Epochs



*https://miro.medium.com/v2/resize:fit:563/1*4_BDTvgB6WoYVXyxO8lDGA.png*

$$\hat{I}(f) = b^{(2)}(\beta_1 - \alpha_1) + \sum_{j=1}^{k} w_j^{(2)} \left[(\beta_1 - \alpha_1) + \frac{\Phi_j}{w_{1j}^{(1)}} \right]$$

$$\Phi_j = Li_1\left[-e^{-b_j^{(1)} - w_{1j}^{(1)} * \alpha_1}\right] - Li_1\left[-e^{-b_j^{(1)} - w_{1j}^{(1)} * \beta_1}\right]$$

$$Li_1(z) = \sum_{k=1}^{\infty} \frac{z^k}{k}$$