

# Machine Learning Modeling of the Microstructural Evolution of Strained Materials

Candidate: Angelo Monteleone

ID number: 884901

Supervisor: Prof. Francesco Montalenti

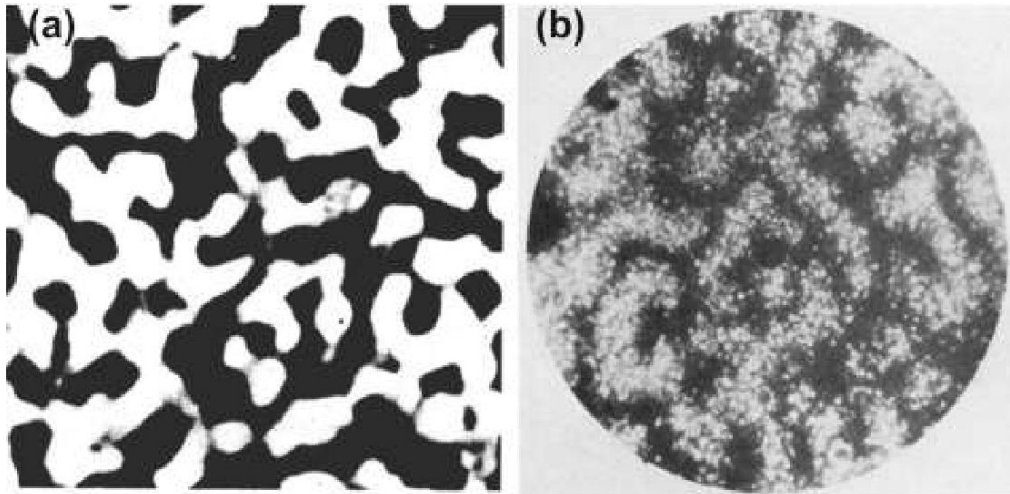
Co-supervisor: Dott. Daniele Lanzoni

Academic year 2023/2024

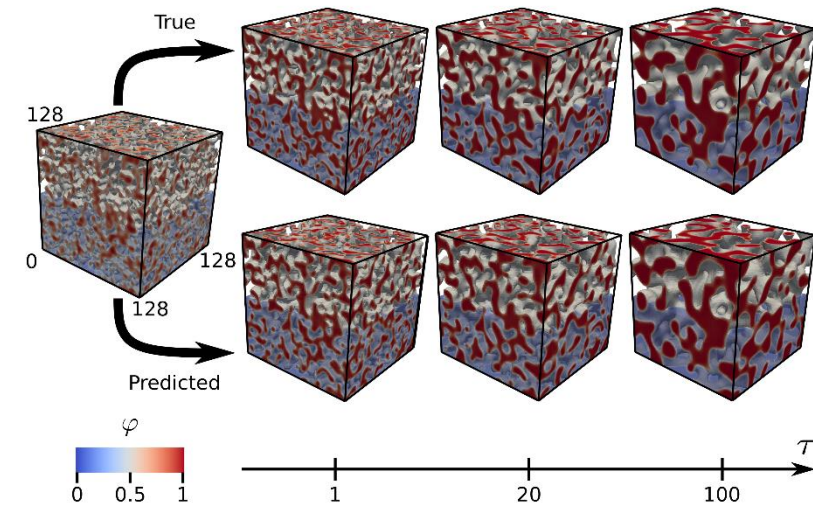


# Motivations of the study

- **Spontaneous phase separation** through spinodal decomposition.
  - Influence on alloy mechanics.
- Simple hypotheses, but complex phenomena.



W.A. Soffa, D. E. Laughlin, 8 - *Diffusional Phase Transformations in the Solid State*, Physical Metallurgy (Fifth Edition), Elsevier, 2014



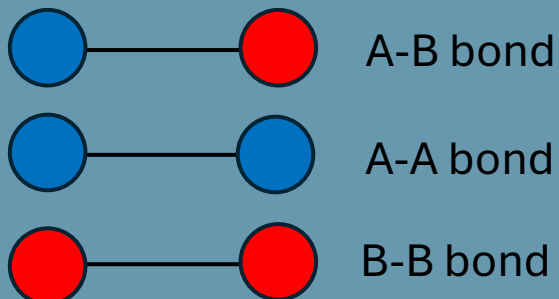
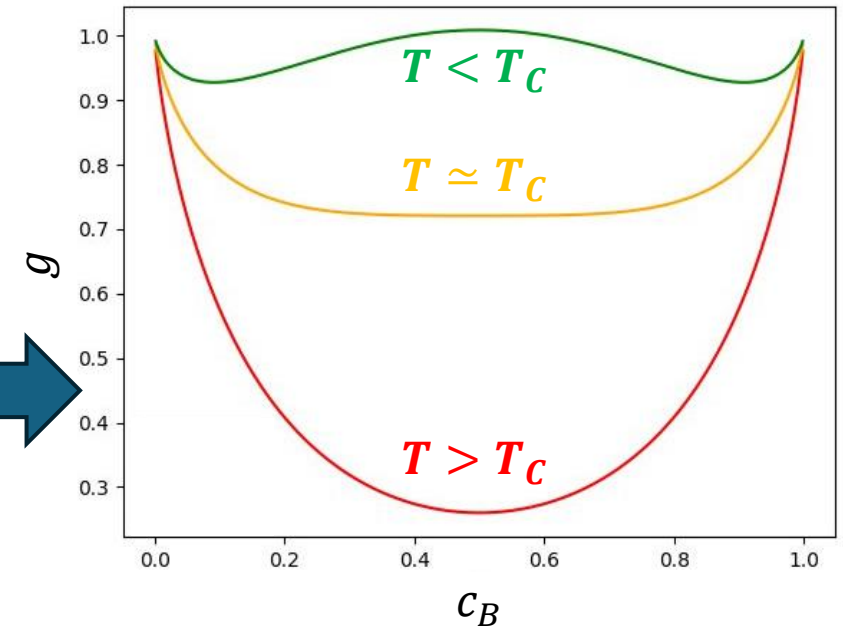
- **ML methods** for evolution prediction: already explored by the research group.
- **Goals:** strain effects and parameter extraction.

D. Lanzoni et al. *Extreme time extrapolation capabilities and thermodynamic consistency of physics-inspired Neural Networks for the 3D microstructure evolution of materials*. *Mach. Learn.: Sci. Technol.* 2024

# Phase separation: hypotheses

- **Two atomic species A and B** with the same crystal structure, concentrations  $c_A$  and  $c_B$
- Constant temperature and pressure
- **Driving potential:** Gibbs free energy

$$g = \underbrace{c_A g_A^0 + c_B g_B^0}_{\text{Pure components}} + \underbrace{Z c_A c_B \Omega}_{\text{Enthalpic term}} + \underbrace{RT(c_A \log(c_A) + c_B \log(c_B))}_{\text{Entropic term}}$$

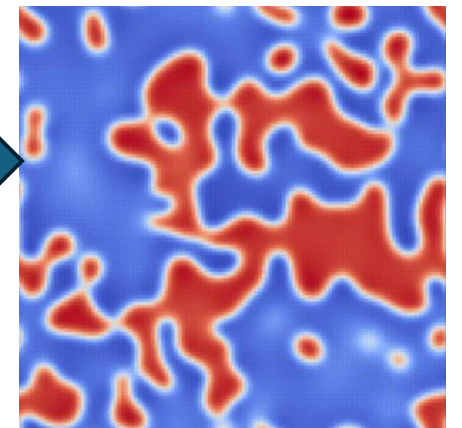


The sign of

$$\Omega = e_{AB} - \frac{1}{2}(e_{AA} + e_{BB})$$

enables **phase separation**

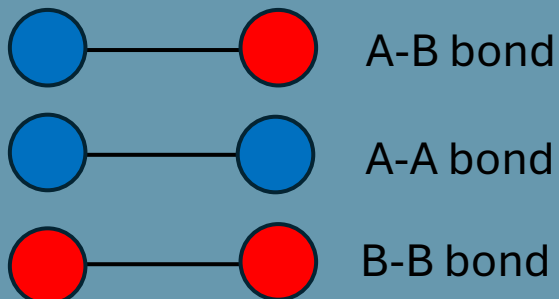
If  $\Omega > 0$



# Phase separation

- **Two atomic species A and B** with the same crystal structure, concentrations  $c_A$  and  $c_B$
- Constant temperature and pressure
- **Driving potential:** Gibbs free energy

$$g = \underbrace{c_A g_A^0 + c_B g_B^0}_{\text{Pure components}} + \underbrace{Z c_A c_B \Omega}_{\text{Enthalpic term}} + \underbrace{RT(c_A \log(c_A) + c_B \log(c_B))}_{\text{Entropic term}}$$



The sign of

$$\Omega = e_{AB} - \frac{1}{2}(e_{AA} + e_{BB})$$

enables **phase separation**

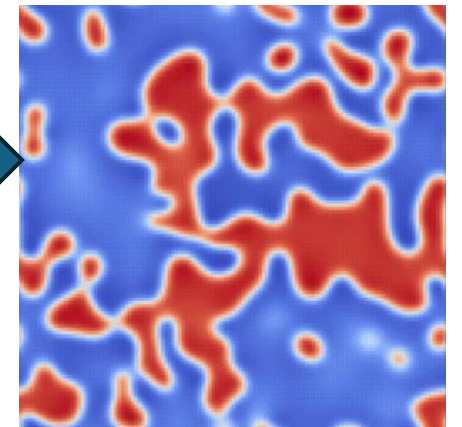
- **Cahn-Hilliard** equation:

$$\frac{\partial \phi}{\partial t} = \nabla \cdot (M \nabla (\frac{\delta G}{\delta \phi}))$$

- **Ginzburg-Landau:** free energy functional

$$G[\phi] = \int_{\Omega} [k |\nabla \phi|^2 + g_B(\phi)] d^3x$$

If  $\Omega > 0$





# Coherent spinodal decomposition dynamics

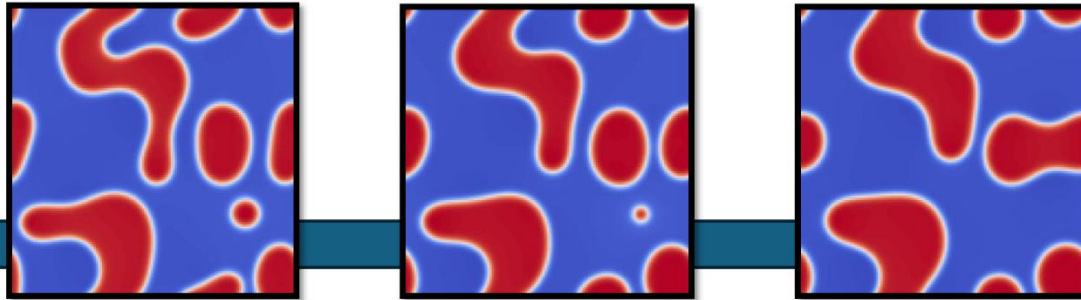
$$G[\varphi] = \int_{\Omega} [k|\nabla\varphi|^2 + g_B(\varphi) + \rho_{\eta}(\varphi)]d^3x$$

$$\eta = \frac{l_{\alpha} - l_{\beta}}{l_{\beta}}$$

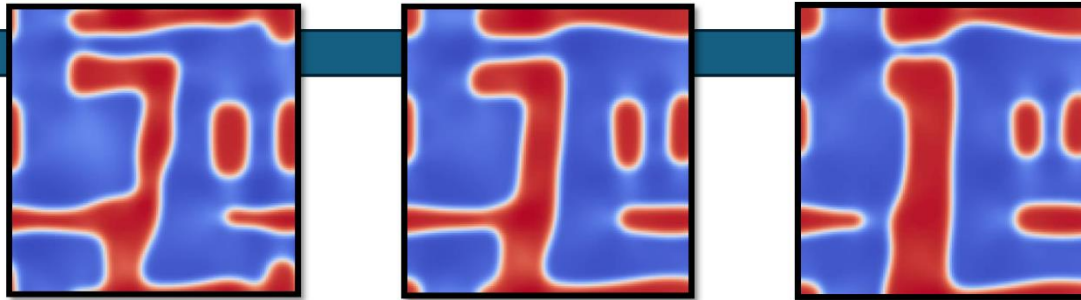
**Elastic effects** → new energy term, elastic energy density  $\rho_{\eta}(\varphi)$ .

- Dependence on the **concentration**  $\varphi$
- Parametrized by the **lattice mismatch**  $\eta$ , which shapes the microstructures that develop over time

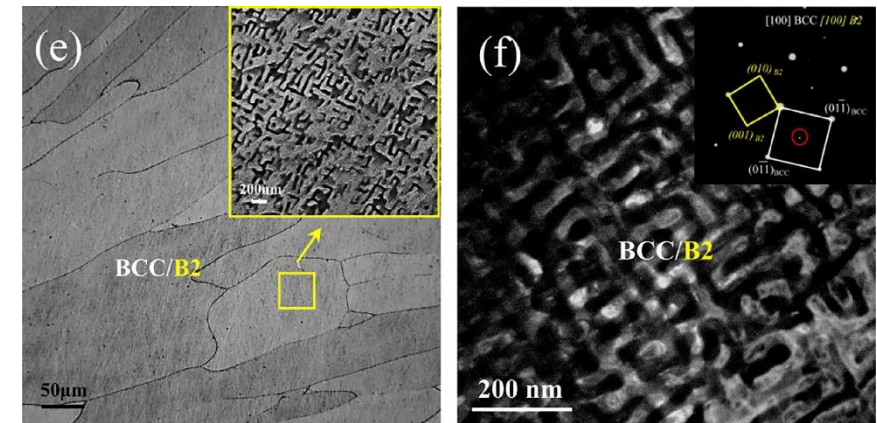
Low misfit  
 $\eta = 0.001$



High misfit  
 $\eta = 0.07$

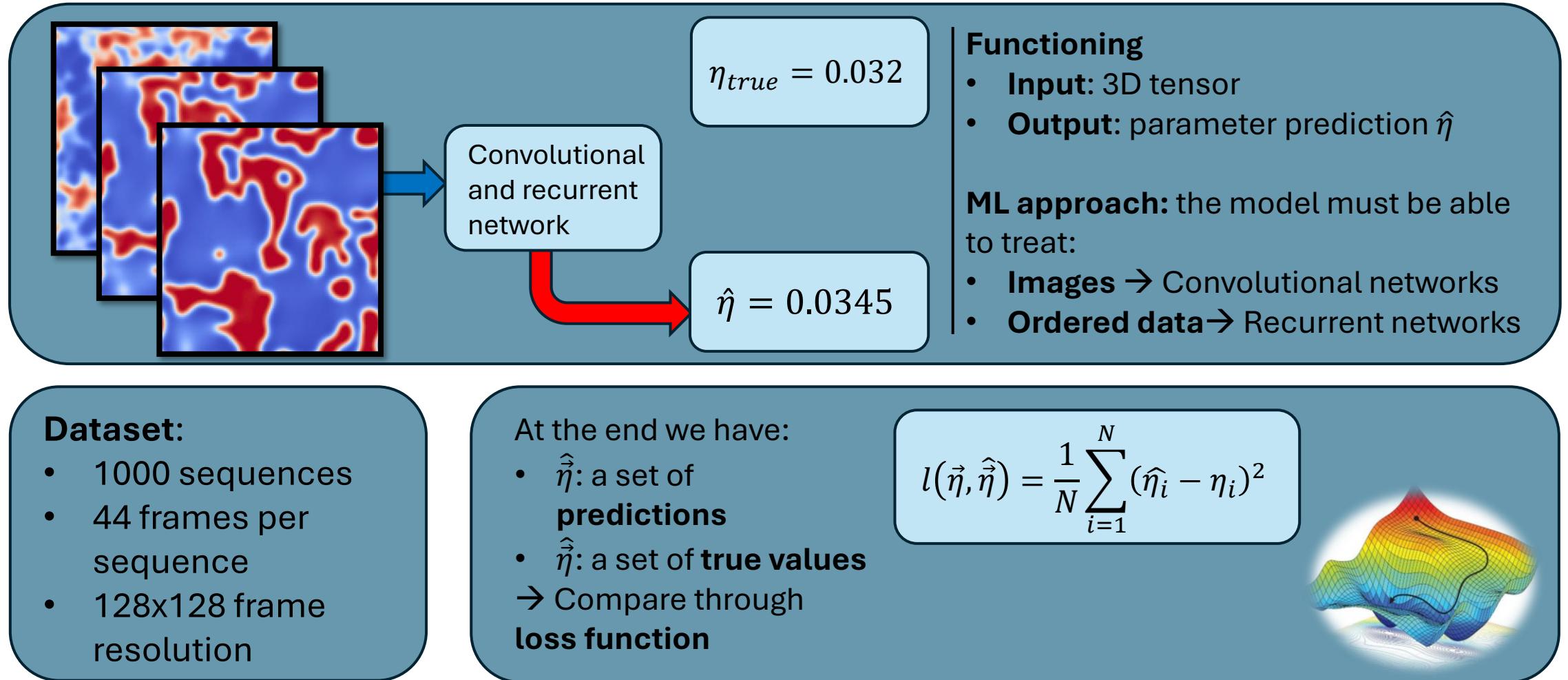


Experimentally



J.L. Li, Z. Li, Q. Wang, C. Dong, P.K. Liaw, *Phase-field simulation of coherent BCC/B2 microstructures in high entropy alloys*, Acta Materialia, Volume 197, 2020, Pages 10-19, ISSN 1359-6454

# Parameter extraction: Evaluation metrics



# Convolutional networks: designed for images

Discrete **convolution operation**:  $R_{ij} = (I * V)_{ij} = \sum_{a,b} I_{ab} V_{i-a,j-b}$

- I: image
- R: resulting representation
- V: kernel → detects **local features**

1	2	1	2
2	1	2	1
1	2	1	2
2	1	2	1

Image

\*

0	1
4	1

Kernel

=

11	7	11
7	11	7
11	7	11

Hidden  
representation

## Visually:

The kernel slides over the image, returning a **linear combination** of the pixels according to its weights.

# Keeping track of the past: Recurrent Networks

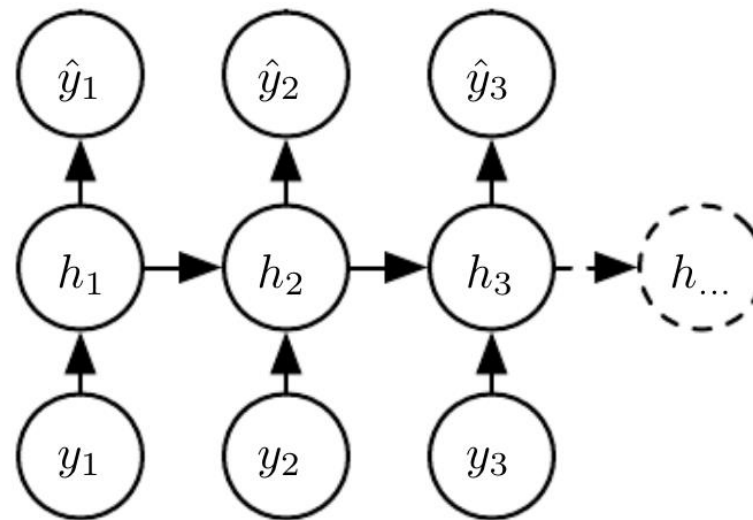
A single image is not enough to determine the parameter!

→ We need to deal with **sequences**

**Difficulties:**

- Arbitrary number of frames
- Influence of the first frames

**Solution:** memory through **hidden units**



Example: **output at every step** with one hidden unit

$y_i$  i-th frame

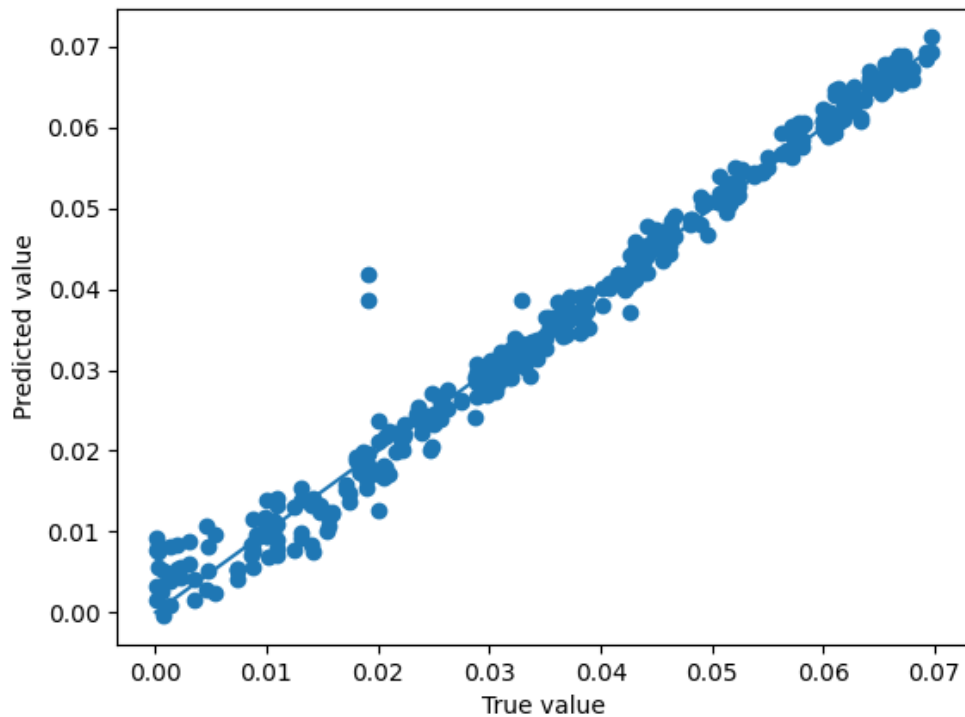
$\hat{y}_i$  i-th output

$h_i$  i-th hidden unit



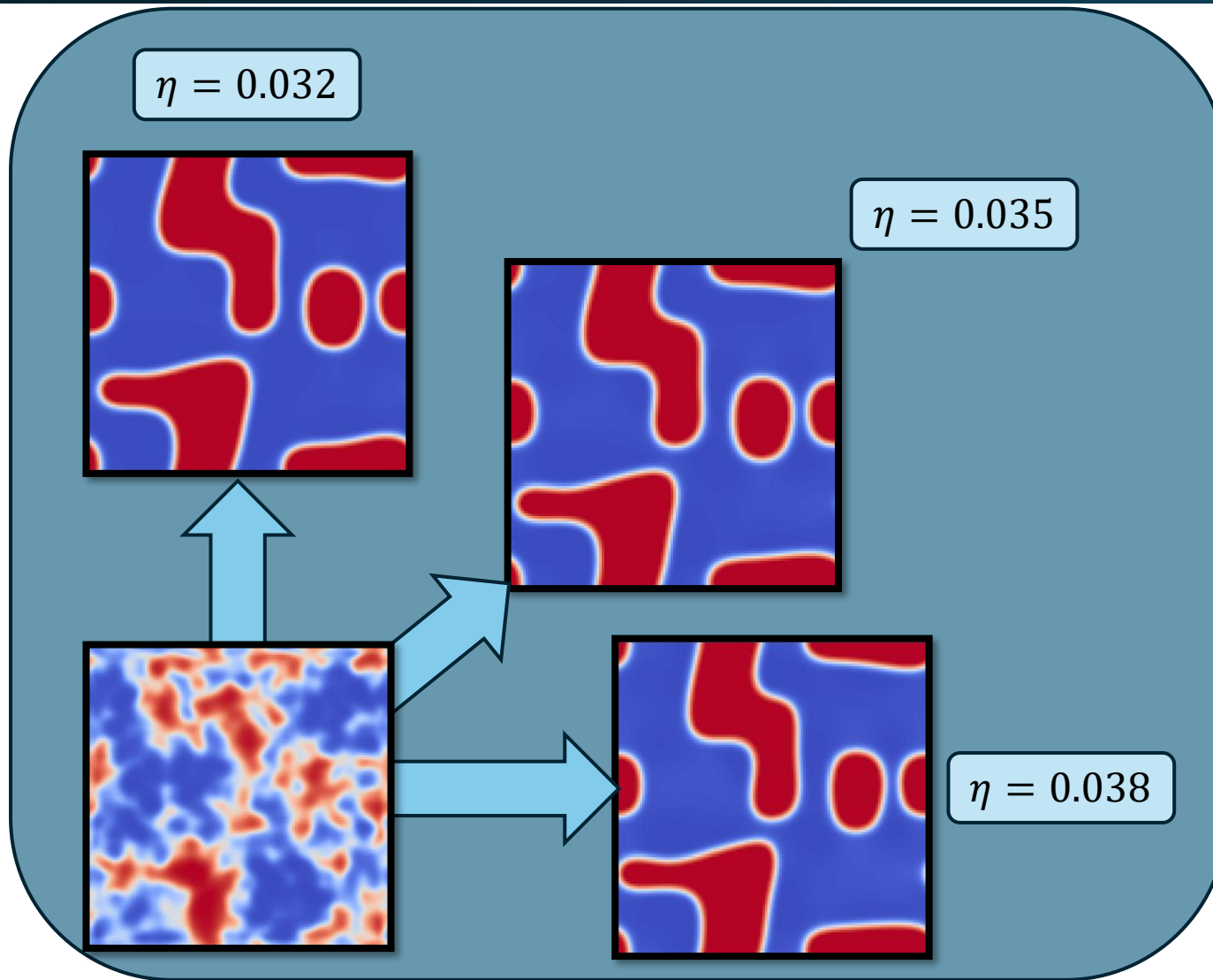
# Parameter extraction: results

## Regression plot



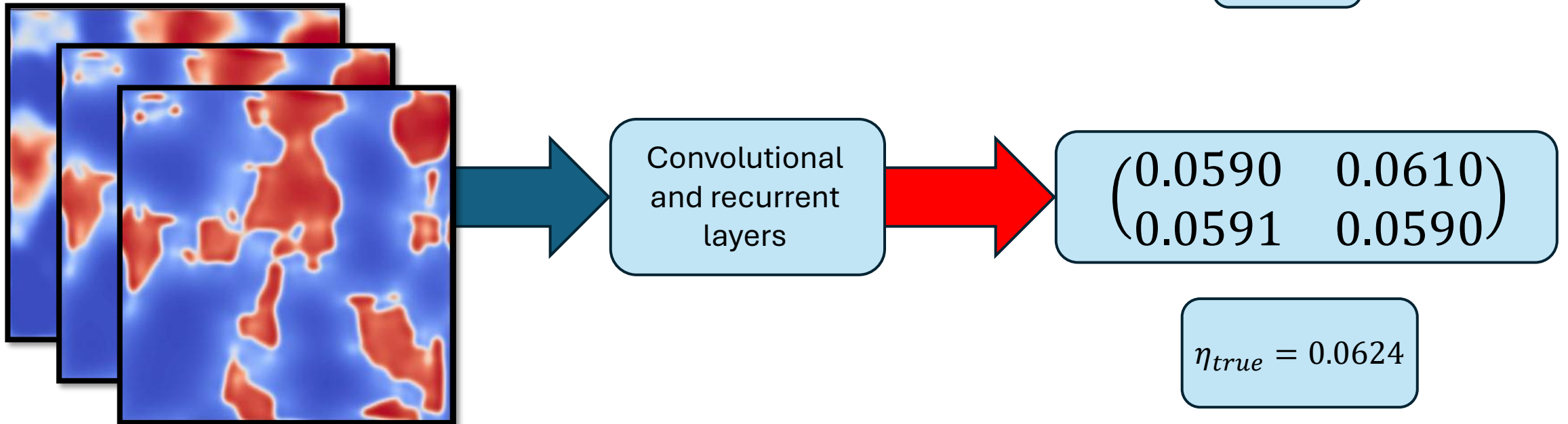
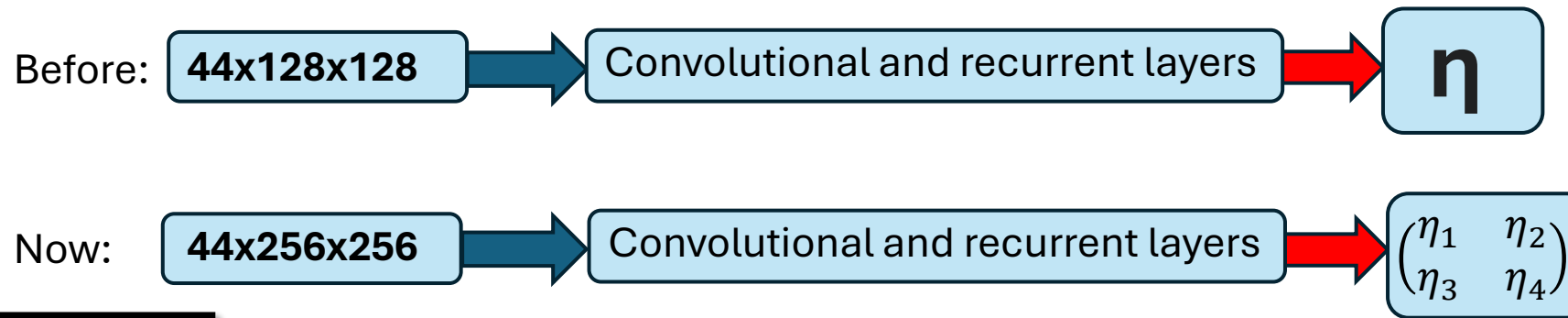
- Comparison between true and predicted value
- Typical absolute error threshold:  $\sigma_{\eta} = 3 \times 10^{-3}$
- Analysis of overestimates, underestimates and outliers

# Differences within the error



- Evolutions from the same initial profile end up in very similar states.
- Low degree of difference within the chosen threshold and time sequence limit.

# Higher resolution sequences



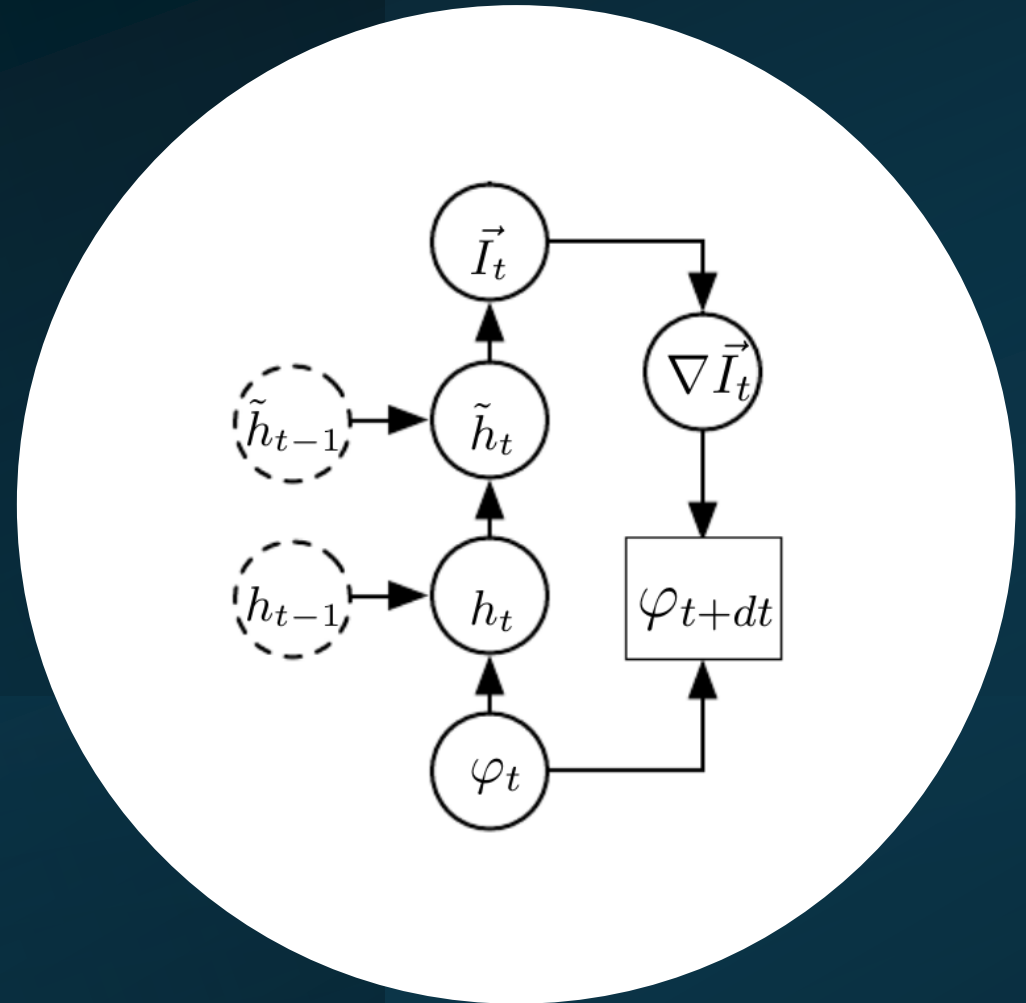
# Evolution simulations

The network output is defined as a current to ease parameter conservation.

This is just a way to encode an inductive bias, which mimics the continuity equation.

Instead of:  $\frac{\partial \varphi}{\partial t} = -\nabla \cdot \vec{I}$

Predict:  $\varphi_{t+dt} = \varphi_t - \nabla \cdot \vec{I}_t$



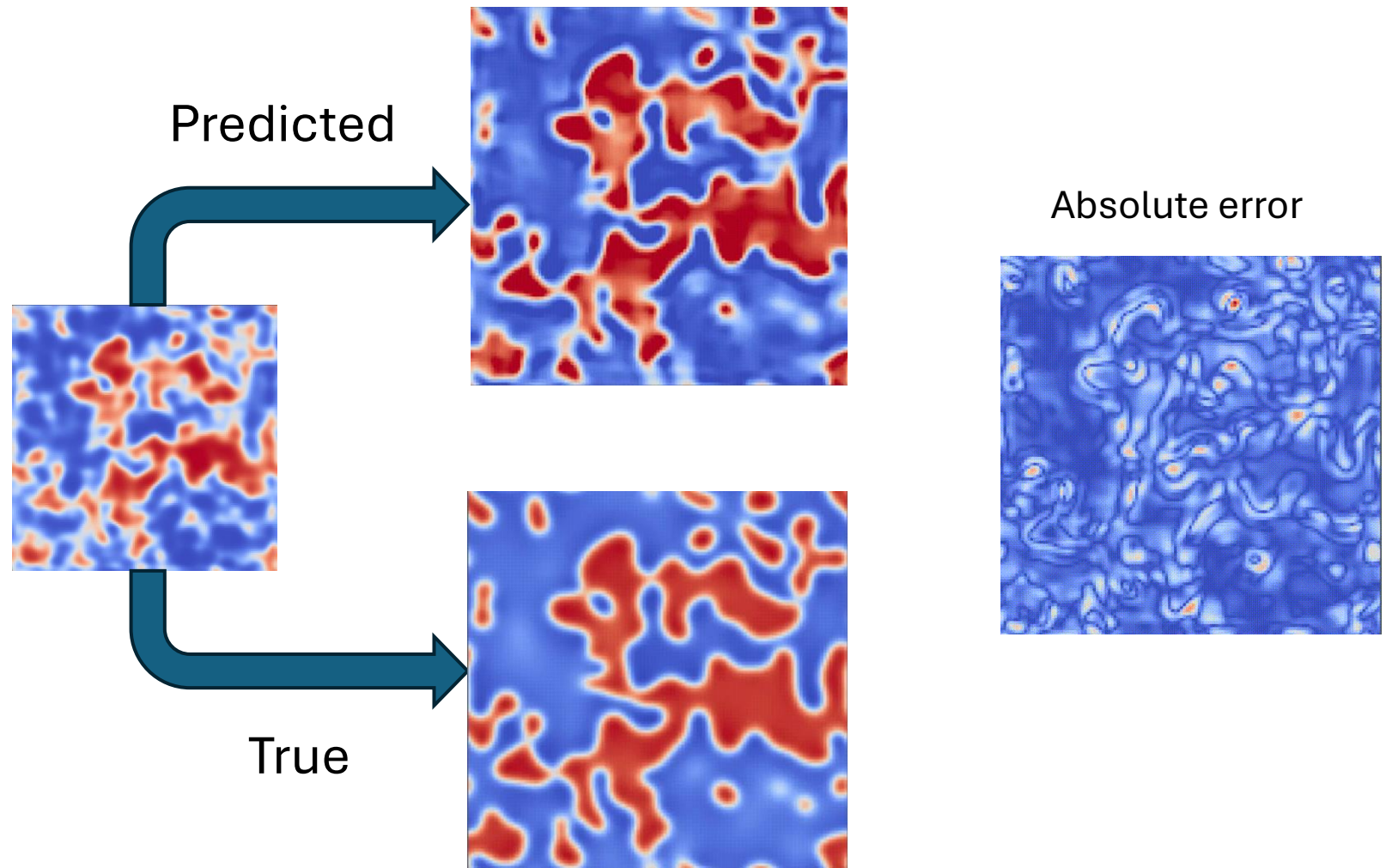


# Prediction capabilities

Process:

- Input: initial state
- Output: evolution for  $n$  frames

Compare predicted and real sequence



# Conclusions and perspectives

## **Parameter extraction:**

Being able to apply the model on experimental sequences would be the ultimate goal.

## **Evolution simulation:**

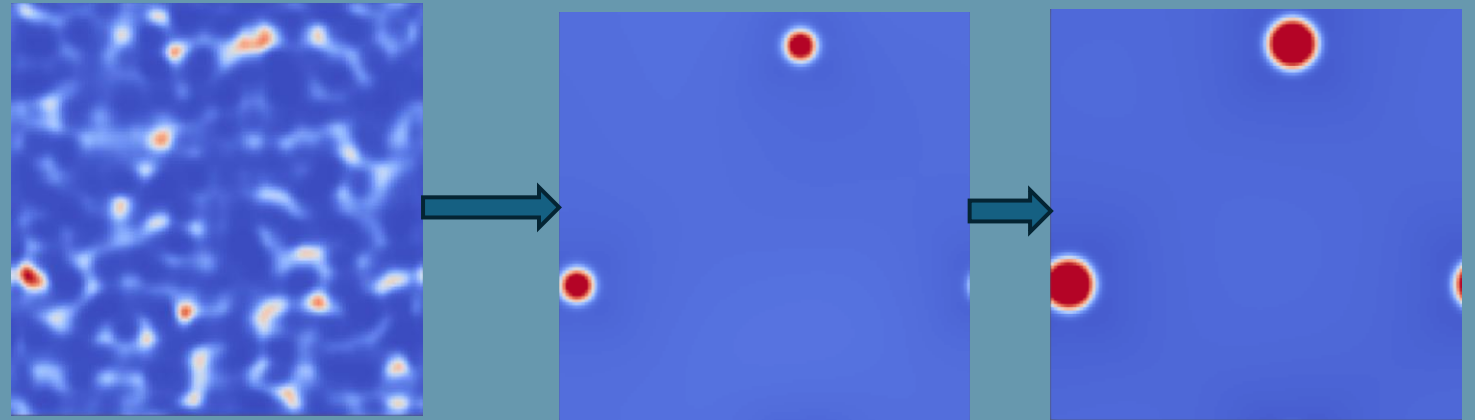
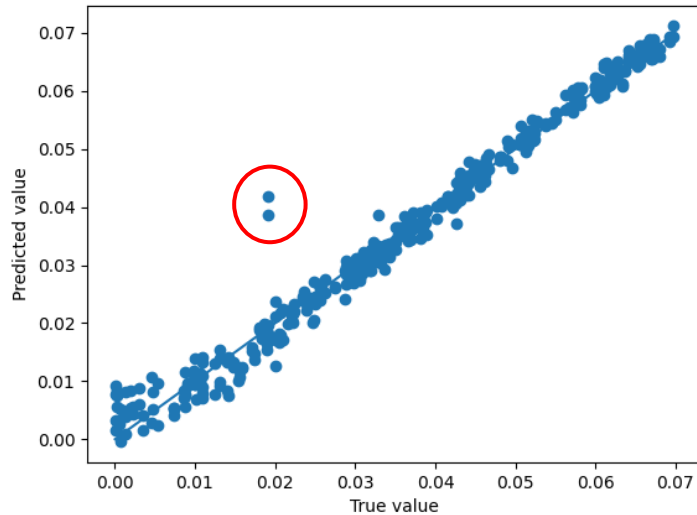
The predictions still present some problems, further research is needed for their resolution.

Thank you for your attention!

# Extra material



# Outliers analysis



This same sequence deceived every model we trained. Including more anomalous examples could lead to more effective NN models.

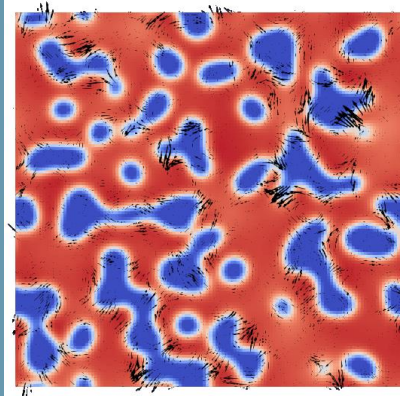
# Evolution simulations

The predicted current could be different from the real one.  
What matters is the divergence: If  $\vec{I} \mapsto \vec{I} + \nabla \times \vec{F}$   
then  $\nabla \cdot \vec{I} \mapsto \nabla \cdot \vec{I}$

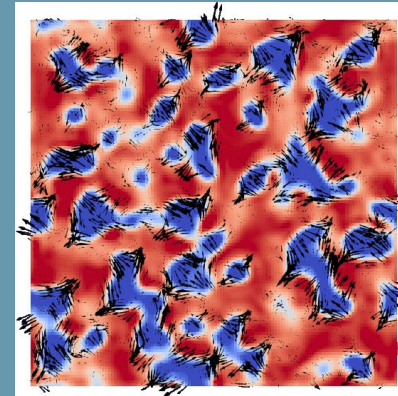
The divergence presents major differences in the first frames.

## Initial profile and current field

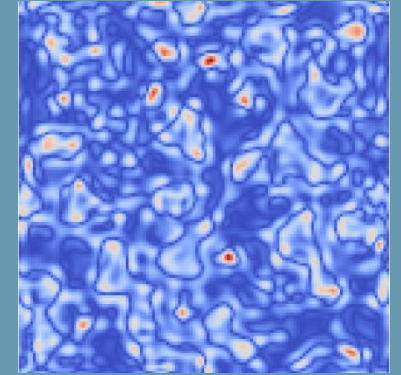
Real



Predicted

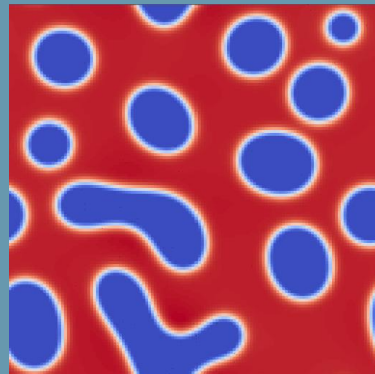


Divergence error

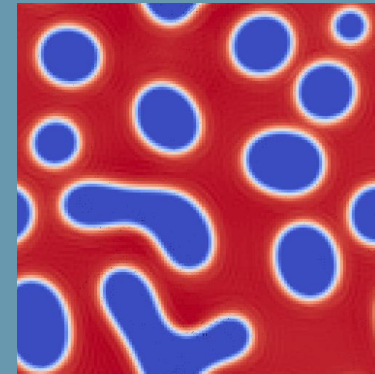


## Final profile and current field

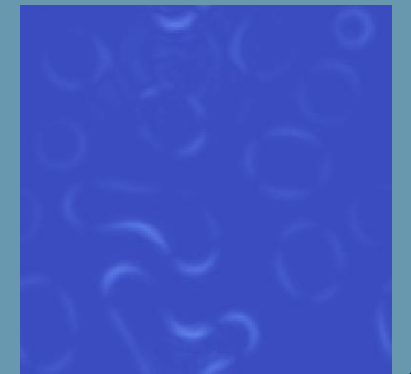
Real

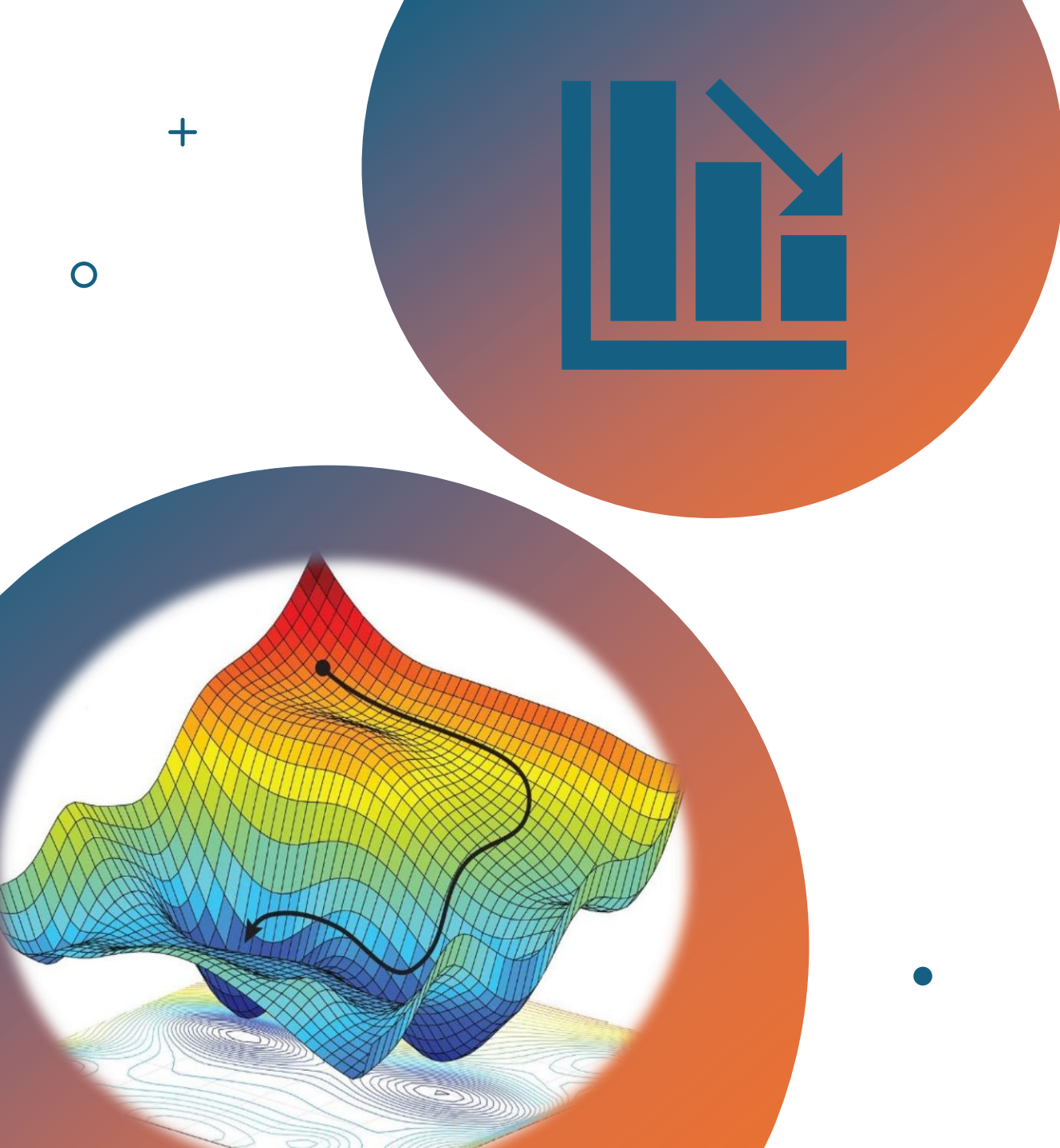


Predicted



Divergence error



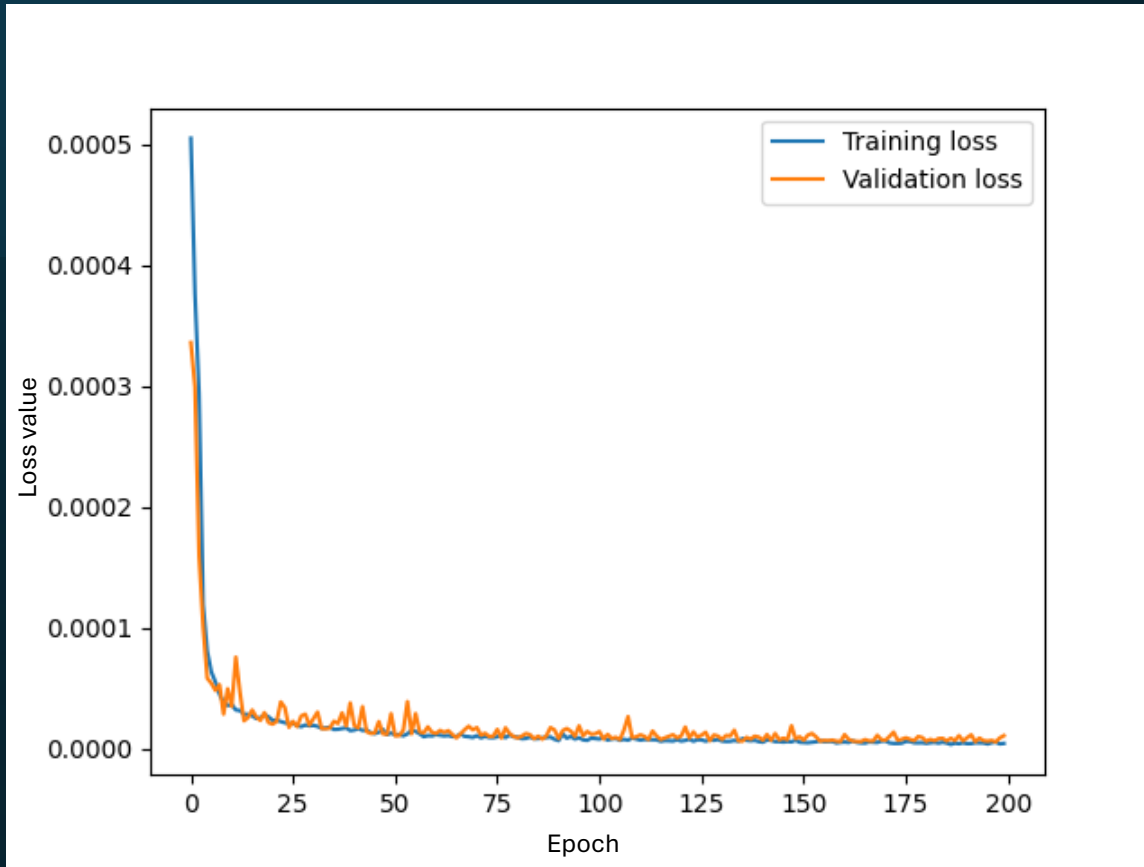


# What does the loss mean?

- Optimization algorithm: **Stochastic Gradient Descent (SGD)**.
- $\nabla_W l(\vec{\eta}, \hat{\vec{\eta}}; W)$  represents the direction of greatest increase of the loss.
- Parameter update:
$$W_{new} = W - \frac{\xi}{|D_t|} \nabla_W \left( \sum_{i=D_t} l(\eta_i, \hat{\eta}_i; W) \right)$$
- **Learning rate  $\xi$**

# Parameter extraction: results

## Loss plot



- Visualization of loss minimization.
- No sign of overfitting, the validation loss stays low



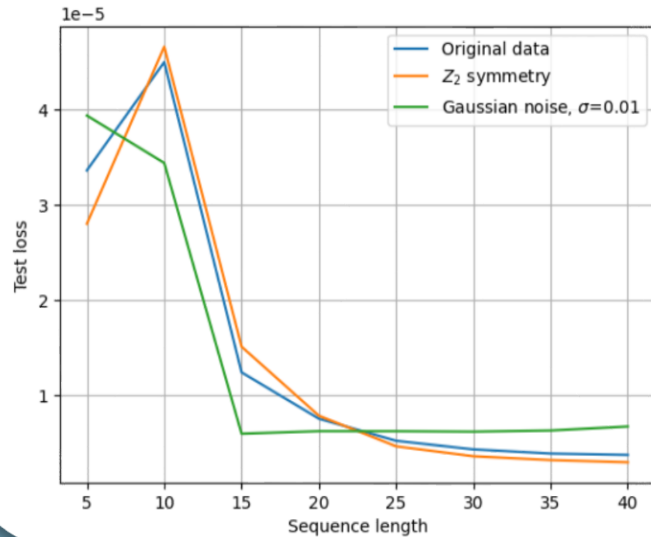
# Extended results

TRAINING TYPE	SELECTED EPOCH	VALIDATION LOSS ( $\times 10^{-7}$ )
Sequences of 22 frames	72	1.786
Sequences of 11 frames	74	3.360
Whole sequences (44 frames)	155	5.364
Casual length	138	0.653
Casual length with noise	123	2.723
Whole sequences with noise	145	0.750
Whole sequences with average pooling	142	2.390

- Learning rate:  $10^{-4}$
- Training epochs: 200
- Batch size: 2
- Training time: from 5 to 20 hours

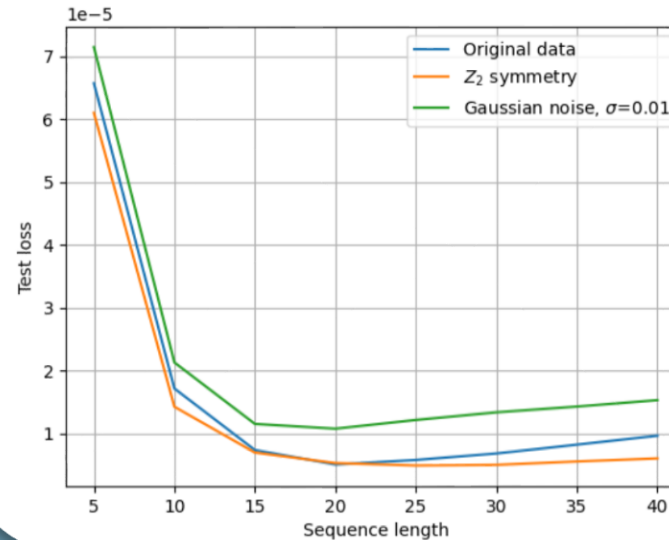
# Results: tests on random sequences

## Training on whole sequences



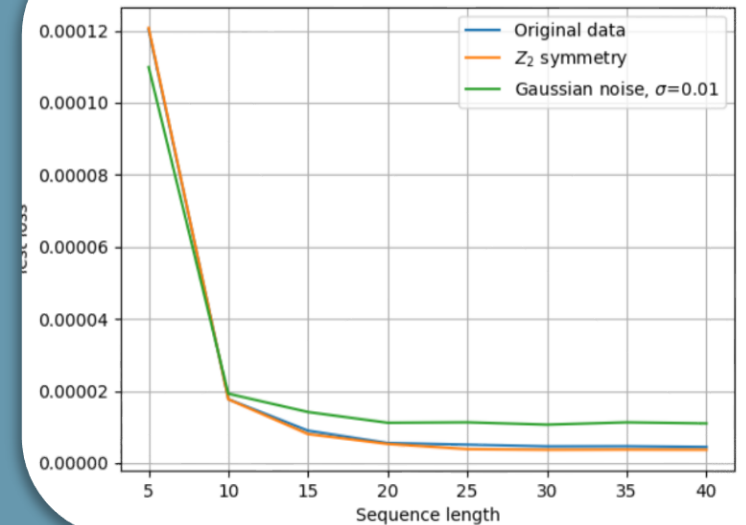
Validation loss:  $5.36 \times 10^{-7}$

## Training on halved sequences



Validation loss:  $1.79 \times 10^{-7}$   
**Slight overfitting:** the model tends to memorize properties of the training data

## Training on random sequences



Validation loss:  $6.53 \times 10^{-8}$   
**Better generalization:** the performance is better on a wider class of data