TRAIL

TRUSTED AI LABS

Temperature Field Prediction in Additive Manufacturing Process using Machine Learning

F. De Geeter, Y. Claes, V. Dachet, V. A. Huynh-Thu, L. Pirenne, O. Rochman, A. Gratia, L. Dierckx, T. V. Andrianandrianina Johanesa, L. Arbaoui, L. Salesses, C. Sainvitu

Additive manufacturing (AM) has many benefits
But it is not perfect

Can create pieces with very complex shapes

Can create very lightweight pieces

Faster than other manufacturing methods

Quality of pieces depends on the process

Sensible to the temperature field

Can simulate with FE, but slow and expensive

Machine Learning (ML) models can be used as fast simulators

ML models can simulate physical processes

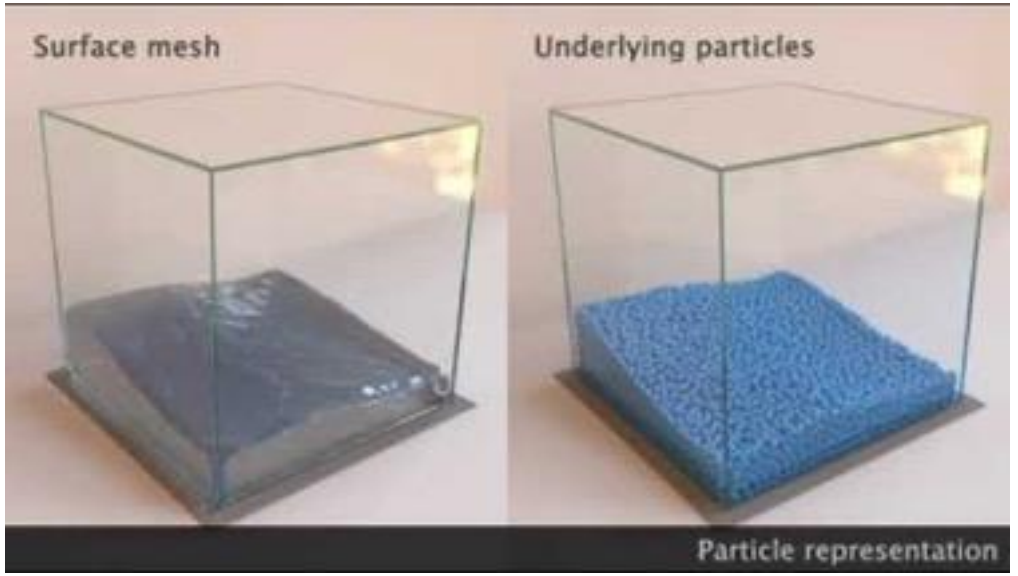
- Faster than classic simulators
- But quality depends on the training data

Can therefore train a model to either assist or replace the simulator

➜ From the past states and the laser parameters, predict next state

Need data

➜ Can create it with a FE simulator



A. Sanchez-Gonzalez et al., Learning to Simulate Complex Physics with Graph Networks, Proc. 37th Int. Conf. ML, PMLR, 119 (2020)

AM is a spatial-temporal process
Can “decouple” spatiality and temporality

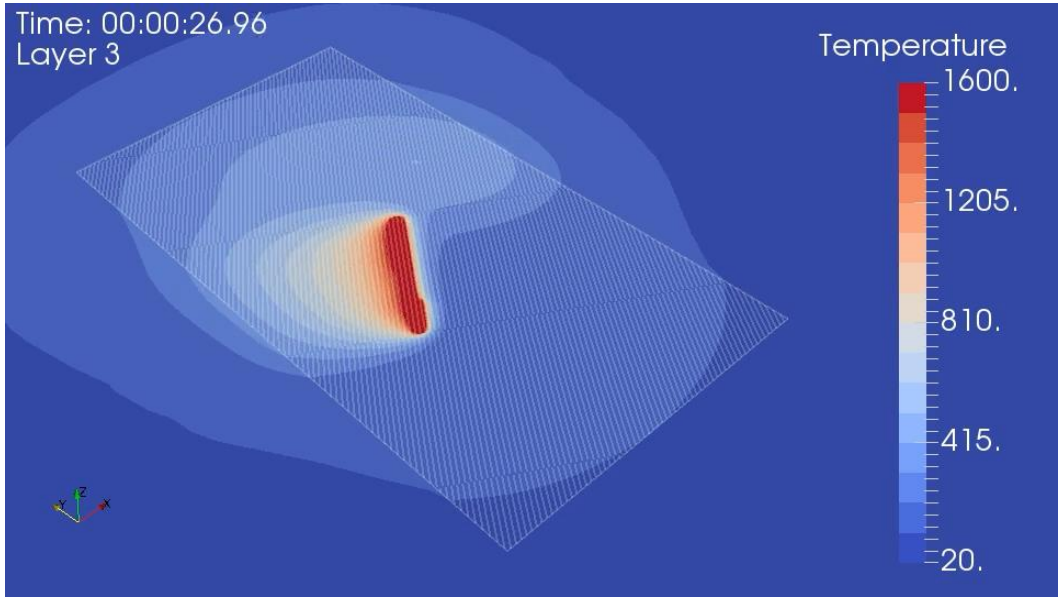
There are good ML models for handling spatial features

The same is also true for temporal features

But handling both can be hard to learn

Idea: Train 2 components

- One handling the spatiality
- The other handling the temporality
- Both are neural networks with their own specificities
- Inspired from the World Models [1]



[1] Ha, David, and Jürgen Schmidhuber. “World Models,” March 28, 2018. https://doi.org/10.5281/zenodo.1207631.

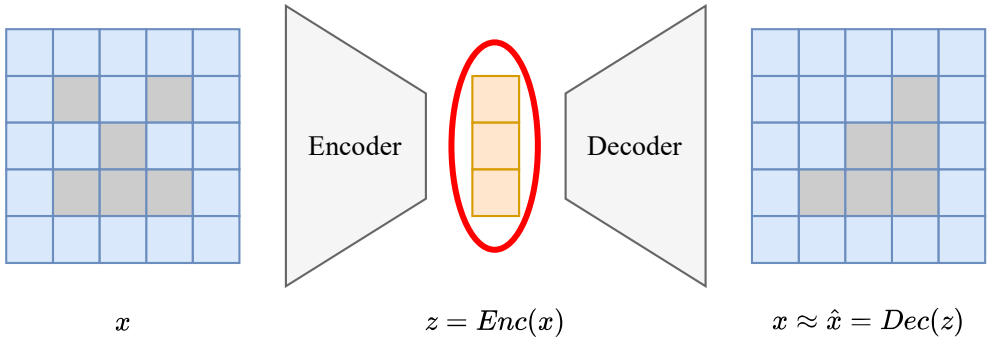
(Variational) Auto-Encoders (VAEs) are good at compressing space

AE are a well-known type of ML model

- Train to reconstruct their input
- Contains an encoder and a decoder
- Must go through a low-dimensional space
  - Also called **Latent space**
  - ➜ Learns to compress data

Make the AE variational to constrain the latent space to be clean

➜ Will ease the training of the other component



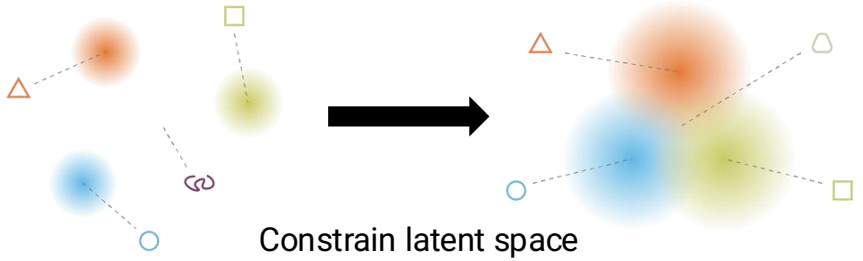


Image: towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

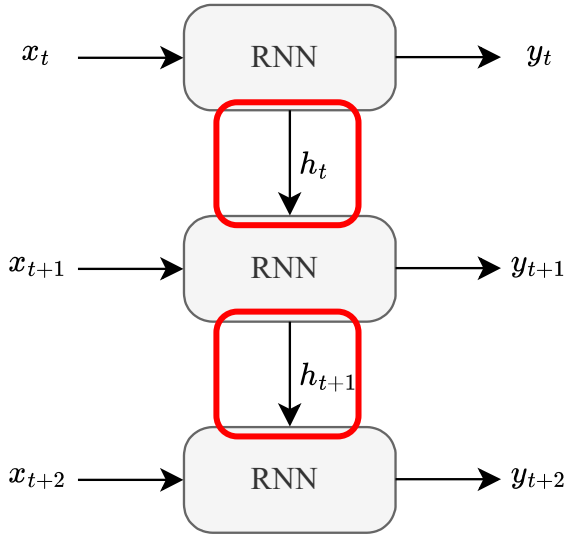
Recurrent Neural Networks (RNNs) are good at modelling time series

RNNs are neural networks with some internal state

➜ Allows them to have memory

Can train a RNN to simulate inside the latent space

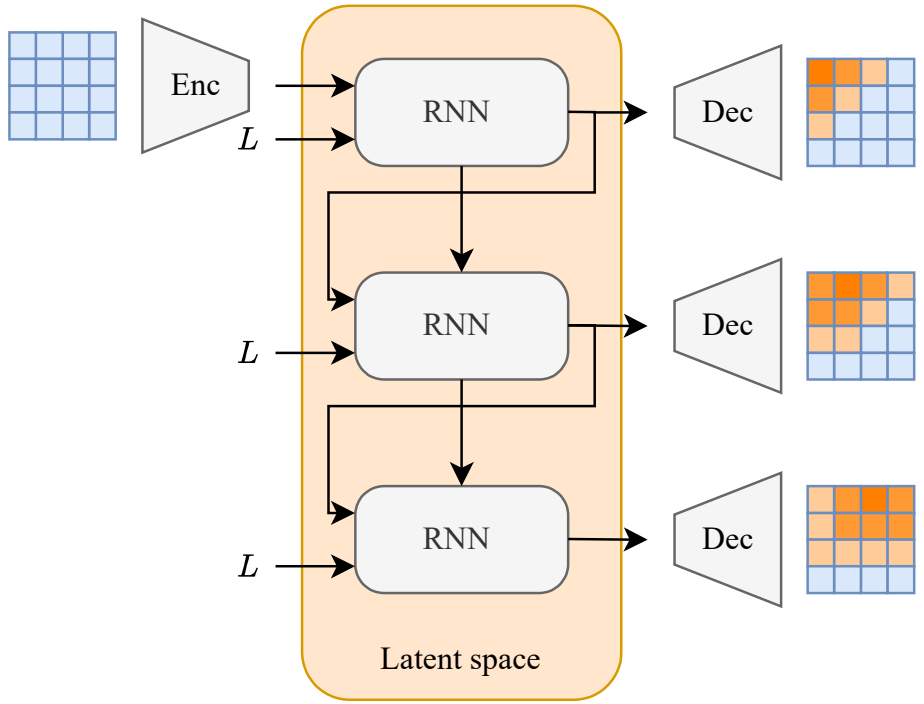
- Take as input:
  - the past encoded state
  - the input features (laser-related)
- Predict the next state



A VAE and a RNN can be combined to simulate the temperature evolution

1. Encode initial field
2. Simulate in latent space
3. Decode encoded predicted fields

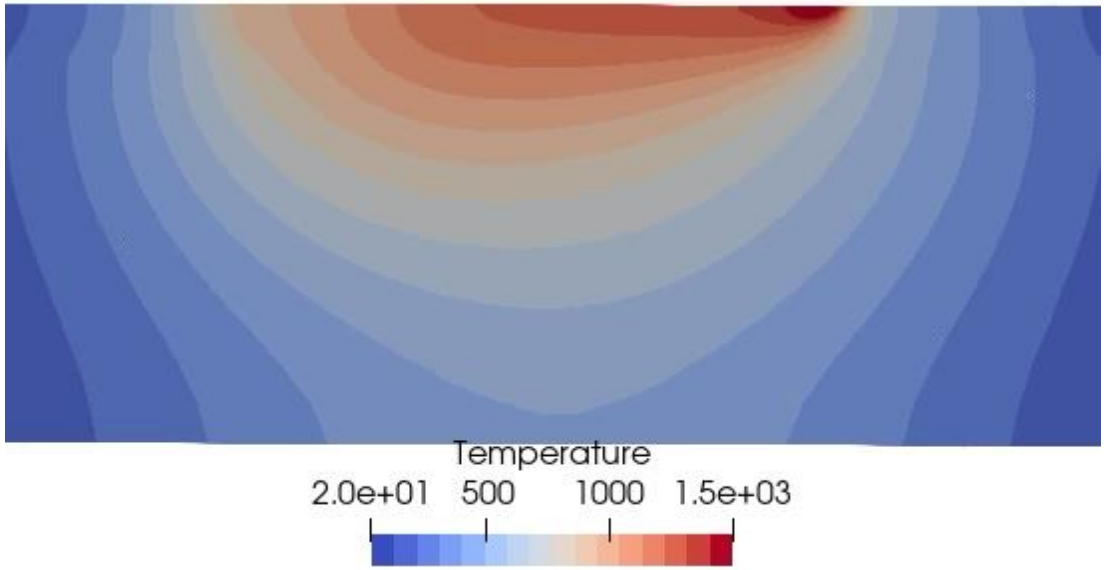
The 2 components can be trained separately



This model was tested on a dataset generated by Ceanero

Some information about the dataset:

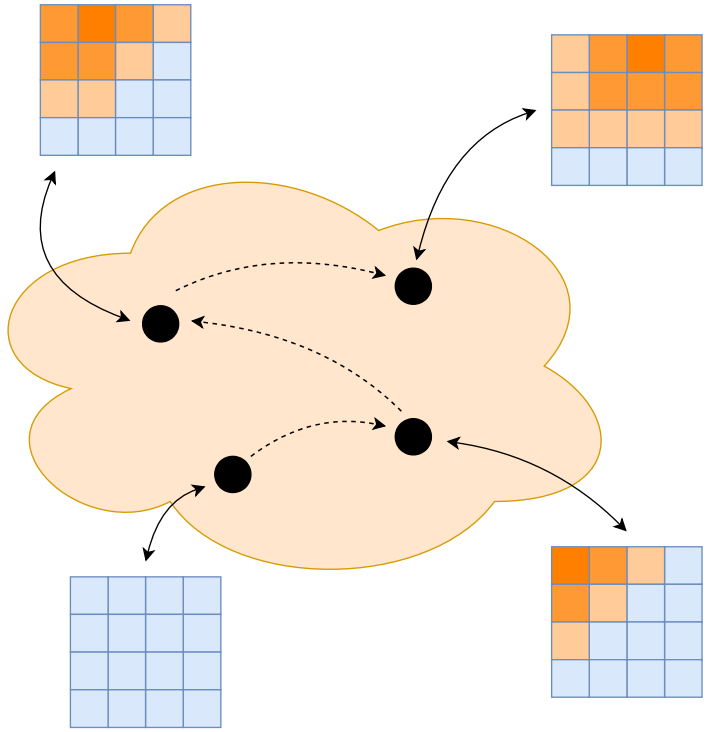
- 256 simulations
  - Generated by a 2D FE model
  - Variable laser nominal power and break time across simulations
- Simple rectangular shape
- Laser moves on the top layer at constant speed
  - New layer after each pass
  - Final mesh of 11 x 131 nodes
- Access to temperature field, laser power and laser position



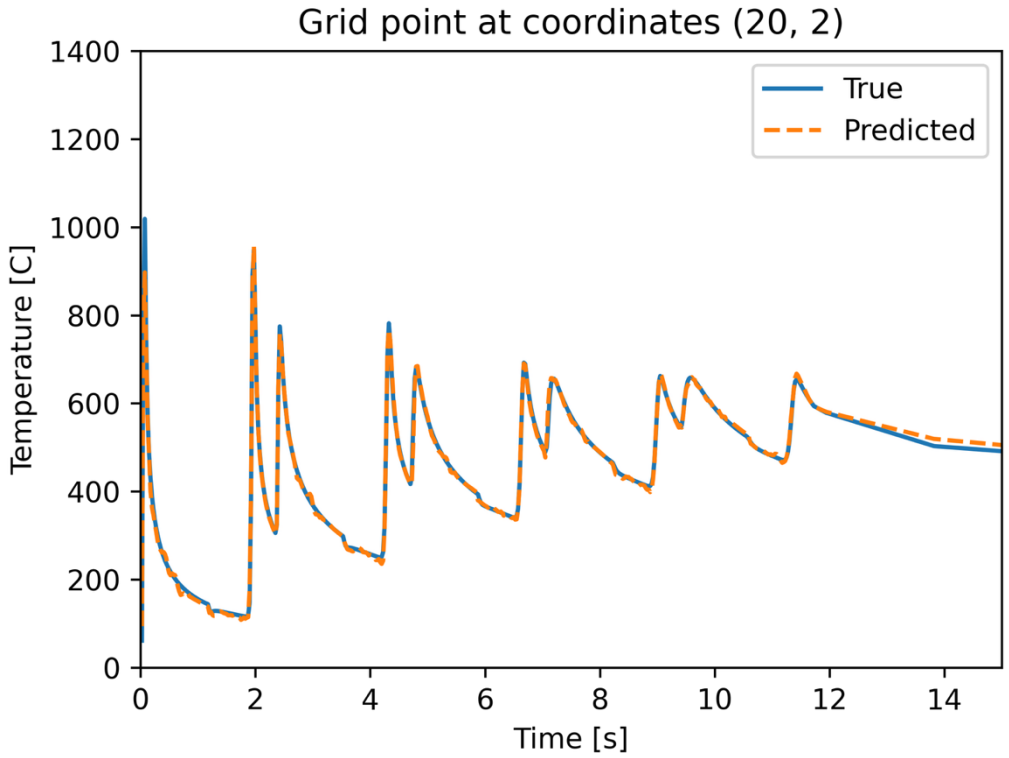
Goal: Predict the **temperature field** at each timestep, given the **initial state** and the **laser position** and **power** at each timestep

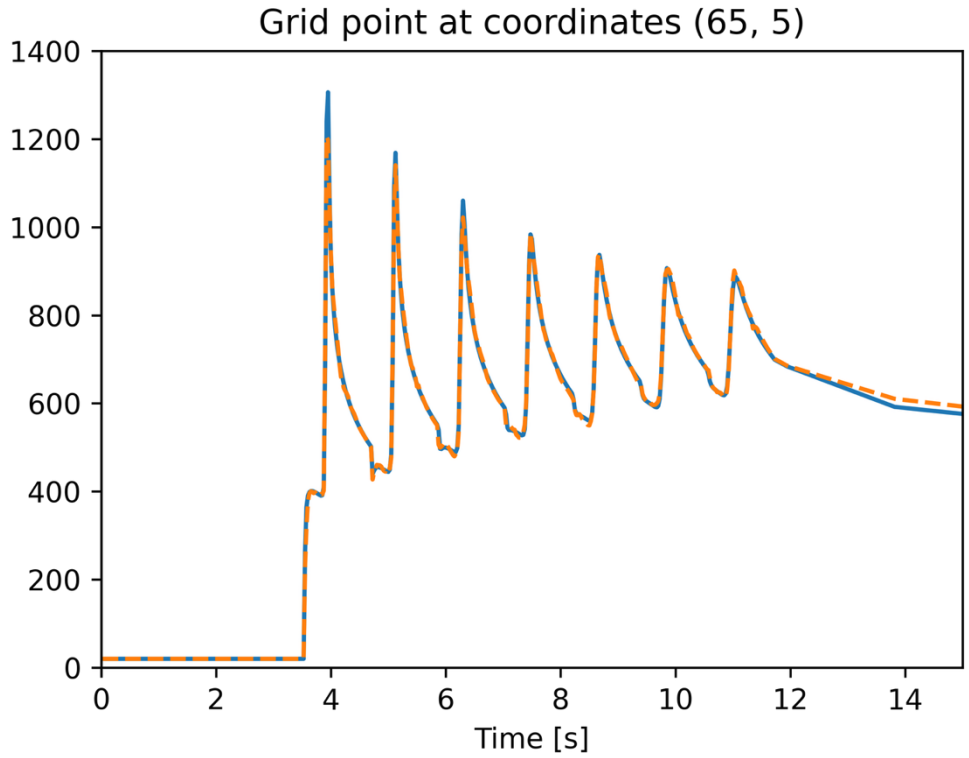
The VAE and the RNN are trained separately

1. Train the VAE to encode and decode the temperature fields
   - Latent space has 64 dimensions
     - vs 11 x 131 = 1441 dimensions in “real space”
2. Train the RNN to simulate inside latent space



The model achieved good performance





Mean error: ± 12°C

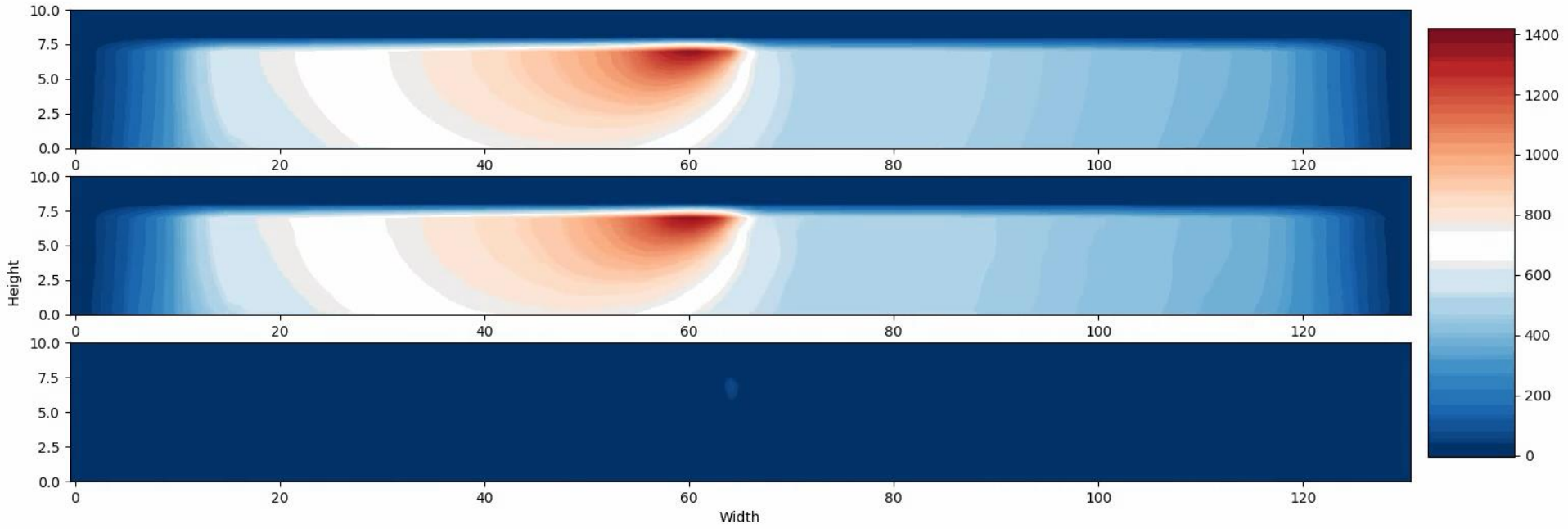
Higher errors at the laser and on the boundaries

Here is an example of simulation

Target simulation

Predicted simulation

Absolute difference



There are three potential directions

Data

- “Simple” dataset here
- Try on more complex dataset
- Especially with more complex shapes

Model

- “Simple” approach here
  - But if it works, use it
- May not work with more complex shapes
- Try other architectures
  - Graph Neural Networks
  - Fourier Neural Operators

Physics

- Purely data-driven here
- Possible to add some physics in the model
  - Either through constraints
  - Or by combining data-driven model with physical model

Here are some take-home messages

We presented a machine learning model that:

- Can simulate the temperature field evolution of a piece being printed
- Is simple yet efficient
- Has two components:
  - One spatial component that compresses the space
  - One temporal component that models the process inside this reduced space
- Achieve good performance on a high-fidelity dataset of 2D simulations

Thank you !

This work is supported by the "ARIAC by DigitalWallonia4.ai" research project (grant agreement No 2010235 - TRAIL Institute) and benefited from computational resources made available on Lucia, the Tier-1 supercomputer of the Walloon Region, infrastructure funded by the Walloon Region (grant agreement No 1910247).

Variational Auto Encoder - Architecture

Encoder

- 3 convolutional layers with 16, 32 and 64 channels.
  - Each using batch normalization and a leaky ReLU.
  - Parameters: padding of 1, stride of 2, dilation of 1 and kernel size of 3.
- 2 fully-connected layers with 128 units.
  - Each using a leaky ReLU.
- 2 final fully-connected layers of 64 units to output the means and the log variances.

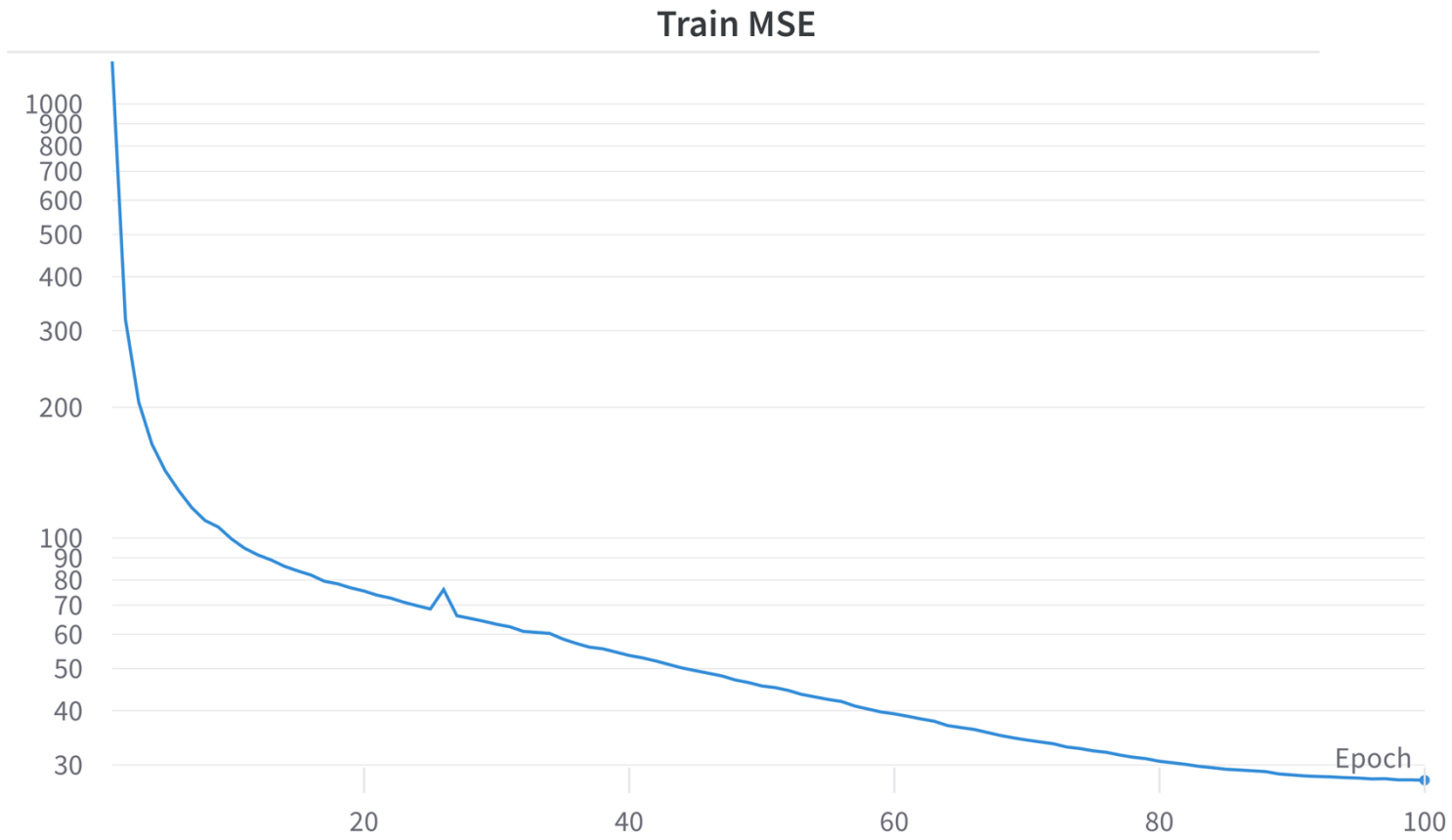
Decoder

- 2 fully-connected layers with 128 and 2176 units.
  - Each using a leaky ReLU.
- 3 transposed convolutional layers with 64, 32 and 16 channels.
  - Each using batch normalization and a leaky ReLU.
- 1 final convolutional layer to reconstruct the grid.

Variational Auto Encoder - Training Curves

Training parameters:

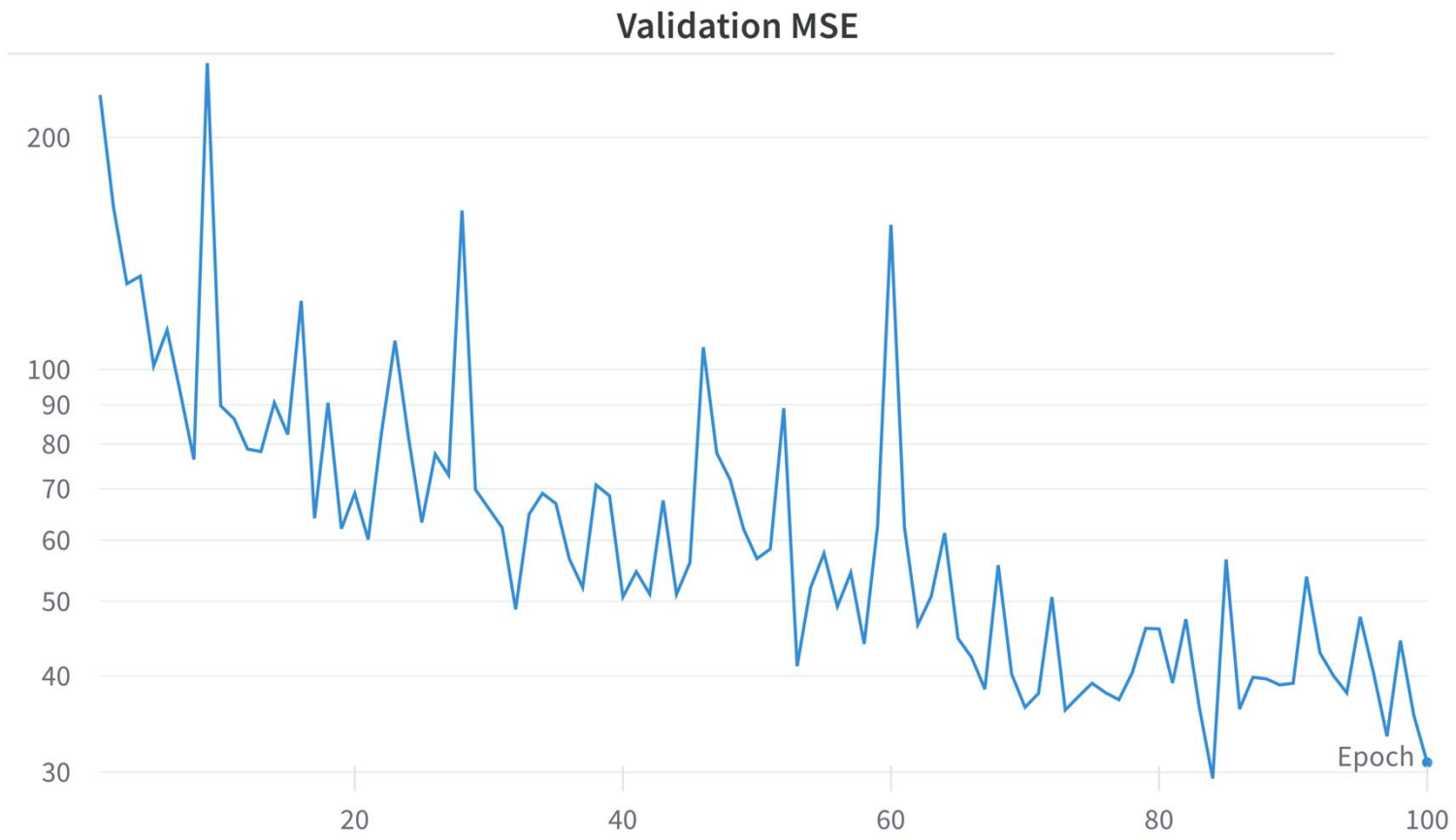
- Epochs: 100
- Batch size: 32
- Learning rate: 0.001 (scheduling)
- Sequence stride: 1
- KL weight: 0.1



Variational Auto Encoder - Training Curves

Training parameters:

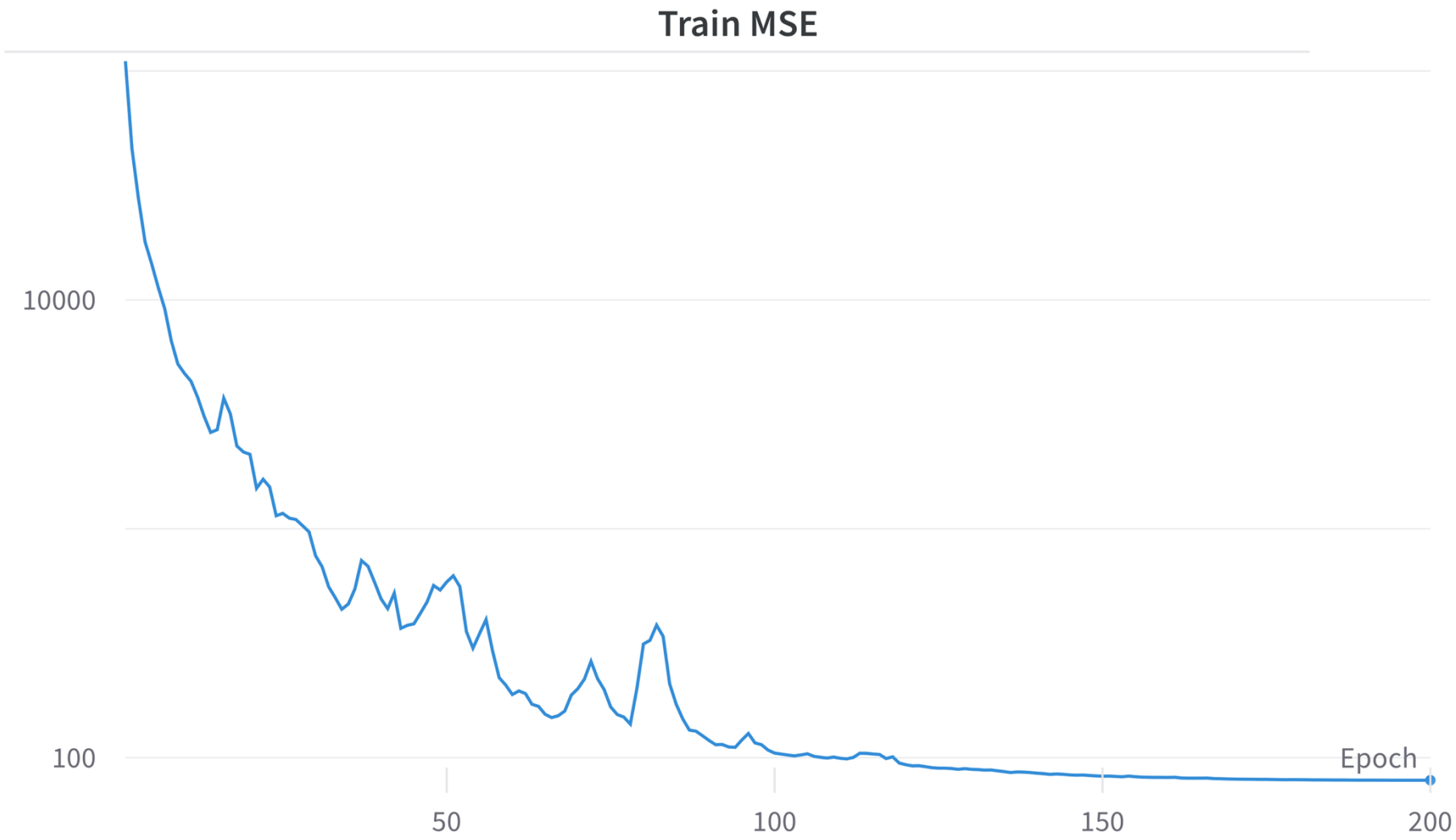
- Epochs: 100
- Batch size: 32
- Learning rate: 0.001 (scheduling)
- Sequence stride: 1
- KL weight: 0.1



Latent Simulator – Training Details

1. Architecture:
   - 1 GRU recurrent layer with a hidden state of size 1024
   - 2 FC layers with leaky ReLUs: (1024 -> 512) & (512, 64)
2. Training hyperparameters:
   - Batch size: 16
   - LR: 0.001 (LR scheduling)
   - Sequence stride: 5
3. Training procedure:
   - Unnormalized laser inputs are fed into the GRU with the current encoded grid (latent vector)
   - Hidden state fed to FC layers, which yield the next latent vector
   - Next latent is decoded into next grid, on which the MSE is computed

Latent Simulator – Training Curves



Latent Simulator – Training Curves