A virtual spectral diagnostic for plasma accelerated bunches at FLASHForward

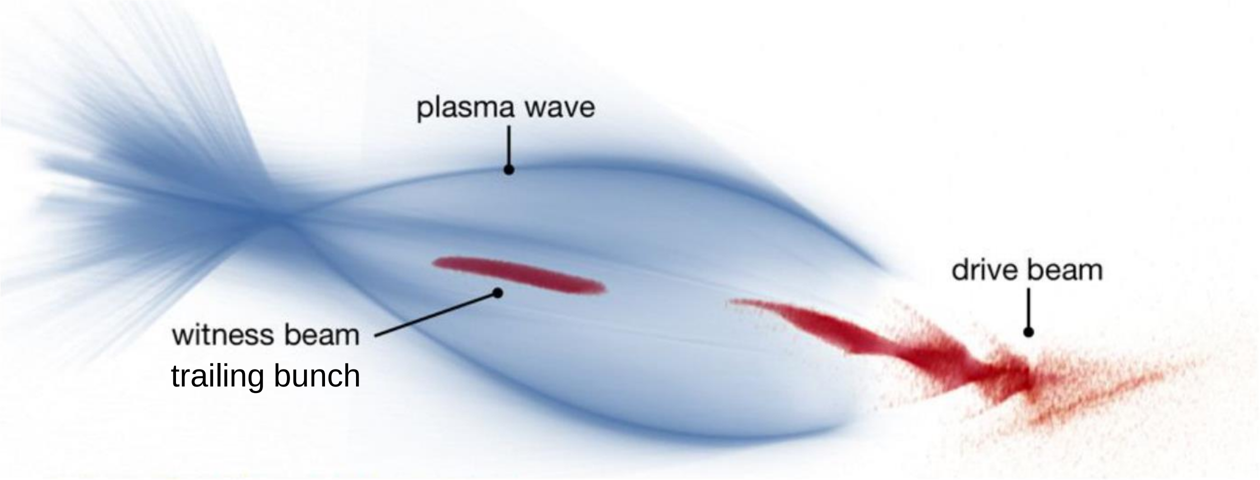
AKBP 7: Novel Accelerator Concepts II and FELs, DPG Göttingen 2025

Philipp Burghart, Lewis Boulton and Jonathan Wood for the FLASHForward collaboration

HELMHOLTZ

Plasma Wakefield Acceleration

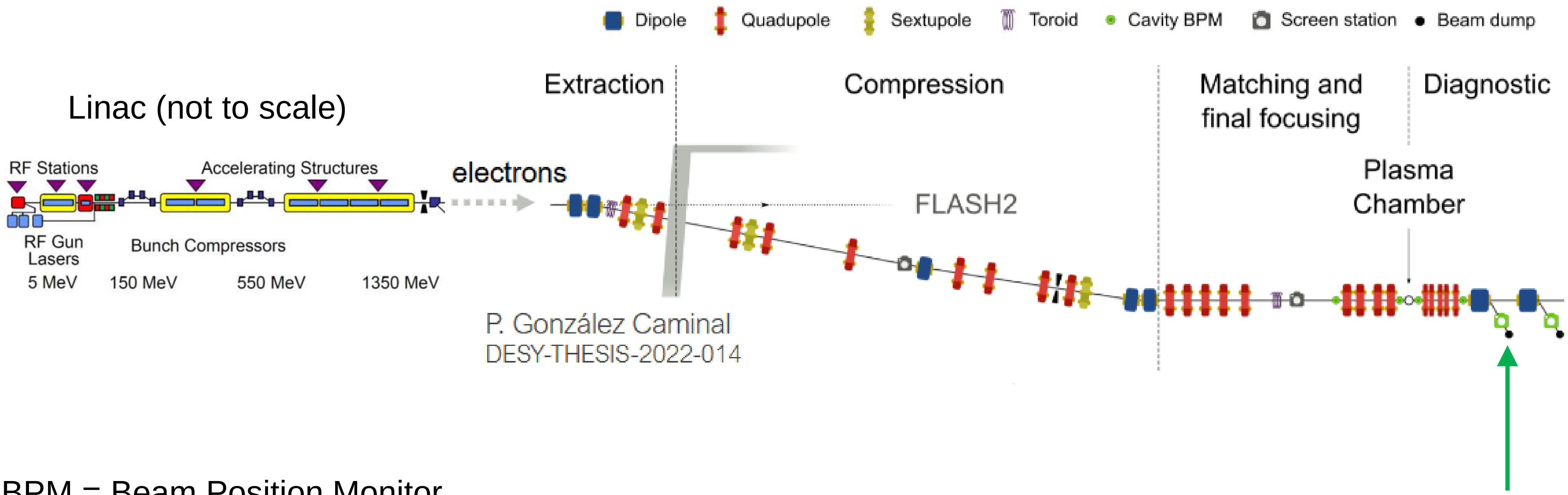
- Conventional accelerators use superconducting RF cavities
  - Limited by breakdown voltage to fields of order 100 MV/m
- Can be overcome using plasma
- Multi-GV/m acceleration gradients achievable
- Promises to lower the size and cost of future machines
- However, PWFA is affected by the entire 6D phase space of the driver
  - Initial phase space affected by many inputs

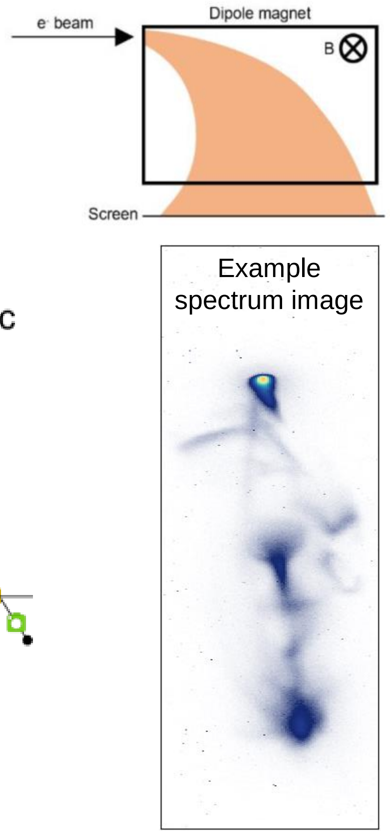


Courtesy of P. González Caminal

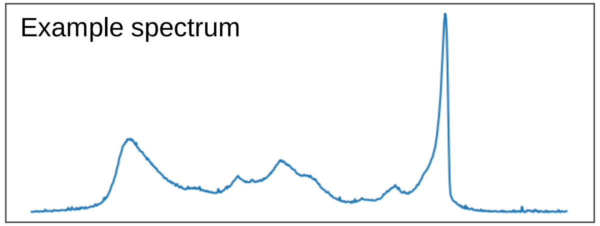
FLASHForward

PWFA beamline





Electron spectrometer



BPM = Beam Position Monitor

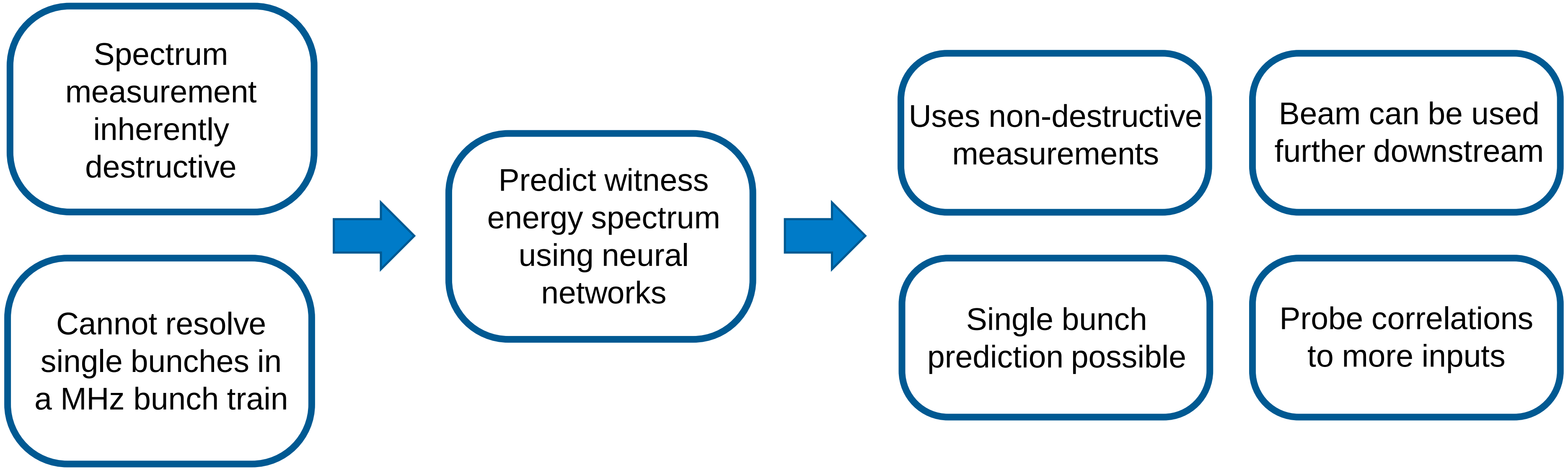
BCM = Beam Compression Monitor

TOR = Charge measurement with toroidal magnet

RF = Radio frequency acceleration module

e.g. Energy depletion and re-acceleration of driver electrons in PWFA (F. Peña et. al., *Phys. Rev. Research* **6**, 043090, 2024)

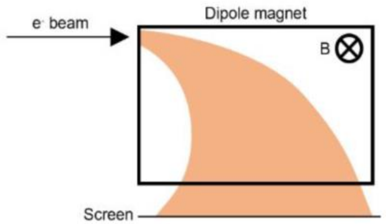
Motivation

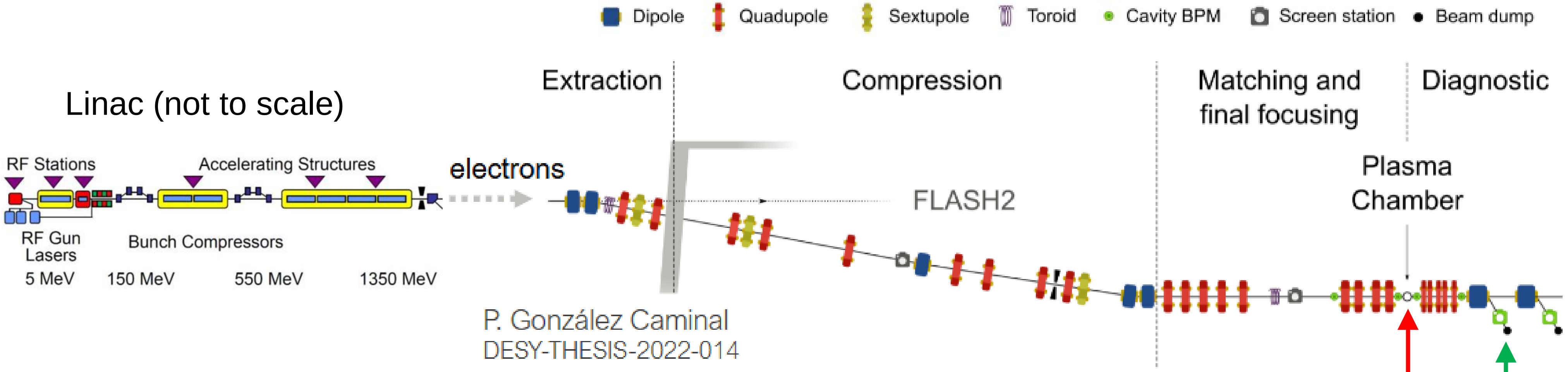


- Examples in literature:
  - Decoding sources of variability in an LPA (A. Maier et. Al., *Phys. Rev. X* 10, 031039, 2020)
  - Prediction of FEL laser power from machine parameters (T. Korten et. al., arXiv:2411.09468v2)
  - Prediction of longitudinal phase space images (C. Emma et. al., *Phys. Review Accel. Beams 21* (11), 112802, 2018)

FLASHForward

PWFA beamline

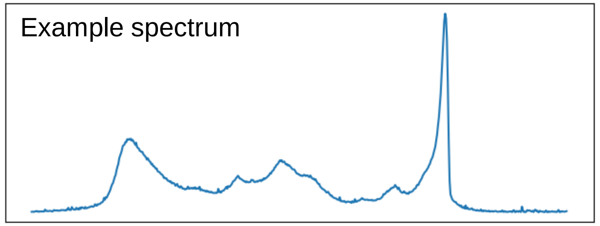




BCM & RF params from the Linac

BPM & Toroids from the FLF beamline

Electron spectrometer



Input values used for prediction

Electric discharge: Temporal position of peak current

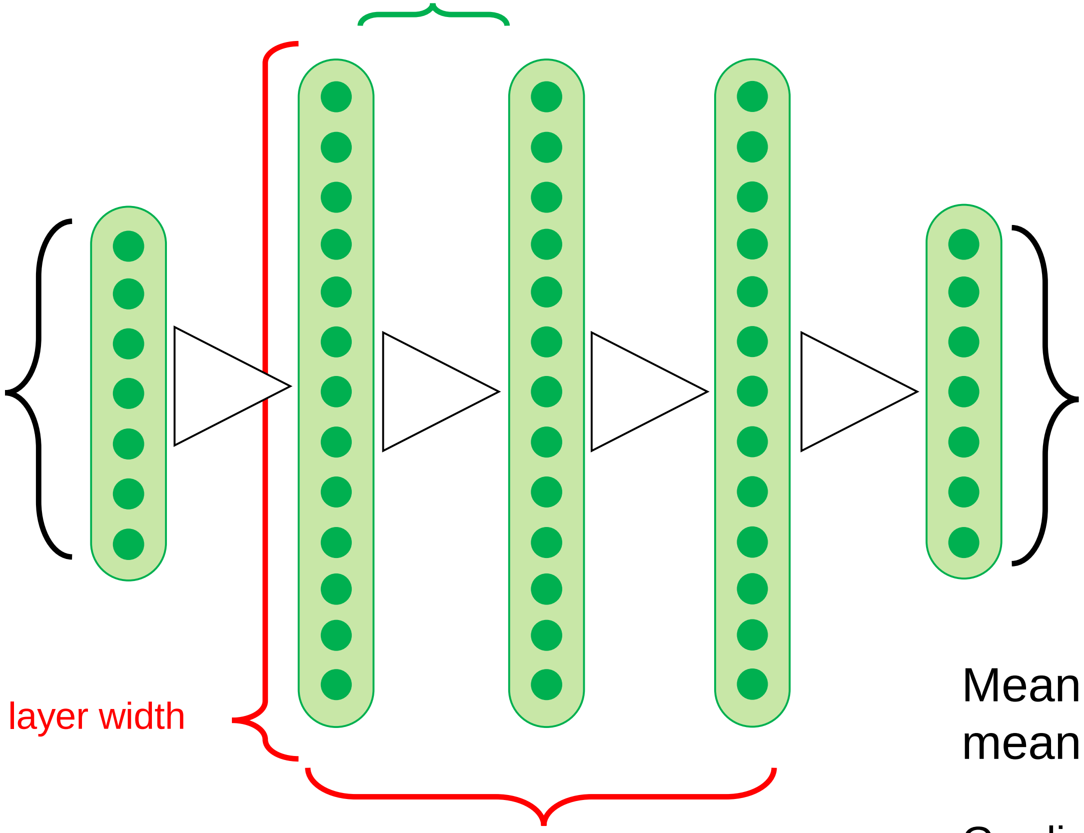
Neural Networks

General information

Parameters

Each layer performs a linear operation with weights A and biases b: $\vec{y} = A\vec{x} + \vec{b}$, then applies an activation function

Inputs (e.g. BPM, BCM, TOR values)



Outputs (the pixel values of a spectrum)

Hidden layer width

Number of hidden layers

Hyperparameters

Mean squared error (MSE) Loss = mean((predicted-real)^2)

Gradient of Loss → weight adjustment

Magnitude of adjustment = *learning rate*

How to work with a NN?

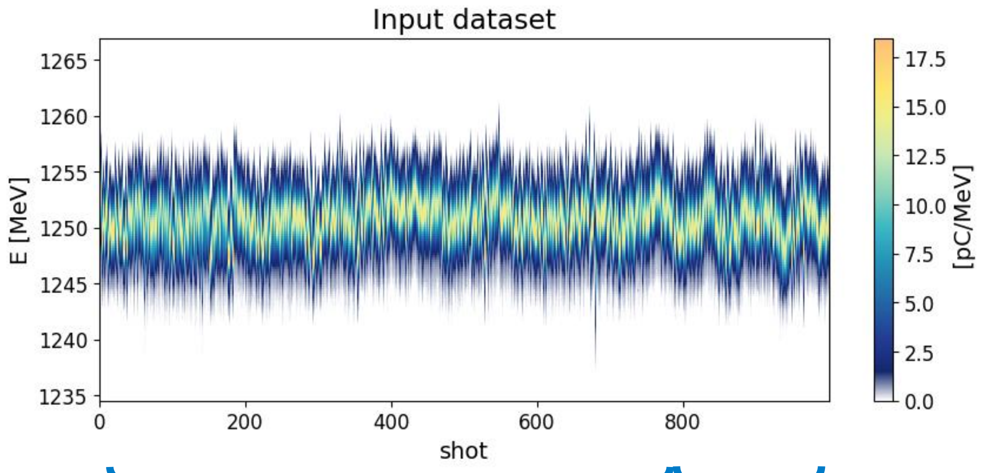
Split data into train and test sets

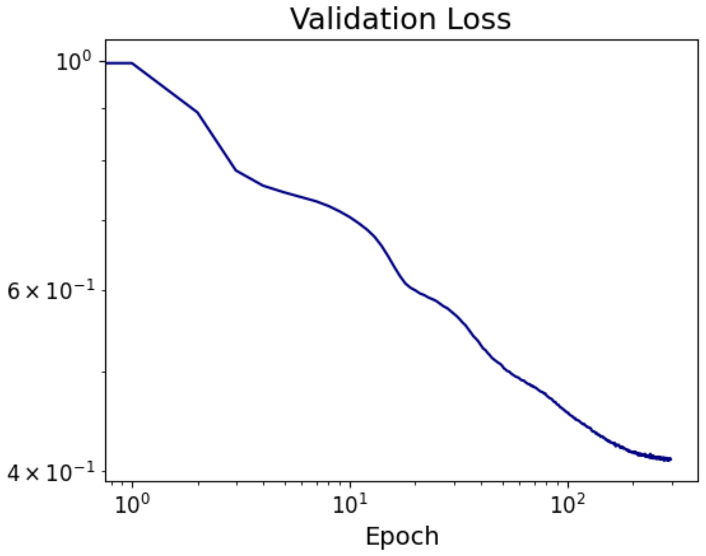
Perform training on train dataset

While training, track loss with validation set

Finish training when *validation* loss stops decreasing (no overfitting!)

Evaluate model performance on test dataset





This is only for visualization: The selection is randomized

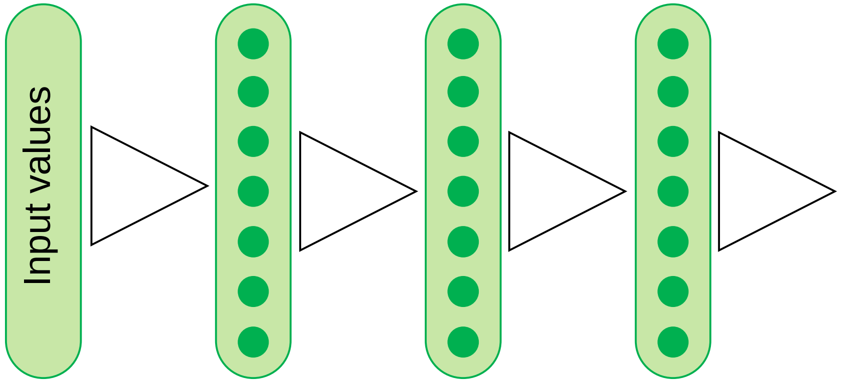
Test dataset

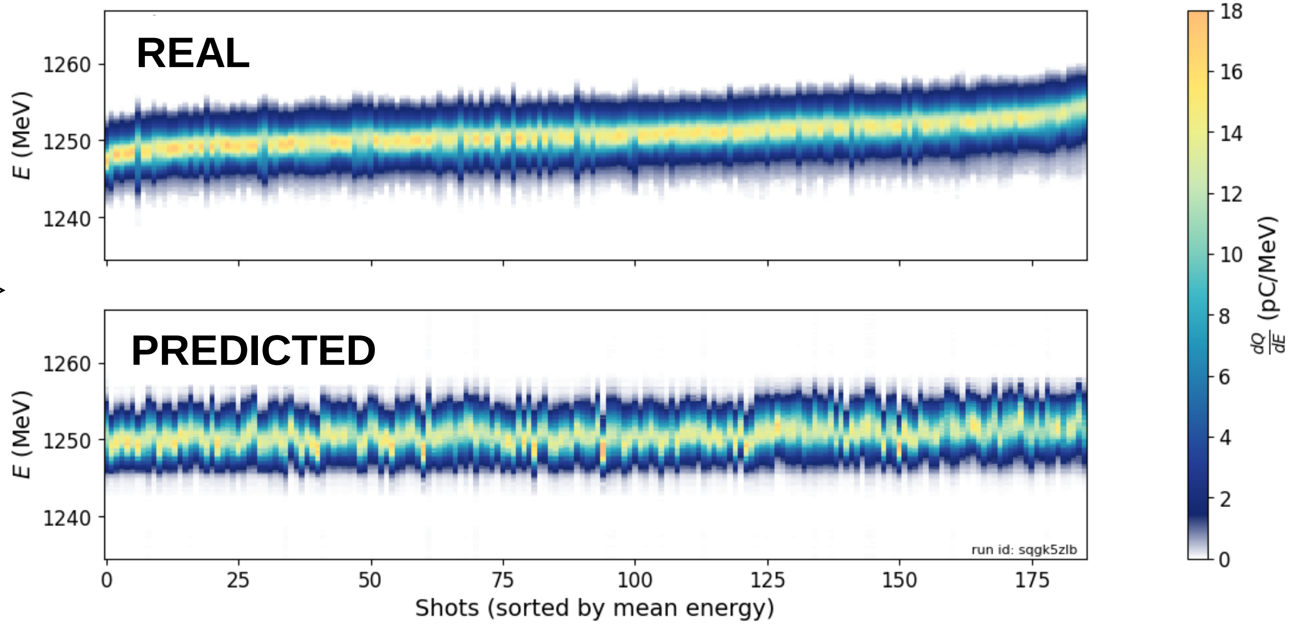
Training dataset

Validation

Densely connected neural network

- Input dataset: 1000 shot ‘statistics’ run with no active changes to beamline settings
- Dataset split into train-test-validation (80%-20% twice)
- Network structure: 3 densely connected linear layers (width 100) with ReLU activation, MSE loss
- Output: 100 “pixel”-wide spectrum

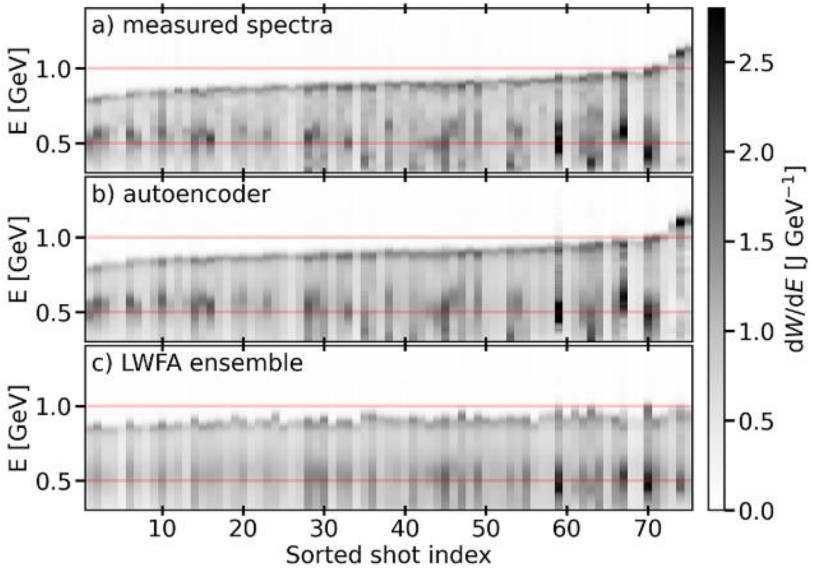




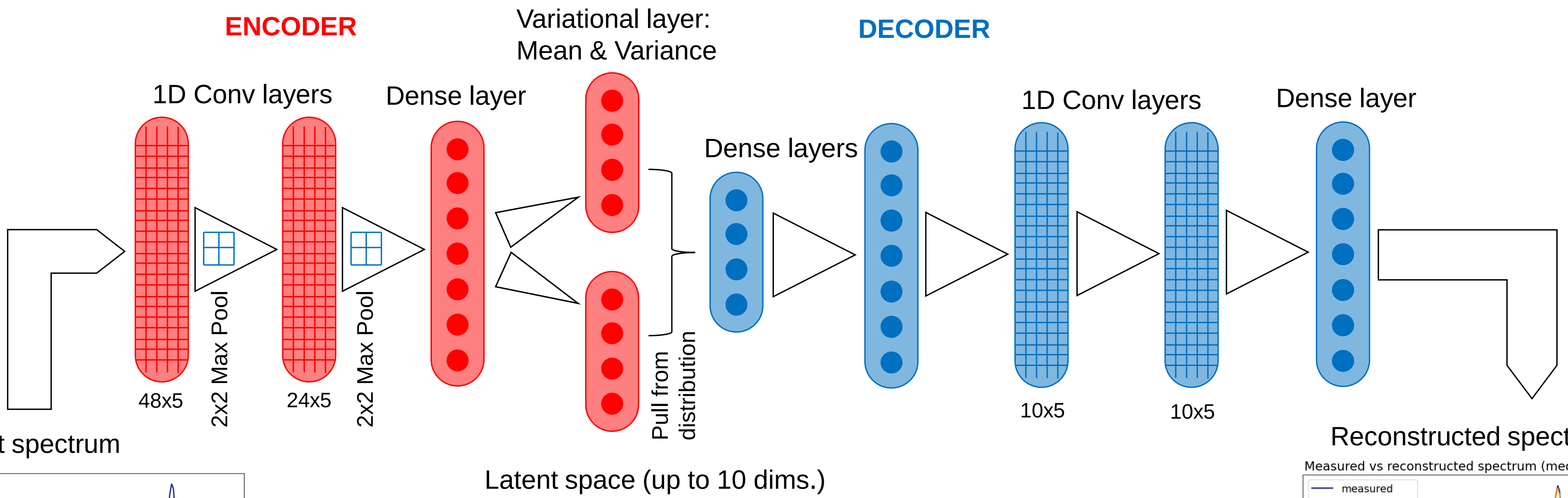
- Main observation: Prediction mostly defaults to the mean spectrum, outliers not well captured

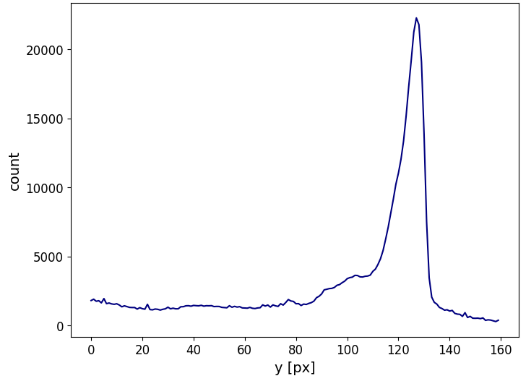
Change to new network architecture

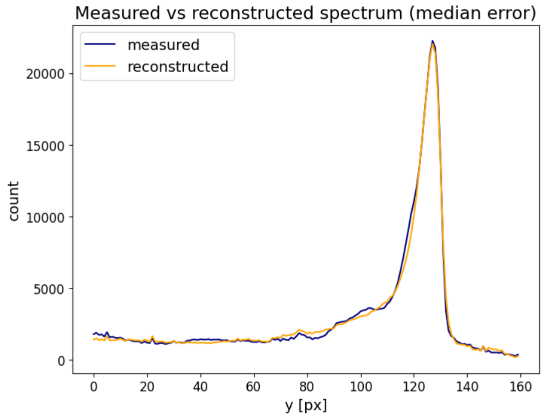
- Include some convolutional layers to introduce awareness of spatial relationships in the spectra
- All spectra look 'similar' → Encode them in fewer dimensions (latent space)
- Many different inputs can be mapped to latent space!
- Example from literature:
  - Prediction of electron spectra from **LWFA** (M. J. V. Streeter *et al.*, *High Power Laser Science and Engineering*, 2023)



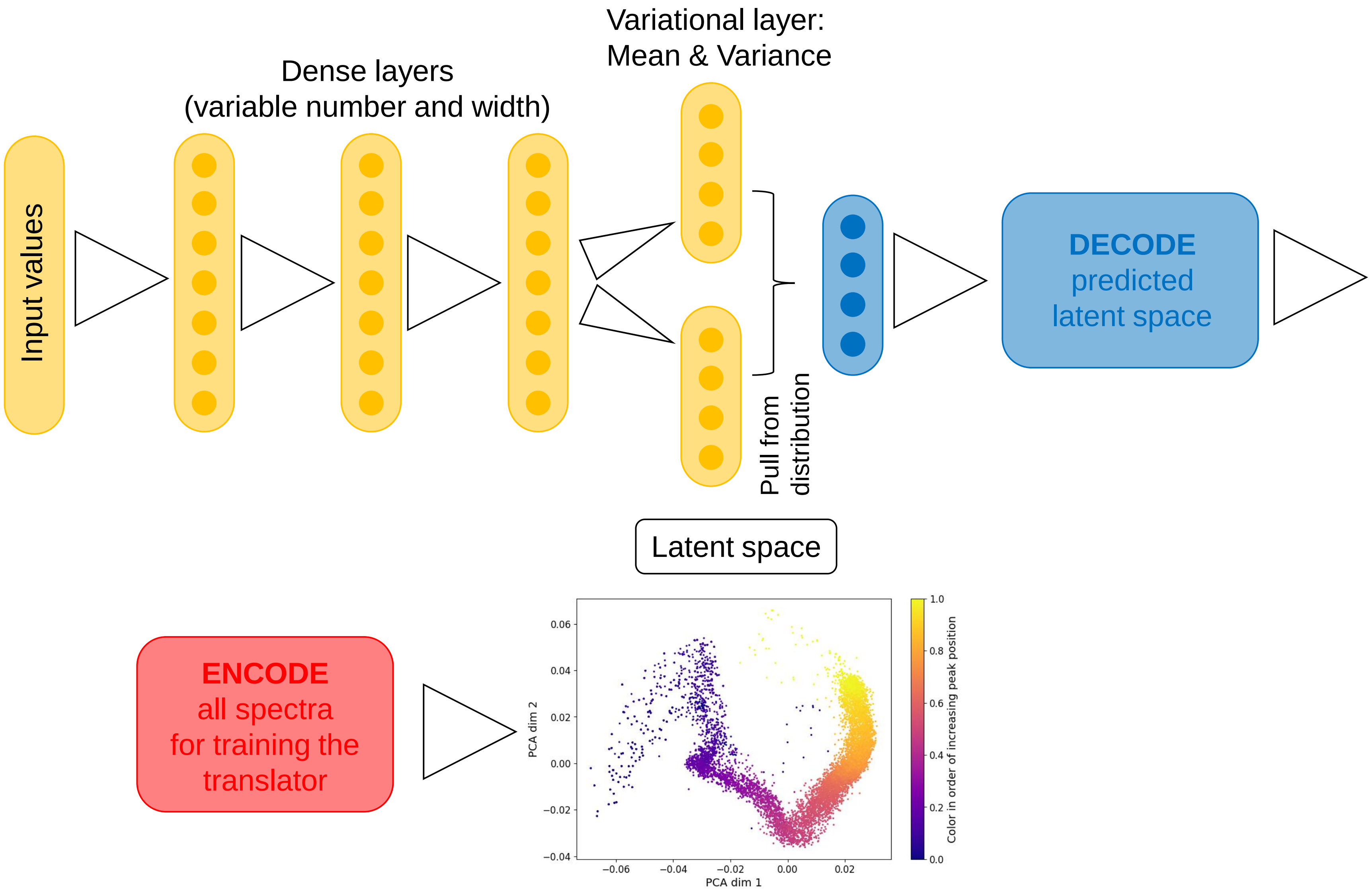
Convolutional variational autoencoder



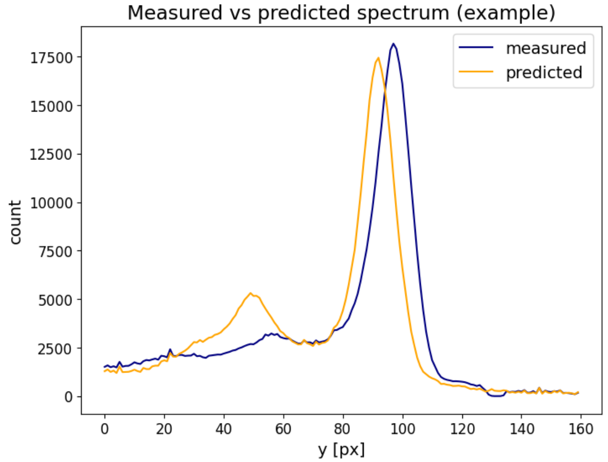




Translator

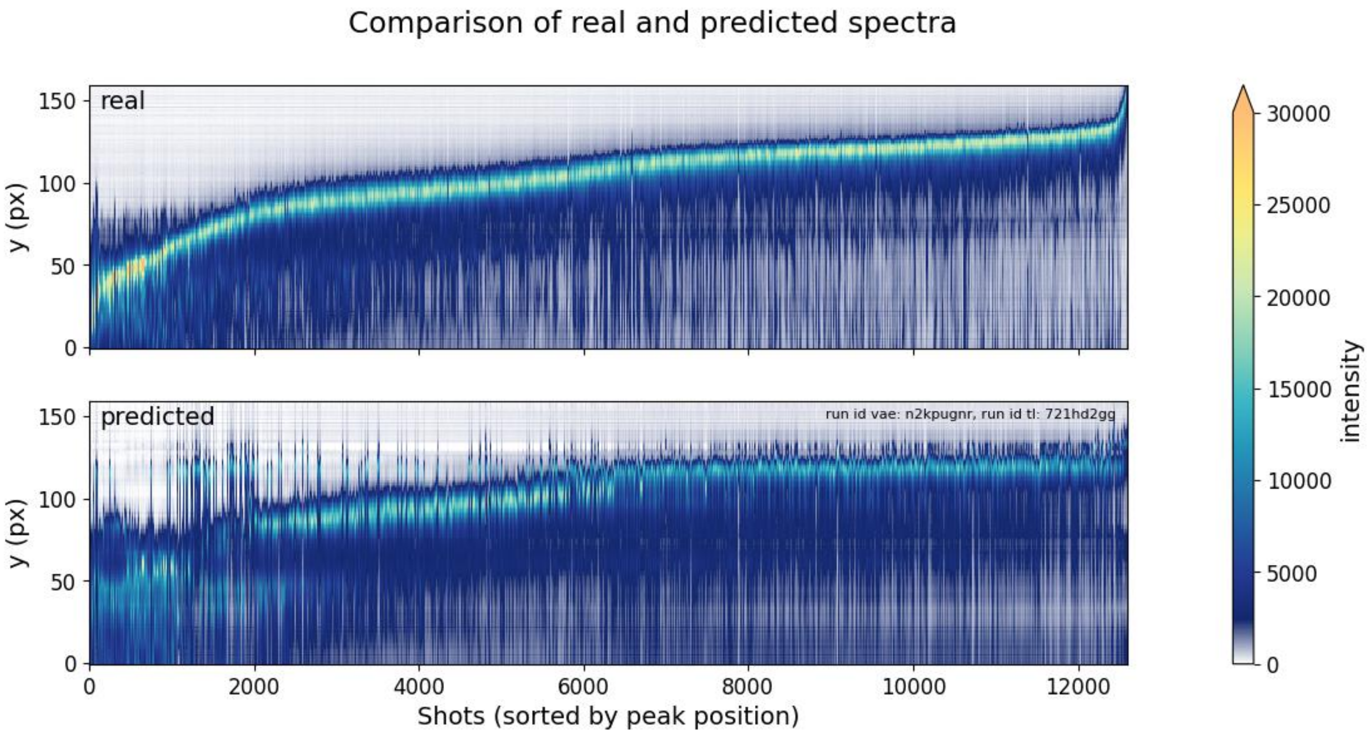


Variational layer:
Mean & Variance
Dense layers
(variable number and width)
Input values
Pull from
distribution
DECODE
predicted
latent space
Latent space
ENCODE
all spectra
for training the
translator
PCA dim 1
PCA dim 2
Color in order of increasing peak position

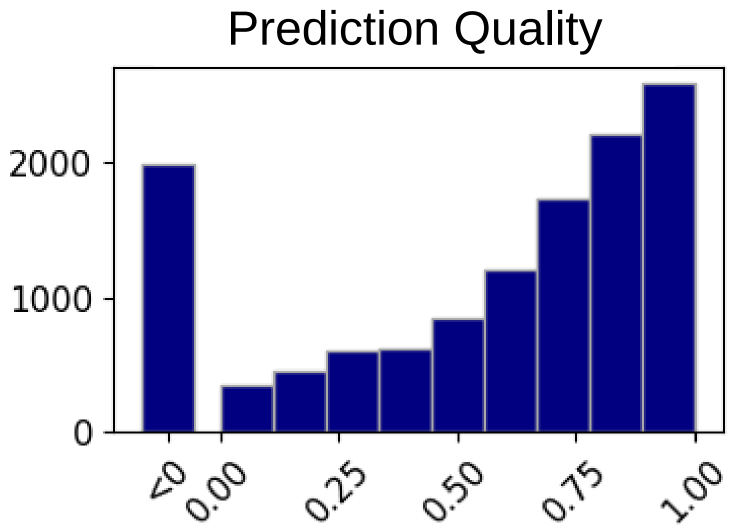


Measured vs predicted spectrum (example)
measured
predicted
count
y [px]

Results & How to improve them



- Prediction works best for medium energy shots
- High energy shots get grouped together
  - Maybe not enough variation in input data?



- Do hyperparameter optimization
- Include longer dataset with active changes to the beamline settings
- Include more inputs

Summary

- Ran experiments with two different model architectures trying to predict electron spectra
  1. Densely connected network → bad prediction of outlying shots
  2. Conv VAE + Translator → seems promising, needs further refinement
- Next steps:
  - Sweep hyperparameters more
  - Use a longer span of data to improve prediction quality → investigate performance over time
  - Include more different inputs, e.g. encode sideview images of the plasma