Neural Network Approaches for Quasar and Galaxy continuum Estimation

Speaker:
Francesco Pistis[1,2,3]
In collaboration with:
Michele Fumagalli[1,4]
Matteo Fossati[1,5]

Affiliation(s):
[1] Università Milano-Bicocca
[2] National Centre for Nuclear Research
[3] INAF-OAS
[4] INAF-OAT
[5] INAF-OAB

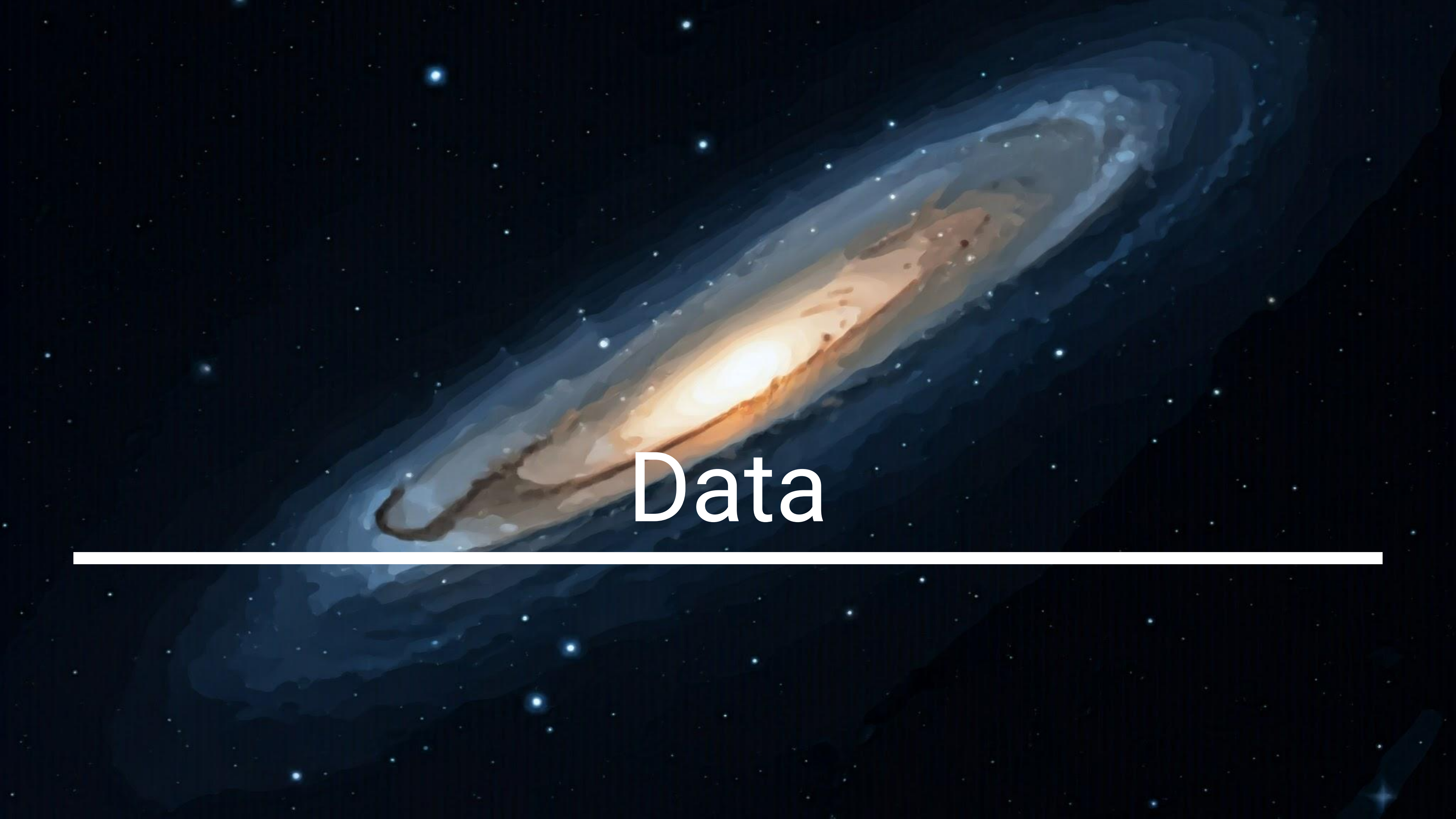
Università degli Studi di Milano Bicocca
Funded by the European Union
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
Piano Nazionale di Ripresa e Resilienza

Scientific goal

Scientific goal of the project

- Automated pipeline for WEAVE survey
- Fit of the quasar continua
- Detection of absorption lines
- Measurement of absorption lines

Data



Data training and physical test

WEAVE quasar catalog (mock)

- Continua from PCA (Paris et al. 2011)
- Lyα forest: Sherwood simulation
- LLS and DLAs following distribution as done in Fumagalli et al. 2020
- Metals random draw from distribution (empirical approach)
- Noise and instrumental properties with WEAVEify (WEAVE survey simulation)
- Sample size: 30 000 mock spectra (after redshift and magnitude selections)

VIPERS galaxy catalog (observed)

- Catalog of observed galaxies
- Sample size: 50 000 (after selection on redshift and its quality)

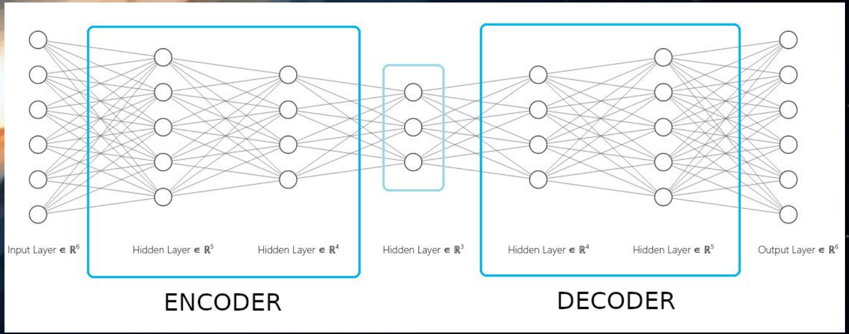
DESI Early Data Release (observed)

- Catalog of observed quasars and galaxies
- Selection on redshift (to recover the other catalog distributions)
- Selection to remove BALs (Filbert et al. 2023)

Methodology

NN 1: Autoencoder

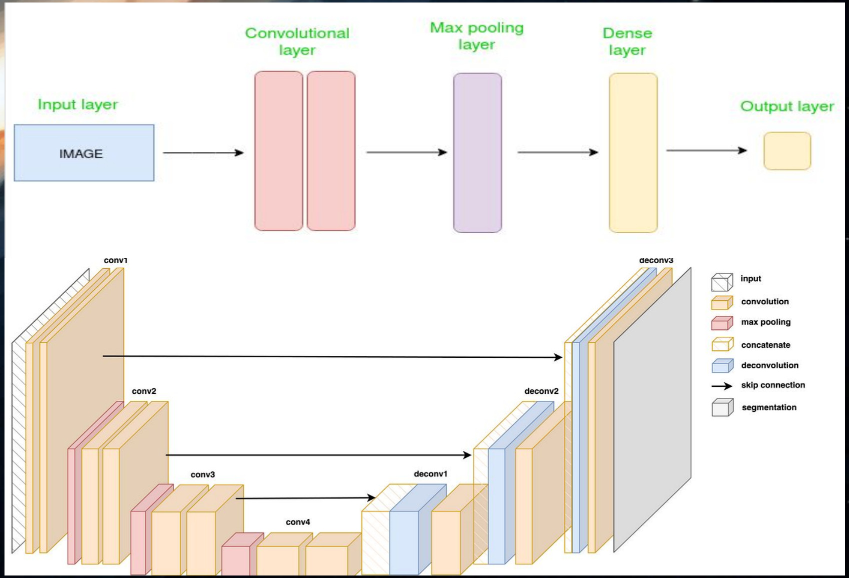
- Designed to have an output similar to the input
- Training: the NN learns how to reproduce the input
- Encoder: compressing the data
- Middle layer: the NN learns a compressed representation of the data
- Decoder: recover of the data from a compressed representation
- **Compared to intelligent quasar continuum neural network (iQNet) [Liu & Bordoloi 2021]**



Credits: A friendly introduction to autoencoders.

NN 2: CNN & U-Net

- Skip connections (U-Net): high-resolution features are preserved and propagated through the network
- Convolutional layer: add a set of filters (kernels) to the input
- Max pooling: reduce the dimension giving the max values in the filters
- **Compared to Lyman-α Continuum Analysis Network (LyCAN) [Turner et al. 2024]**



Credits: [Introduction to Convolution Neural Network - GeeksforGeeks](#)

[U-Net Explained: Understanding its Image Segmentation Architecture](#)

Optimization: Autoencoder

- Not a traditional autoencoder architecture
- Activation function not equal for all layers
- Same architecture for different inputs

Table 1. Architecture of the optimized autoencoders applied to WEAVE (red and full spectra) and VIPERS data.

Model	Layer Type	Output Shape	Additional Info
WEAVE (red part)	InputLayer	(None, 3920)	
	Masking	(None, 3920)	
	Dense	(None, 896)	Activation: linear, Units: 896
	Activation	(None, 896)	Activation: ReLU
	Dense	(None, 128)	Activation: linear, Units: 128
	Activation	(None, 128)	Activation: ELU
	Dense	(None, 896)	Activation: linear, Units: 896
	Activation	(None, 896)	Activation: ReLU
	Dense	(None, 384)	Activation: linear, Units: 384
	Activation	(None, 384)	Activation: ELU
	Dense	(None, 512)	Activation: linear, Units: 512
	Activation	(None, 512)	Activation: ELU
	Dense	(None, 128)	Activation: linear, Units: 128
	Activation	(None, 128)	Activation: ReLU
	Dense	(None, 768)	Activation: linear, Units: 768
	Activation	(None, 768)	Activation: ELU
	Dense	(None, 4900)	Activation: linear, Units: 4900
WEAVE (full spectra)	InputLayer	(None, 4900)	
	Masking	(None, 4900)	
	Dense	(None, 896)	Activation: linear, Units: 896
	Activation	(None, 896)	Activation: ReLU
	Dense	(None, 128)	Activation: linear, Units: 128
	Activation	(None, 128)	Activation: ELU
	Dense	(None, 896)	Activation: linear, Units: 896
	Activation	(None, 896)	Activation: ReLU
	Dense	(None, 384)	Activation: linear, Units: 384
	Activation	(None, 384)	Activation: ELU
	Dense	(None, 512)	Activation: linear, Units: 512
	Activation	(None, 512)	Activation: ELU
	Dense	(None, 128)	Activation: linear, Units: 128
	Activation	(None, 128)	Activation: ReLU
	Dense	(None, 768)	Activation: linear, Units: 768
	Activation	(None, 768)	Activation: ELU
	Dense	(None, 4900)	Activation: linear, Units: 4900
VIPERS	InputLayer	(None, 2000)	
	Masking	(None, 2000)	
	Dense	(None, 512)	Activation: linear, Units: 512
	Activation	(None, 512)	Activation: ReLU
	Dense	(None, 896)	Activation: linear, Units: 896
	Activation	(None, 896)	Activation: ELU
	Dense	(None, 128)	Activation: linear, Units: 128
	Activation	(None, 128)	Activation: ELU
	Dense	(None, 1024)	Activation: linear, Units: 1024
	Activation	(None, 1024)	Activation: ReLU
	Dense	(None, 768)	Activation: linear, Units: 768
	Activation	(None, 768)	Activation: ReLU
	Dense	(None, 2000)	Activation: linear, Units: 2000

Notes. The activation functions used are the exponential linear unit (ELU) and the rectified linear unit (ReLU) function.

Credits: Pistis et al. (in prep.)

Optimization: U-Net

- Kernel sizes not equal for all layers
- Some layers sizes are repeated

Table 2. Architecture of the optimized U-Nets applied to WEAVE and VIPERS data.

Model	Layer Type	Output Shape	Additional Info
WEAVE	InputLayer	(None, 5000, 1)	
	Masking	(None, 5000, 1)	
	Conv1D	(None, 5000, 16)	Kernel Size: (5,), Activation: ReLU
	MaxPooling1D	(None, 2500, 16)	Pool Size: 2
	Conv1D	(None, 2500, 16)	Kernel Size: (5,), Activation: ReLU
	MaxPooling1D	(None, 1250, 16)	Pool Size: 2
	Conv1D	(None, 1250, 32)	Kernel Size: (5,), Activation: ReLU
	MaxPooling1D	(None, 625, 32)	Pool Size: 2
	Conv1DTranspose	(None, 1250, 32)	Kernel Size: (2,), Activation: ReLU
	Concatenate	(None, 1250, 64)	
	Conv1D	(None, 1250, 32)	Kernel Size: (5,), Activation: ReLU
	Conv1DTranspose	(None, 2500, 16)	Kernel Size: (2,), Activation: ReLU
	Concatenate	(None, 2500, 32)	
	Conv1D	(None, 2500, 16)	Kernel Size: (5,), Activation: ReLU
	Conv1DTranspose	(None, 5000, 16)	Kernel Size: (2,), Activation: ReLU
	Concatenate	(None, 5000, 32)	
	Conv1D	(None, 5000, 16)	Kernel Size: (5,), Activation: ReLU
	Conv1D	(None, 5000, 1)	Kernel Size: (1,), Activation: linear
VIPERS	InputLayer	(None, 2000, 1)	
	Masking	(None, 2000, 1)	
	Conv1D	(None, 2000, 8)	Kernel Size: (5,), Activation: ReLU
	MaxPooling1D	(None, 1000, 8)	Pool Size: 2
	Conv1D	(None, 1000, 16)	Kernel Size: (5,), Activation: ReLU
	MaxPooling1D	(None, 500, 16)	Pool Size: 2
	Conv1D	(None, 500, 64)	Kernel Size: (5,), Activation: ReLU
	MaxPooling1D	(None, 250, 64)	Pool Size: 2
	Conv1DTranspose	(None, 500, 64)	Kernel Size: (2,), Activation: ReLU
	Concatenate	(None, 500, 128)	
	Conv1D	(None, 500, 64)	Kernel Size: (5,), Activation: ReLU
	Conv1DTranspose	(None, 1000, 16)	Kernel Size: (2,), Activation: ReLU
	Concatenate	(None, 1000, 32)	
	Conv1D	(None, 1000, 16)	Kernel Size: (5,), Activation: ReLU
	Conv1DTranspose	(None, 2000, 8)	Kernel Size: (2,), Activation: ReLU
	Concatenate	(None, 2000, 16)	
	Conv1D	(None, 2000, 8)	Kernel Size: (5,), Activation: ReLU
	Conv1D	(None, 2000, 1)	Kernel Size: (1,), Activation: linear

Notes. The activation function used is the rectified linear unit (ReLU) function.

Credits: Pistis et al. (in prep.)

Optimization: Convolutional NN

- Not a traditional CNN architecture
- Kernel sizes not equal for all layers
- Activation function not equal for all layers
- Same architecture for different Inputs

Table 3. Architecture of the optimized CNNs applied to WEAVE (red and full spectra) and VIPERS data.

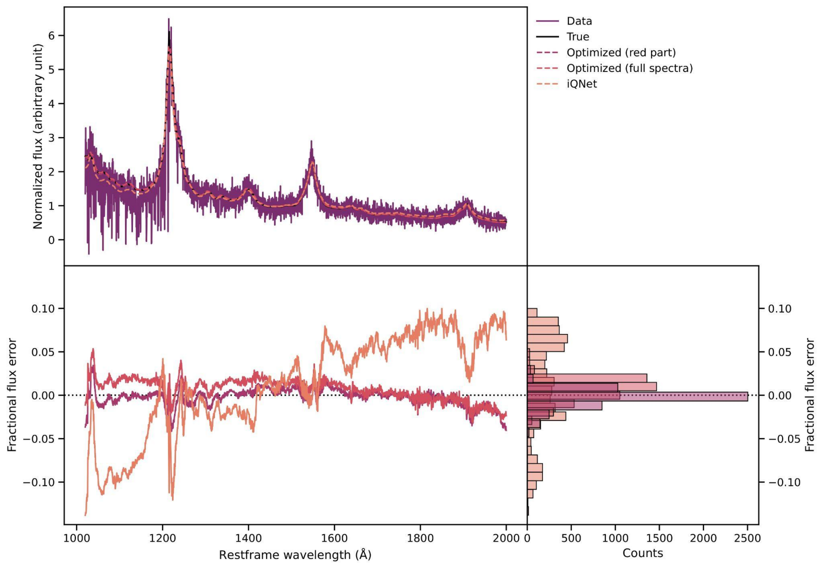
Model	Layer Type	Output Shape	Additional Info
WEAVE (red part)	InputLayer	(None, 3920, 1)	
	Conv1D	(None, 3920, 256)	Kernel Size: (5,), Activation: ELU
	MaxPooling1D	(None, 1960, 256)	
	Conv1D	(None, 1960, 192)	Kernel Size: (3,), Activation: ReLU
	MaxPooling1D	(None, 653, 192)	
	Conv1D	(None, 653, 64)	Kernel Size: (3,), Activation: ReLU
	MaxPooling1D	(None, 217, 64)	
	Conv1D	(None, 217, 160)	Kernel Size: (3,), Activation: ELU
	MaxPooling1D	(None, 72, 160)	
	Dropout	(None, 72, 160)	Dropout Rate: 0.2
	Flatten	(None, 11520)	
	Dense	(None, 512)	Activation: ReLU, Units: 512
	Dense	(None, 1024)	Activation: ReLU, Units: 1024
	Dense	(None, 4900)	Activation: linear, Units: 4900
WEAVE (full spectra)	InputLayer	(None, 4900, 1)	
	Conv1D	(None, 4900, 256)	Kernel Size: (5,), Activation: ELU
	MaxPooling1D	(None, 2450, 256)	
	Conv1D	(None, 2450, 192)	Kernel Size: (3,), Activation: ReLU
	MaxPooling1D	(None, 816, 192)	
	Conv1D	(None, 816, 64)	Kernel Size: (3,), Activation: ReLU
	MaxPooling1D	(None, 272, 64)	
	Conv1D	(None, 272, 160)	Kernel Size: (3,), Activation: ELU
	MaxPooling1D	(None, 90, 160)	
	Dropout	(None, 90, 160)	Dropout Rate: 0.2
	Flatten	(None, 14400)	
	Dense	(None, 512)	Activation: ReLU, Units: 512
	Dense	(None, 1024)	Activation: ReLU, Units: 1024
	Dense	(None, 4900)	Activation: linear, Units: 4900
VIPERS	InputLayer	(None, 2000, 1)	
	Conv1D	(None, 2000, 256)	Kernel Size: (5,), Activation: ELU
	MaxPooling1D	(None, 1000, 256)	
	Conv1D	(None, 1000, 192)	Kernel Size: (3,), Activation: ReLU
	MaxPooling1D	(None, 333, 192)	
	Conv1D	(None, 333, 64)	Kernel Size: (3,), Activation: ReLU
	MaxPooling1D	(None, 111, 64)	
	Conv1D	(None, 111, 160)	Kernel Size: (3,), Activation: ELU
	MaxPooling1D	(None, 37, 160)	
	Dropout	(None, 37, 160)	Dropout Rate: 0.2
	Flatten	(None, 5920)	
	Dense	(None, 512)	Activation: ReLU, Units: 512
	Dense	(None, 1024)	Activation: ReLU, Units: 1024
	Dense	(None, 2000)	Activation: linear, Units: 2000

Notes. The activation functions used are the rectified linear unit (ReLU), exponential linear unit (ELU), and linear activation.

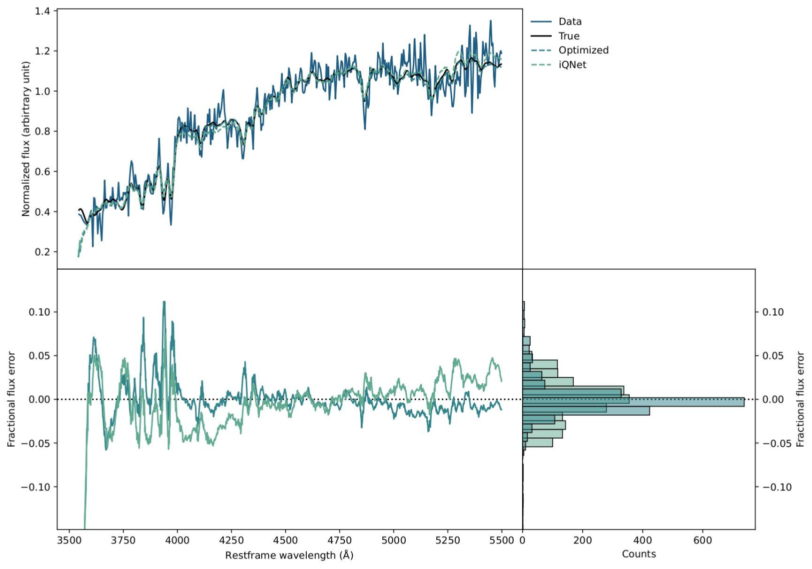
Credits: Pistis et al. (in prep.)

Results

Application: autoencoders

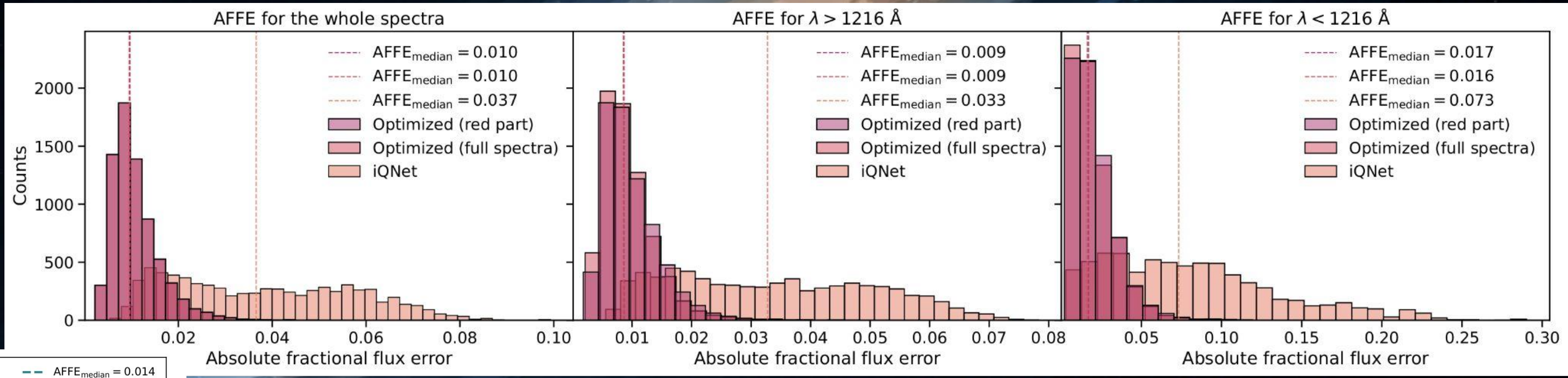


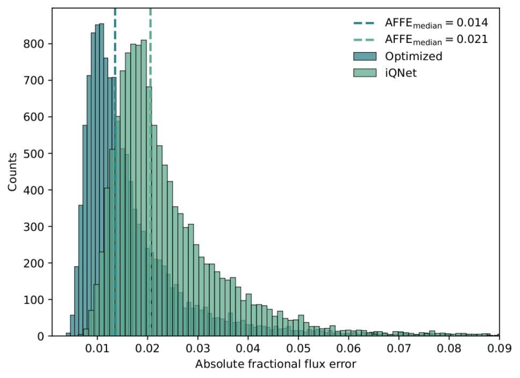
Credits: Pistis et al. (in prep.)



Credits: Pistis et al. (in prep.)

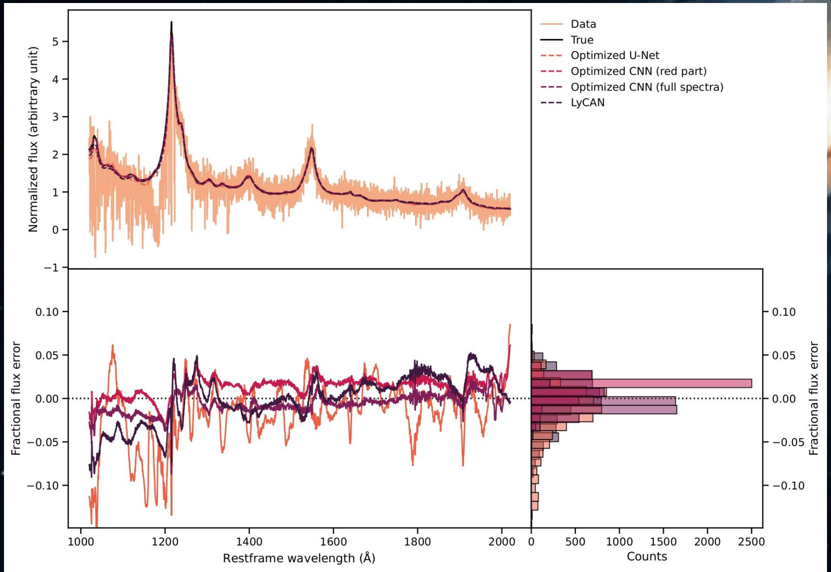
Application: autoencoders



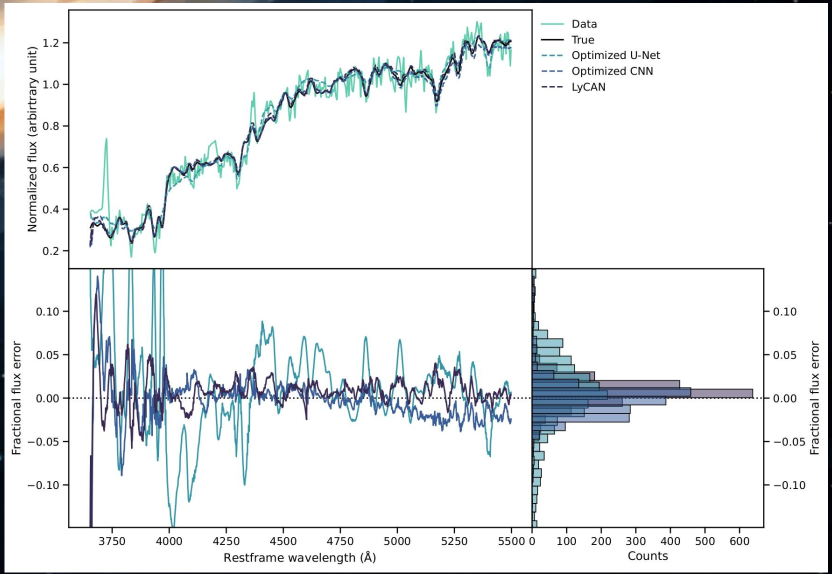


Credits: Pistis et al. (in prep.)

Application: U-Net and CNNs

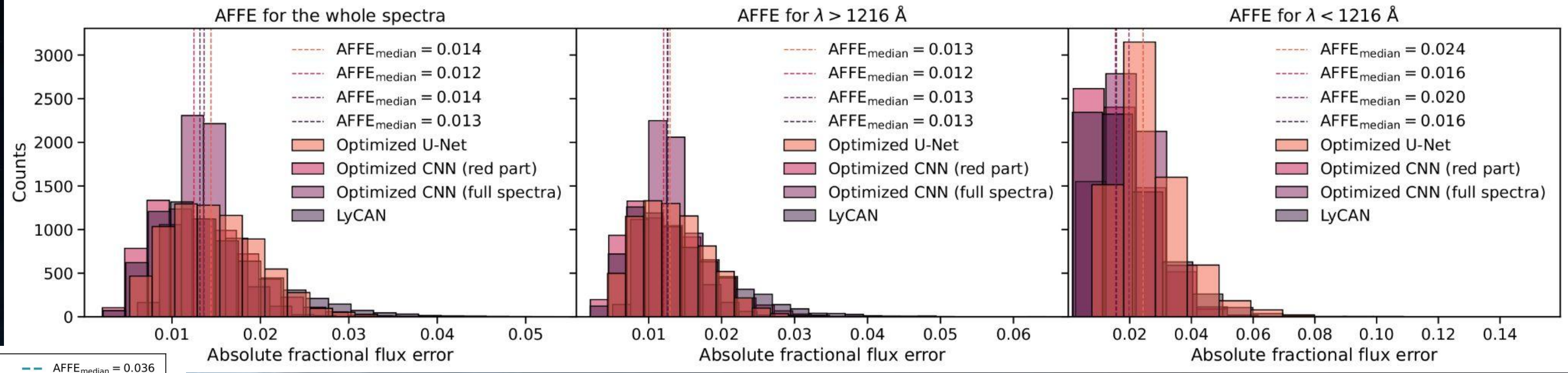


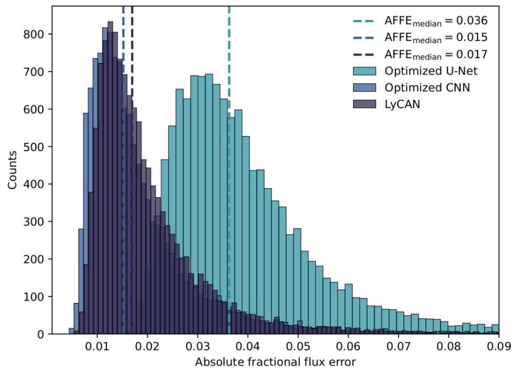
Credits: *Pistis et al. (in prep.)*



Credits: *Pistis et al. (in prep.)*

Application: U-Net and CNNs



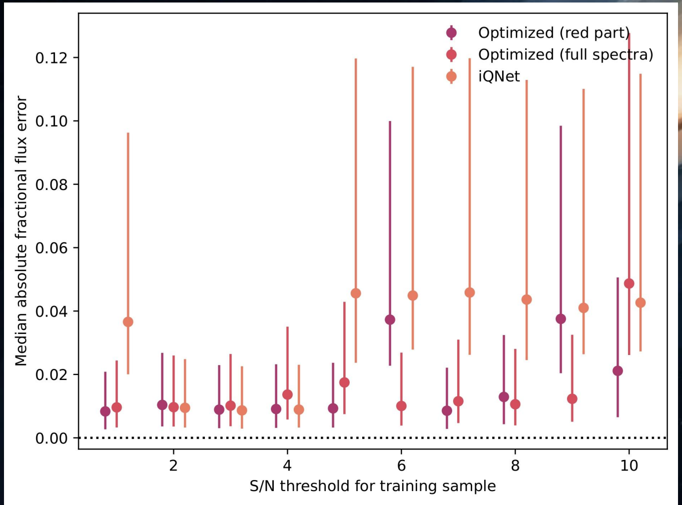


Credits: Pistis et al. (in prep.)

Tests

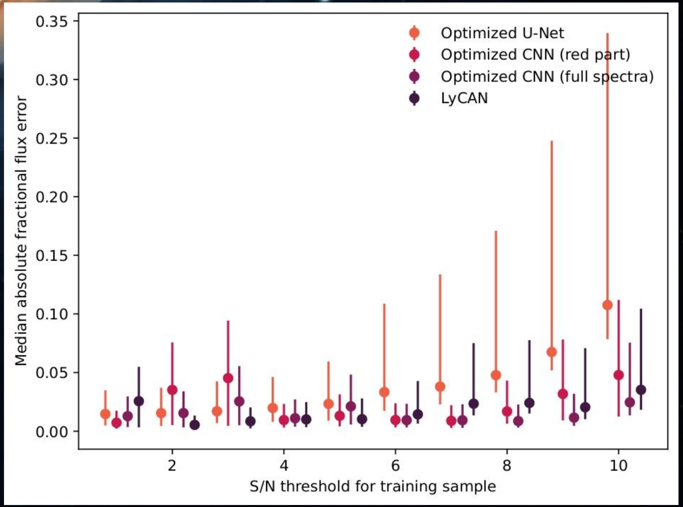
Test 1: S/N threshold for training sample

Autoencoders



Credits: Pistis et al. (in prep.)

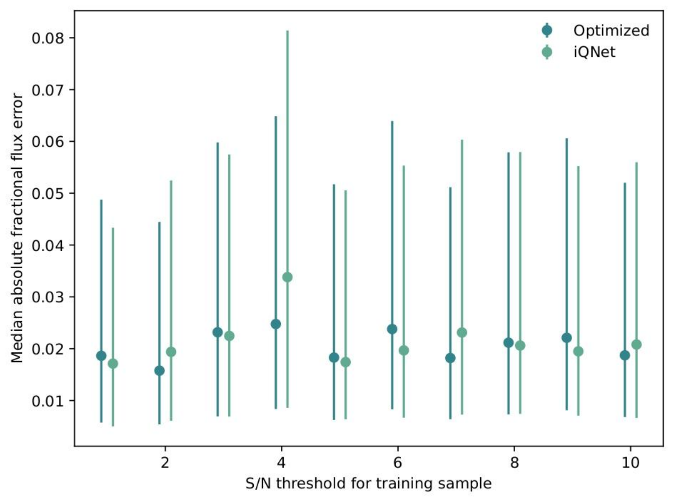
U-Net & CNNs



Credits: Pistis et al. (in prep.)

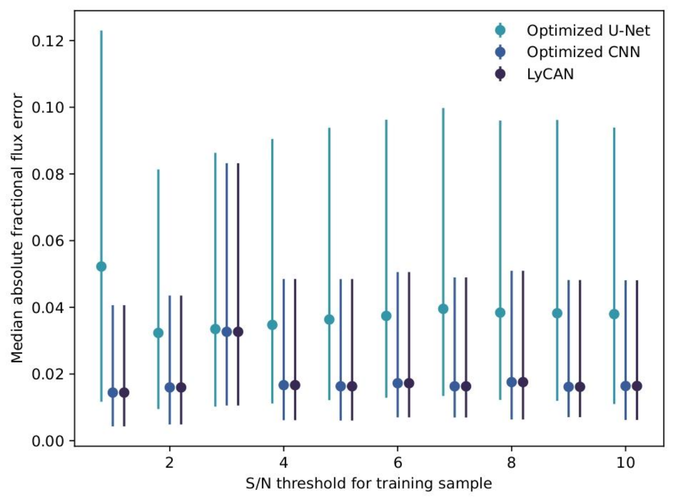
Test 1: S/N threshold for training sample

Autoencoders



Credits: Pistis et al. (in prep.)

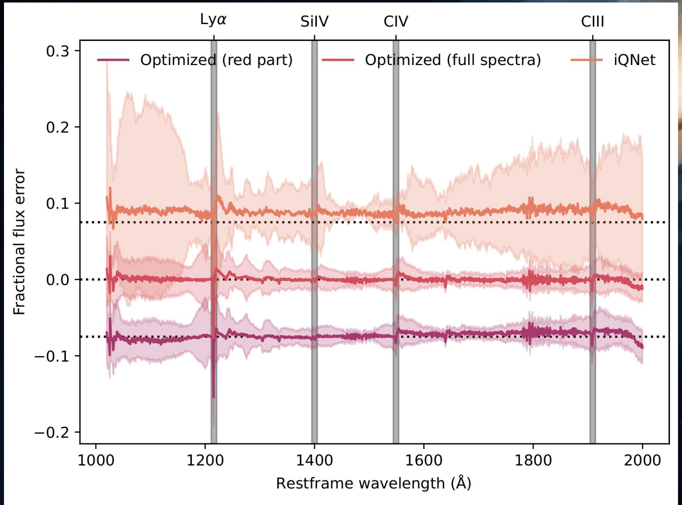
U-Net & CNNs



Credits: Pistis et al. (in prep.)

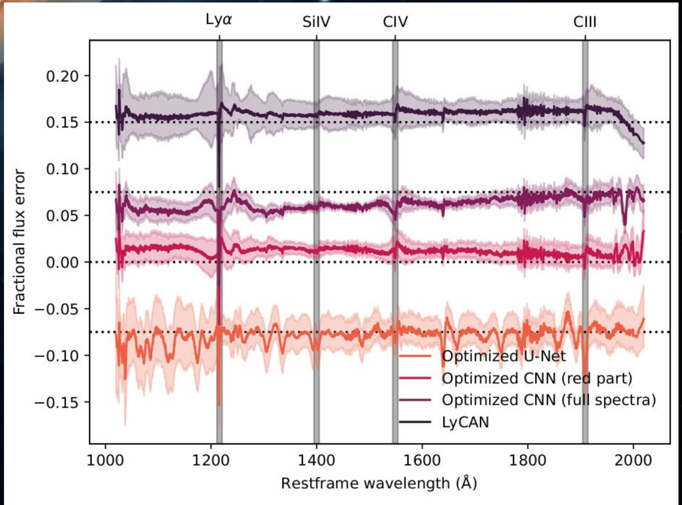
Test 2: Bias in wavelength

Autoencoders



Credits: Pistis et al. (in prep.)

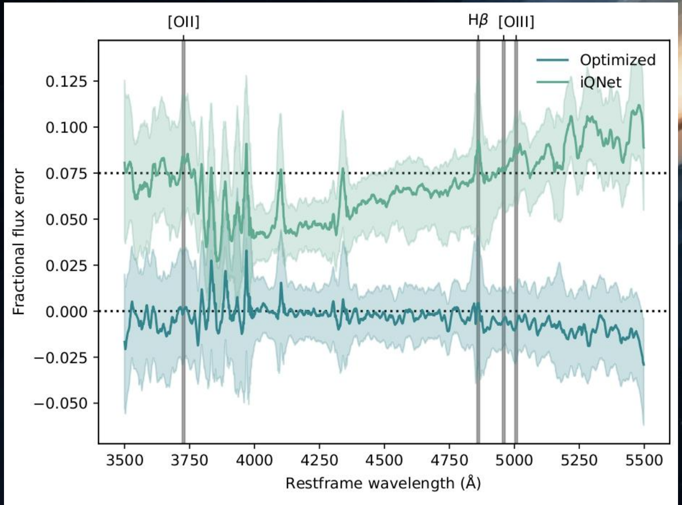
U-Net & CNNs



Credits: Pistis et al. (in prep.)

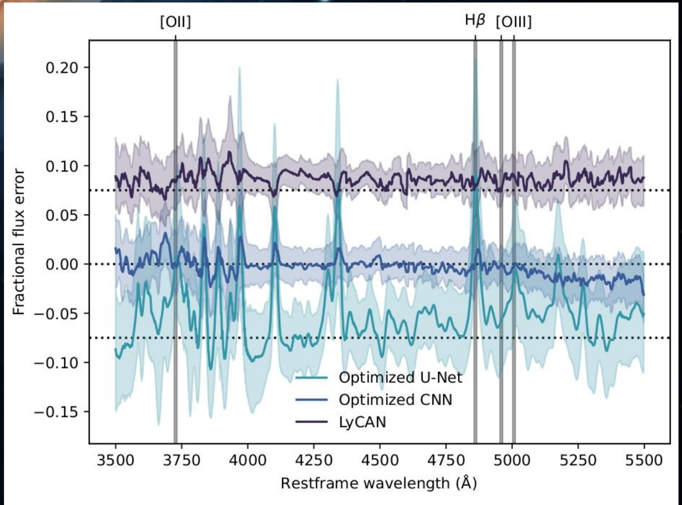
Test 2: Bias in wavelength

Autoencoders



Credits: Pistis et al. (in prep.)

U-Net & CNNs



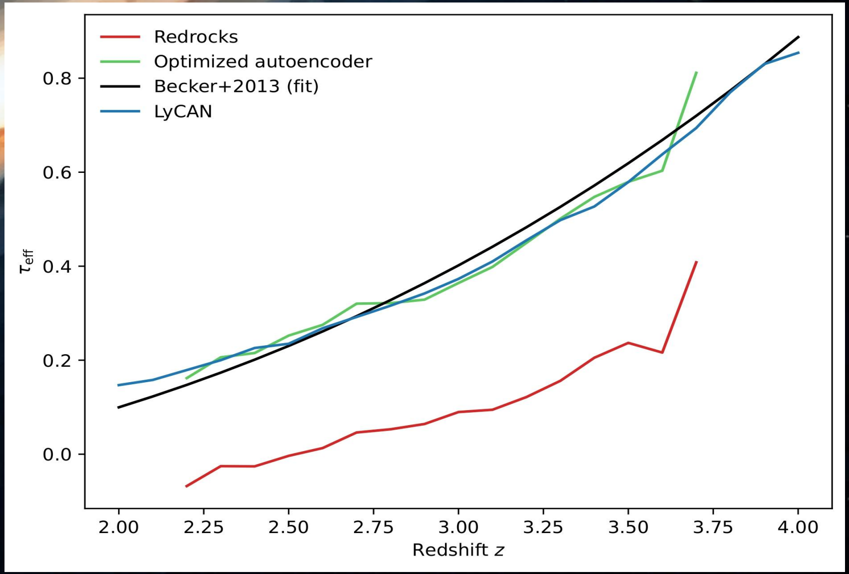
Credits: Pistis et al. (in prep.)

Test 3 (quasars): Lyα forest optical depth – DESI

- Direct applications of the weights of the training on the WEAVE mock to DESI EDR
- 1070 Å < λ < 1160 Å
- At each $z_{Ly\alpha}$ bin (0.01) we compute the median transmitted flux $\langle f \rangle$
- Optical depth as

$$\tau = -\,log(\langle f \rangle)$$

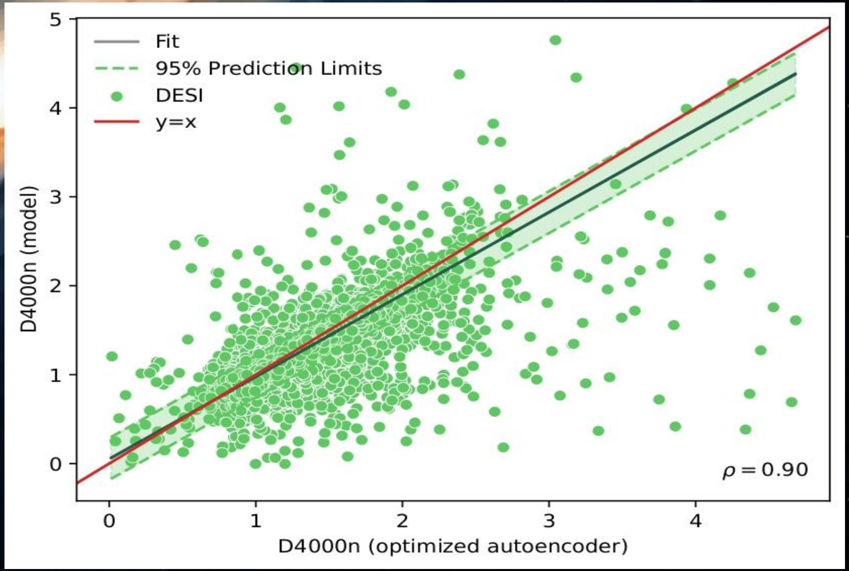
- No correction applied



Credits: Pistis et al. (in prep.)

Test 3 (galaxies): D4000n break DESI

- Comparison between D4000n break computed on the predicted continua with NN and the model fit with Redrocks



Credits: Pistis et al. (in prep.)

Conclusions

Conclusions

- **Autoencoders have better performances for both quasars and galaxies (AFFE lower of ~0.2%)**
- **Autoencoders have much lower computational times (factor of ~100 times)**
- **No bias for the S/N threshold on the training sample**
- **No bias on the FFE vs wavelength**
- **Good measurement of the optical depth evolution in the Lyα forest region**
- **Good measurement for the D4000n break**

Thank you for your attention!