Investigating the modelling of delayed gamma from nuclear fission with the help of multi-dimensional gamma spectroscopy

Matteo Ballu

May 16th 2024

Table of Contents

Context and objective

Experimental setup and previous analysis

Building a new analysis tool with machine learning

Extras

Table of Contents

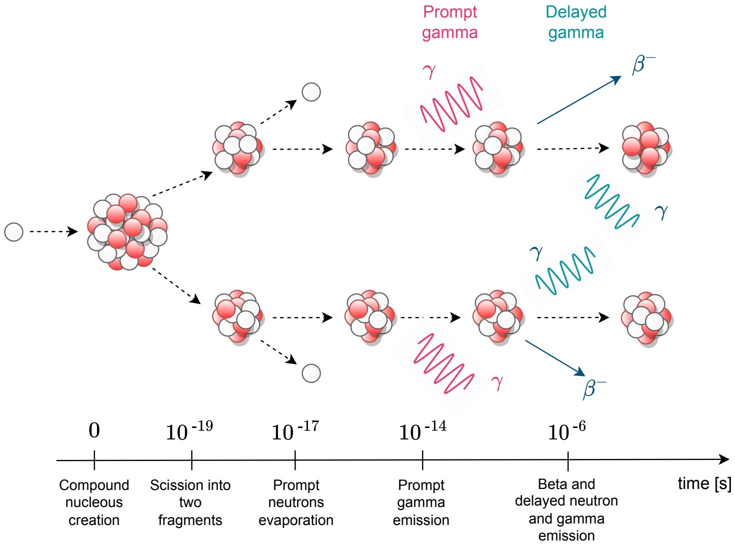
Context and objective

Experimental setup and previous analysis

Building a new analysis tool with machine learning

Extras

Neutron induced fission



- Gamma rays : photon produced by the radioactive decay of a nucleus
- energy : 100 keV - 8 MeV

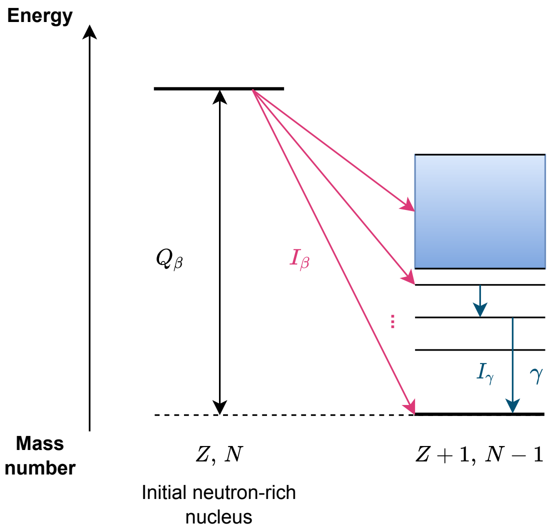
Beta decay for neutron-rich nuclei

β^- decay :

$$\boxed{{}^{A}_{Z}\mathrm{X} \longrightarrow {}_{Z+1}^{\ \ A}\mathrm{Y} + e^- + \bar{\nu}_e}$$

$$Q_\beta = E_{e^-} + E_{\bar{\nu}_e} + E^*$$

I_β gives the probability that the Y nucleus is produced at a given excited state



How to get I_β ?

- I_β can be estimate from I_γ thanks to gamma spectroscopy
- *pandemonium effect* : missing transitions and bias in intensities

Objective of the thesis

provide experimental verification of fission-delayed gamma-ray modelling
Include:

- beta decay process
- fission fragment deexcitation

but strongly depends on the *speed* of the analysis of the data we have

Table of Contents

Context and objective

Experimental setup and previous analysis

Building a new analysis tool with machine learning

Extras

Description of the experimental setup

Description of the experimental setup

The neutron source

- experimental nuclear reactor
- very high flux : 10^8 s^{-1} cm^{-2} at the target

Description of the experimental setup

The neutron source

- experimental nuclear reactor
- very high flux : 10^8 s^{-1} cm^{-2} at the target

The *active* target

uranium 235 diluted in a scintillating liquid

Interest : gamma emitted after a fission can be tagged

Description of the experimental setup

The neutron source

- experimental nuclear reactor
- very high flux : 10^8 s^{-1} cm^{-2} at the target

The *active* target

uranium 235 diluted in a scintillating liquid

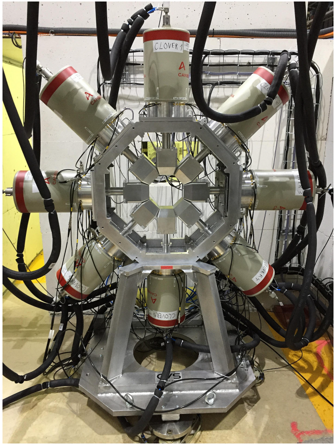
Interest : gamma emitted after a fission can be tagged

The detection system (FIPPS gamma-ray spectrometer)

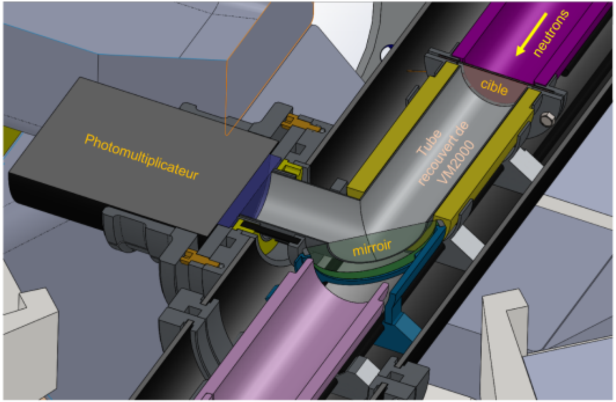
Composed of two parts :

- 16 high-purity germanium clover detectors (HPGe)
- Photo-multiplier : collect the light produced at the target

Description of the experimental setup



(a) 8 of the 16 germanium spectrometers



(b) The target and the light collecting system

Pre-analysis

Raw data treatment

- Validation of the *fission tag* (other PhD student) ✓
- Energy calibration (other PhD student and myself) ✓
- Time calibration (other PhD student) ✓
- Produce the coincidence matrix and cube ✓

Pre-analysis

Raw data treatment

- Validation of the *fission tag* (other PhD student) ✓
- Energy calibration (other PhD student and myself) ✓
- Time calibration (other PhD student) ✓
- Produce the coincidence matrix and cube ✓

Detector GEANT4 simulation

- needed to estimate summation effects

Pre-analysis

Raw data treatment

- Validation of the *fission tag* (other PhD student) ✓
- Energy calibration (other PhD student and myself) ✓
- Time calibration (other PhD student) ✓
- Produce the coincidence matrix and cube ✓

Detector GEANT4 simulation

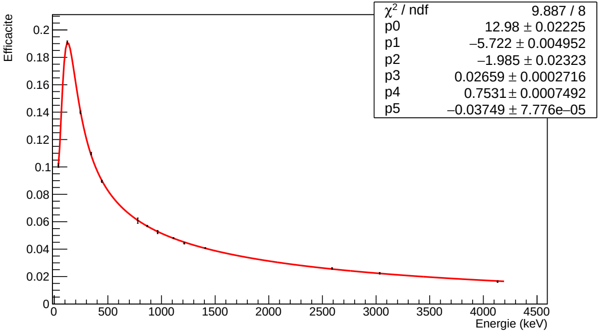
- needed to estimate summation effects

Detector properties

- dead time ✓
- efficiency ✓

Efficiency

- calibration source : europium 152
- relies on a good knowledge of the source's activity which was not easy to estimate
- relative uncertainty around $1\,\%$



Previous analysis

What has been done

- Independent fission yield for a dozen of fission fragments [a]
- the analysis relies extensively on peak fitting on 2d or 3d spectra to extract the number of measured gamma for a given transition
- the current methods are time consuming and the uncertainty can be large

[a] *The nuclear fission process in the light of prompt gamma-rays : measurement of thermal fission yields of U-235 on the FIPPS spectrometer*, P. Herran (2023)

Limitations

- we want to analyse hundreds of peaks with a good control on the uncertainty
- an automatic (or at least semi-automatic) procedure would be welcomed

Table of Contents

Context and objective

Experimental setup and previous analysis

Building a new analysis tool with machine learning

Extras

The motto of every programmer

How to build a machine learning model ?

How to build a machine learning model ?

1. Decide on a problem : *what are we modeling ? what is the task we want to solve ?*

How to build a machine learning model ?

1. Decide on a problem : *what are we modeling ? what is the task we want to solve ?*
2. Curate data : *what data will inform the model ?* **This is the hard part**

How to build a machine learning model ?

1. Decide on a problem : *what are we modeling ? what is the task we want to solve ?*
2. Curate data : *what data will inform the model ?* **This is the hard part**
3. Design an architecture : CNN ? RNN ? Auto encoder ? **This is the fun part**

How to build a machine learning model ?

1. Decide on a problem : *what are we modeling ? what is the task we want to solve ?*
2. Curate data : *what data will inform the model ?* **This is the hard part**
3. Design an architecture : CNN ? RNN ? Auto encoder ? **This is the fun part**
4. Craft a loss function : *what models are "good" ?*

How to build a machine learning model ?

1. Decide on a problem : *what are we modeling ? what is the task we want to solve ?*
2. Curate data : *what data will inform the model ?* **This is the hard part**
3. Design an architecture : CNN ? RNN ? Auto encoder ? **This is the fun part**
4. Craft a loss function : *what models are "good" ?*
5. Employ optimization : *what algorithms to train the model ?*

How to build a machine learning model ?

1. Decide on a problem : *what are we modeling ? what is the task we want to solve ?*
2. Curate data : *what data will inform the model ?* **This is the hard part**
3. Design an architecture : CNN ? RNN ? Auto encoder ? **This is the fun part**
4. Craft a loss function : *what models are "good" ?*
5. Employ optimization : *what algorithms to train the model ?*

We are physicists, how can we introduce some physics ?

Which data to train the model ?

Problem

There is no labeled data available

Solution

We simulate our own data : synthetic data set

- Pro : full control
- Con : possible bias and missing things

New problem

How to truthfully reproduce the data we observe ?

Synthetic dataset : an example

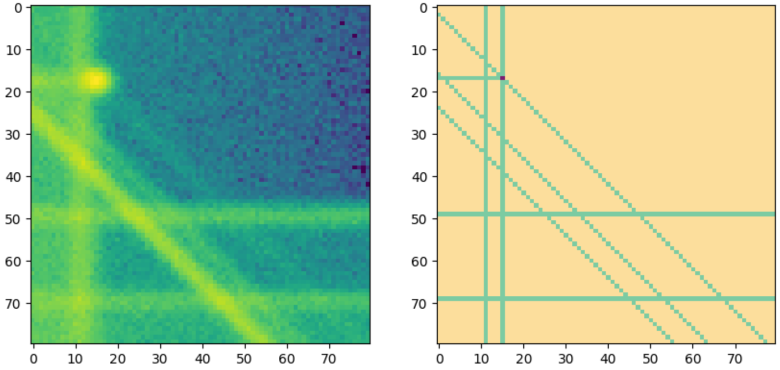
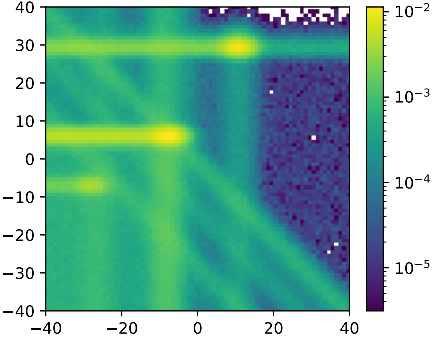
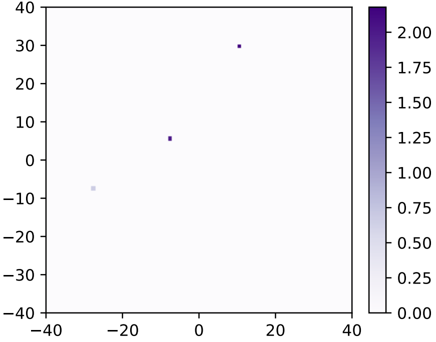


Figure: Example of a synthetic histogram with one peak with it corresponding mask (right)

The architecture : inputs and outputs



(a) Input : histogram with one or several peaks



(b) Expected outputs : peaks positions and intensities

Figure: Inputs and outputs of the neural network

The architecture : the network

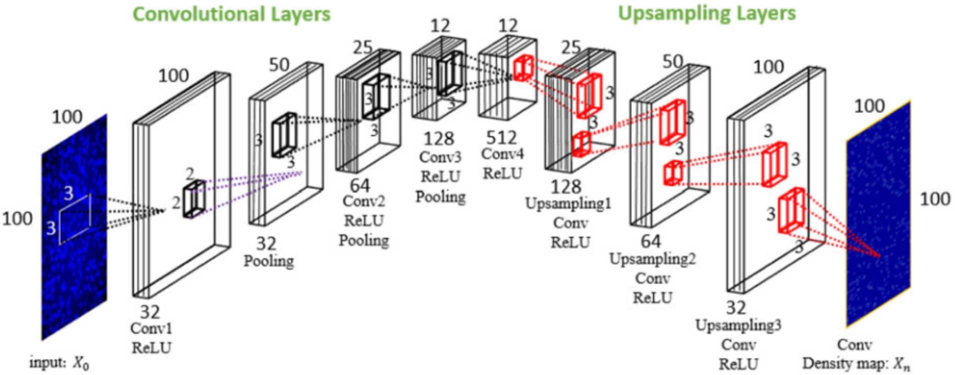


Figure: One of the implemented architecture. Inspired from Xie et al

Loss function

Two tasks : find the peaks locations **and** intensities

Loss function

Two tasks : find the peaks locations **and** intensities

Notations :

- $\hat{y} = (\hat{y}_{ij})$: output of the model
- $y = (y_{ij})$: the expected output
- $\mathcal{I} = \{(i,\, j) \,|\, y_{ij} > 0\}$
- $\hat{\mathcal{I}} = \{(i,\, j) \,|\, \hat{y}_{ij} > 0\}$

Loss function

Two tasks : find the peaks locations **and** intensities

Notations :

- $\hat{y} = (\hat{y}_{ij})$: output of the model
- $y = (y_{ij})$: the expected output
- $\mathcal{I} = \{(i,\, j) \,|\, y_{ij} > 0\}$
- $\hat{\mathcal{I}} = \{(i,\, j) \,|\, \hat{y}_{ij} > 0\}$

And $\mathcal{F} = \hat{\mathcal{I}} \setminus \mathcal{I}$ the indexes of false positives

Loss function

Two tasks : find the peaks locations **and** intensities

Notations :

- $\hat{y} = (\hat{y}_{ij})$: output of the model
- $y = (y_{ij})$: the expected output
- $\mathcal{I} = \{(i, j) \mid y_{ij} > 0\}$
- $\hat{\mathcal{I}} = \{(i, j) \mid \hat{y}_{ij} > 0\}$

And $\mathcal{F} = \hat{\mathcal{I}} \setminus \mathcal{I}$ the indexes of false positives

$$\boxed{\mathcal{L}(y, \hat{y}, \boldsymbol{\theta}) = \mathcal{L}_{\mathcal{I}}(y, \hat{y}, \boldsymbol{\theta}) + \lambda \cdot \mathcal{L}_{\mathcal{F}}(y, \hat{y}, \boldsymbol{\theta})} \tag{1}$$

$$\mathcal{L}_{\mathcal{I}}(y, \hat{y}, \boldsymbol{\theta}) = \sum_{\mathcal{I}} (\hat{y}_{ij} - y_{ij})^2 \qquad \mathcal{L}_{\mathcal{F}}(y, \hat{y}, \boldsymbol{\theta}) = \mathrm{card}(\mathcal{F})$$

Details

- input "augmented" with prior information on the peaks positions
- 2 architectures implemented using PyTorch
- models train on GPU on the CCIN2P3

First results

What metrics to measure the accuracy of the neural network ?

- Relative error : $\frac{y-\hat{y}}{y}$ where y is the expected peak intensity
- number of false positive

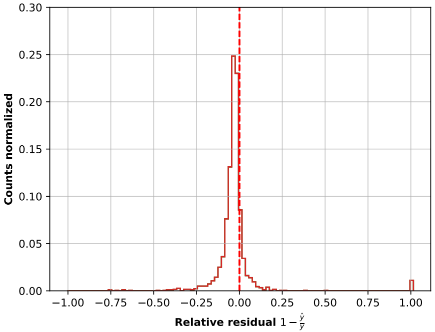


Figure: Distribution of the relative error for the test set

Question : what did we miss with this model ?

Question : what did we miss with this model ?

Uncertainties in the inputs

statistical variation of the content of each bin in the 2d spectra

Question : what did we miss with this model ?

Uncertainties in the inputs

statistical variation of the content of each bin in the 2d spectra

Uncertainties in the outputs

statistical variation of the number of gamma produced and detected

Question : what did we miss with this model ?

Uncertainties in the inputs

statistical variation of the content of each bin in the 2d spectra

Uncertainties in the outputs

statistical variation of the number of gamma produced and detected

Confidence of the model

uncertainties in the prediction of the neural network

Conclusion

What has be done so far ?

- FIPPS raw data treatment and determination of the efficiency of the FIPPS spectrometer with a relative uncertainty around 1%
- building of a generator of synthetic data to train and test machine learning models
- implementation, training and testing of two architectures of CNN

What's next ?

- uncertainty quantification
- test the robustness of the neural network
- test the neural network on real data and compare with classic fit method
- analysis of the FIPPS data

Table of Contents

Context and objective

Experimental setup and previous analysis

Building a new analysis tool with machine learning

Extras

^{136}Te levels

Objective

We want to measure the independent fission yield of tellurium 136.

How ?

by counting the number of gamma rays emitted by a ^{136}Te nucleus and measured by the spectrometer.

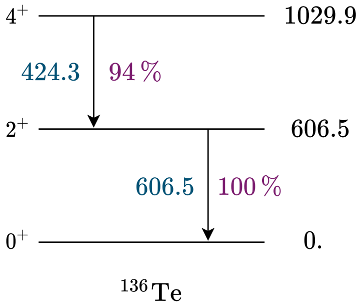


Figure: Energy and transition for the first two levels of ^{136}Te

Aperçu de spectre 1d

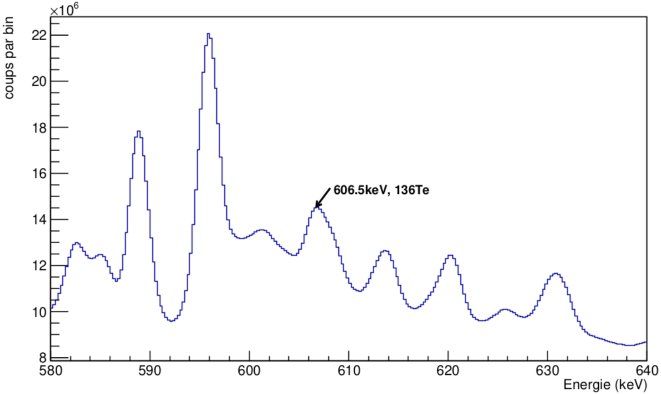


Figure: Zoom around 610 keV on the one full fission events spectrum from FIPPS

Coincidence matrix

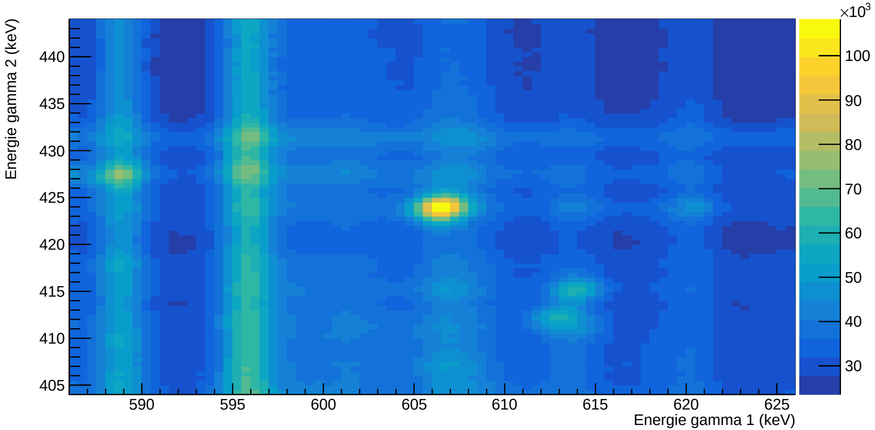
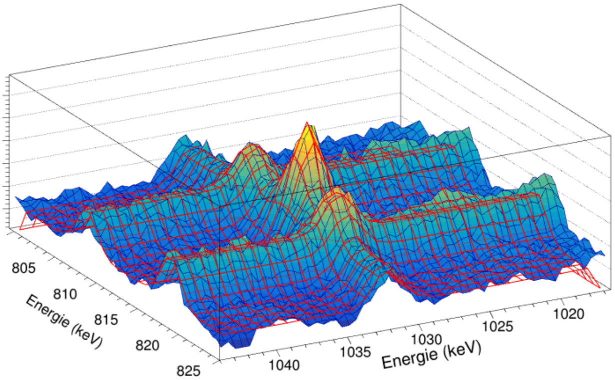


Figure: Region around the peak of interest on the coincidence matrix for fission events

Example of a fit of a peak on the matrix



- Required complex fit model
- Goodness-of-fit difficult to estimate

Projection and fit

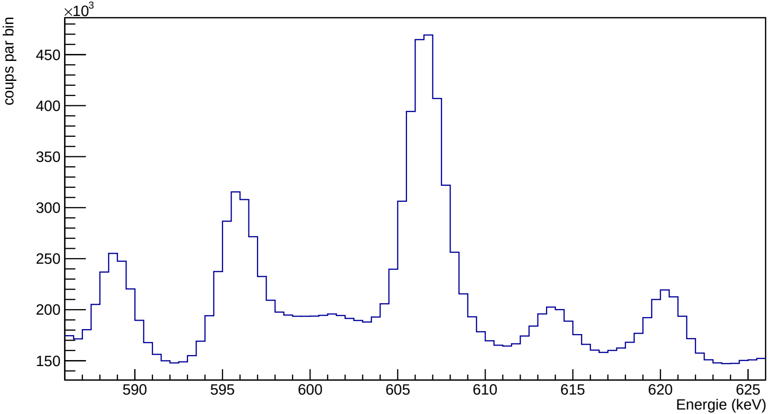


Figure: Projection of the previous histogram after applying a selection on the y-axis between 421 and 424 keV

What already exist

- Détection de pics : Kensert et al. 2022 (chromatographie)
- Ajustement de pics en 1d : Park et al. 2021 (photo émission), Abdel-Aal 2002
- ML et spectroscopie gamma : Kamuda et al. 2020, Daniel et al. 2020
- architectures : U-Net Ronneberger et al. 2015, SE (Squeeze and Excitation)

References I

R. E. Abdel-Aal. Comparison of Algorithmic and Machine Learning Approaches for the Automatic Fitting of Gaussian Peaks. *Neural Computing & Applications*, 11(1):17–29, June 2002. ISSN 0941-0643, 1433-3058. doi: 10.1007/s005210200012. URL http://link.springer.com/10.1007/s005210200012.

G. Daniel, F. Ceraudo, O. Limousin, D. Maier, and A. Meuris. Automatic and Real-Time Identification of Radionuclides in Gamma-Ray Spectra: A New Method Based on Convolutional Neural Network Trained With Synthetic Data Set. *IEEE Transactions on Nuclear Science*, 67(4):644–653, Apr. 2020. ISSN 0018-9499, 1558-1578. doi: 10.1109/TNS.2020.2969703. URL https://ieeexplore.ieee.org/document/8970571/.

M. Kamuda, J. Zhao, and K. Huff. A comparison of machine learning methods for automated gamma-ray spectroscopy. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 954:161385, Feb. 2020. ISSN 01689002. doi: 10.1016/j.nima.2018.10.063. URL https://linkinghub.elsevier.com/retrieve/pii/S0168900218313779.

References II

A. Kensert, E. Bosten, G. Collaerts, K. Efthymiadis, P. Van Broeck, G. Desmet, and D. Cabooter. Convolutional neural network for automated peak detection in reversed-phase liquid chromatography. *Journal of Chromatography A*, 1672:463005, June 2022. ISSN 00219673. doi: 10.1016/j.chroma.2022.463005. URL `https://linkinghub.elsevier.com/retrieve/pii/S0021967322002035`.

S.-H. Park, H. S. Park, H. Lee, and H.-S. Kim. Iterative peak-fitting of frequency-domain data via deep convolution neural networks. *Journal of the Korean Physical Society*, 79(12): 1199–1208, Dec. 2021. ISSN 0374-4884, 1976-8524. doi: 10.1007/s40042-021-00346-1. URL `http://arxiv.org/abs/2107.04287`. arXiv:2107.04287 [cond-mat].

O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28.