Generative models for transient noise studies in Gravitational Waves detectors

🎤**Lorenzo Asprea[1,2]**, 🎤**Francesco Sarandrea[1,2],** *Elia Cellini[3], Federica Legger[1,2], Sara Vallero[1,2]*

[1]On Behalf of the Virgo Collaboration, [2]INFN Turin, [3]University of Turin

EuCAIFCon - Amsterdam xx/xx/2024

Funded by the European Union

The interTwin project is funded by the European Union - Grant Agreement Number 101058386

Why

Gravitational interferometers (GI) are important because they allow scientists to detect and study **Gravitational Waves (GWs)**, which help us **understand** the **Universe**

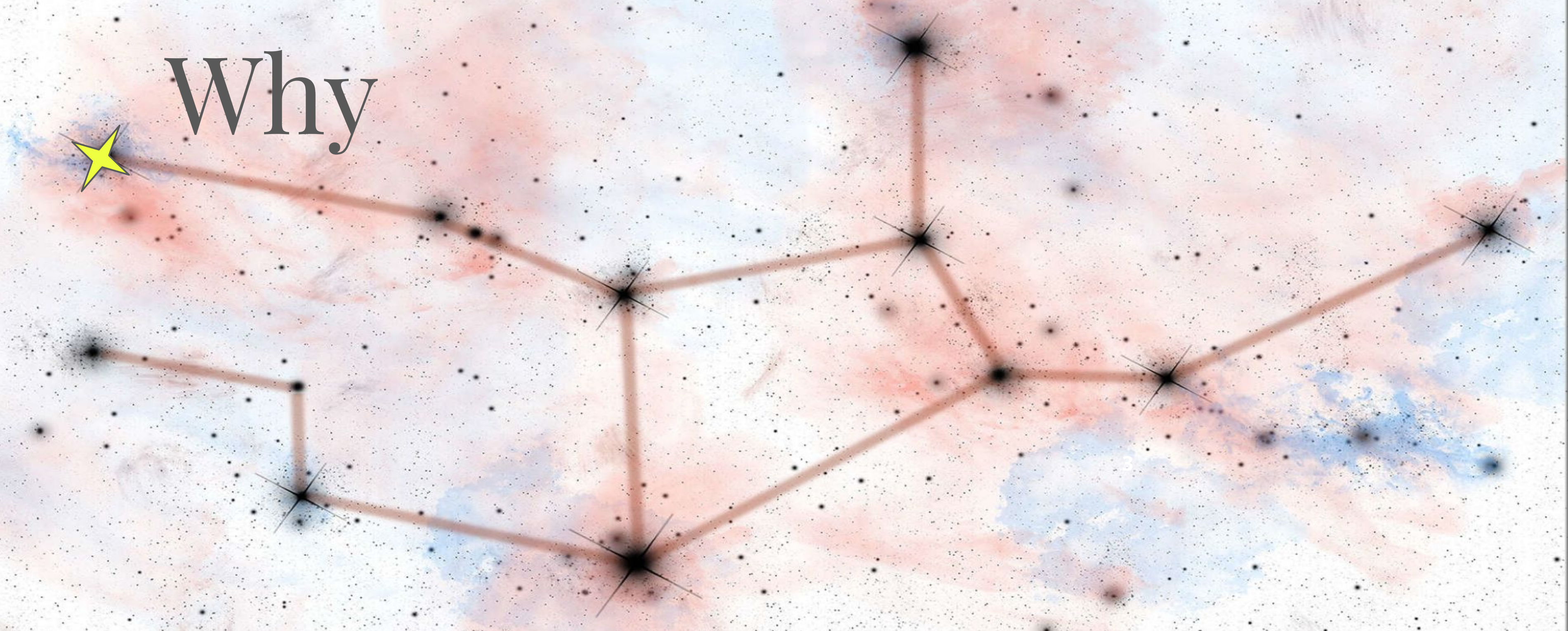
What

GI sensitivity is **limited by noise**. We want to **separate physical information from transient noise** in GI data.

How

Making use of **Generative Models** leveraging on deep NN’s ability to understand **non linear structure in data**.

Why



Gravitational Waves and Interferometers

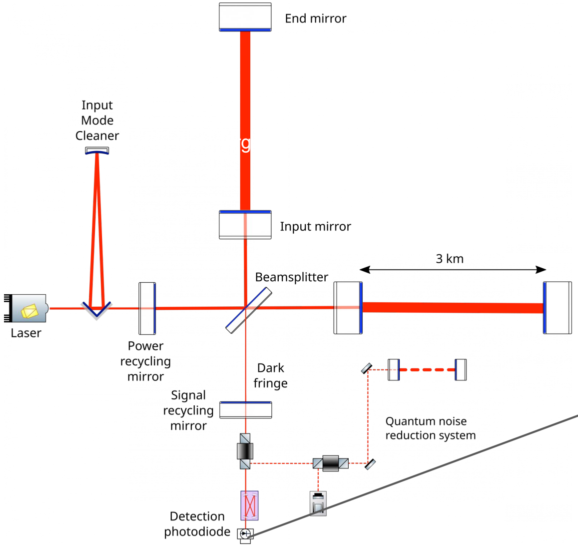
- **Gravitational interferometers** are important because they allow scientists to detect and study **Gravitational Waves (GWs)**.
- **GWs** are ripples in spacetime caused by the acceleration of massive objects, such as **black holes** or **neutron stars**.
- GWs were predicted by **Albert Einstein**'s theory of **General Relativity** but were not directly observed until 2015 with the first detection by the **LIGO – Virgo** Collaboration. (https://doi.org/10.1103/PhysRevLett.116.061102)

GWs are important for:

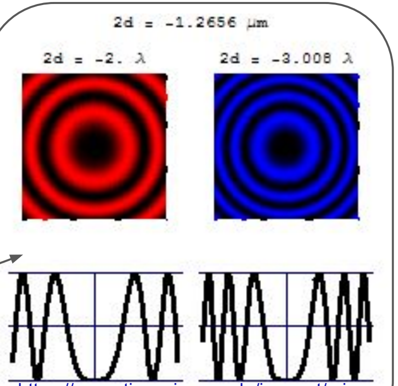
- **Cosmology and the Early Universe**
- **Fundamental Physics**
- **Multi-Messenger Astronomy**

Gravitational Interferometers: very complex machines!

- The **Main Channel** measures the **interference pattern produced by laser** light as it **travels along** the **GI**'s **arms**
- When a **GW** passes by, it slightly **changes the length** of the **arms**, causing the **interference pattern to change**



https://www.virgo-gw.eu/science/detector/optical-layout/

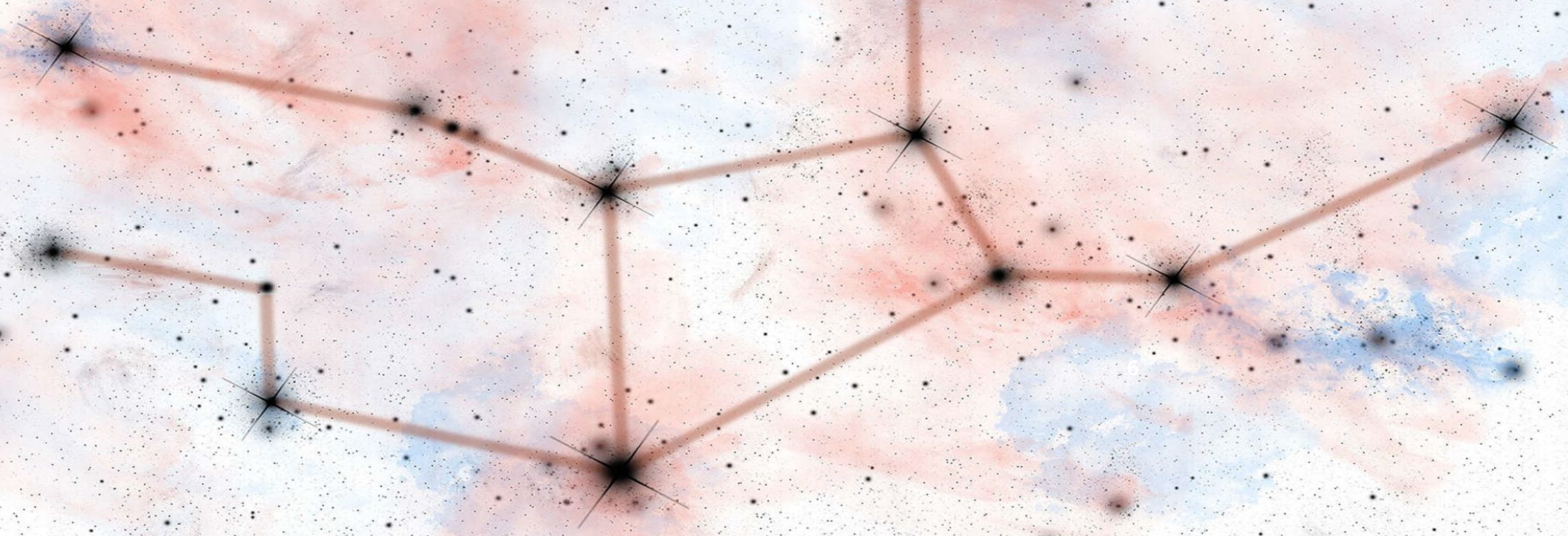


https://wp.optics.arizona.edu/jcwyant/miscellaneous/neat-graphics/michelson-interferometer-interference-fringes/

- **Extreme Precision (~10^{-21}) & Large Scale (~3 Km)**
- **Isolation** from **Noise**, both **transient** and **stationary** from:
  - **Environment** (wind, seismic, weather, sea,**..)**
  - **Instrumentation** (power, resonating frequencies,..**)**
  - **Physics** (vacuum, thermal noise, quantum effects,..**)**
  - **Crazy sources** (planes in the sky, you sneezing,..**)**
- ...

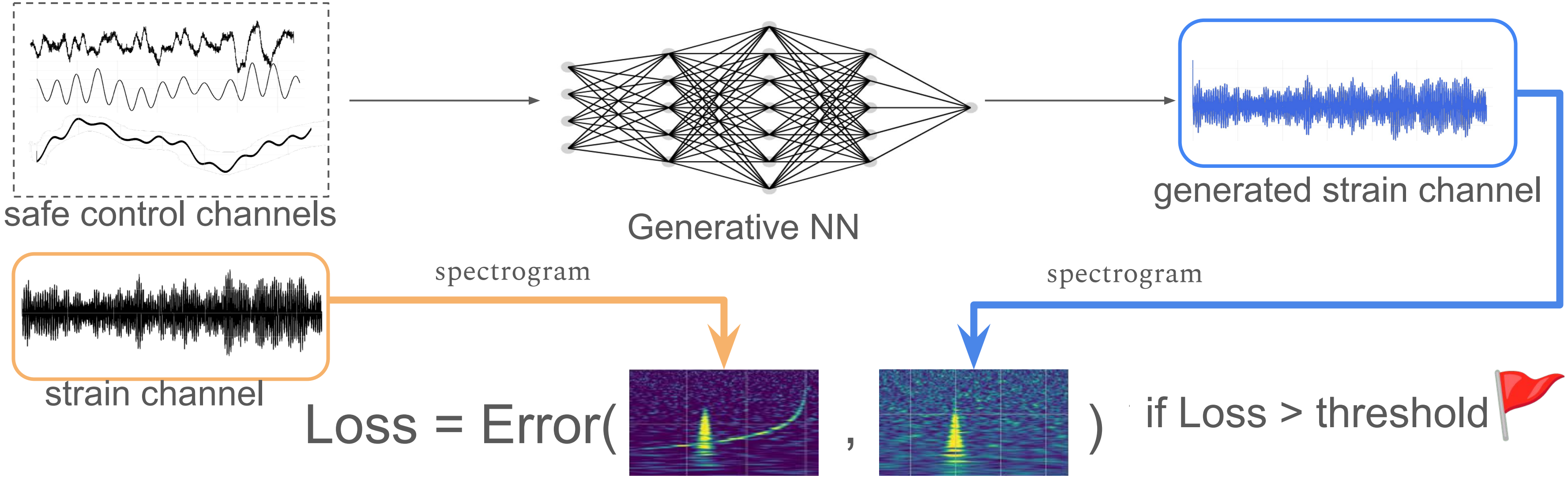
~100K Control Channels to account for **noise**.
Despite such corrections, **data is still buried under it**!

What



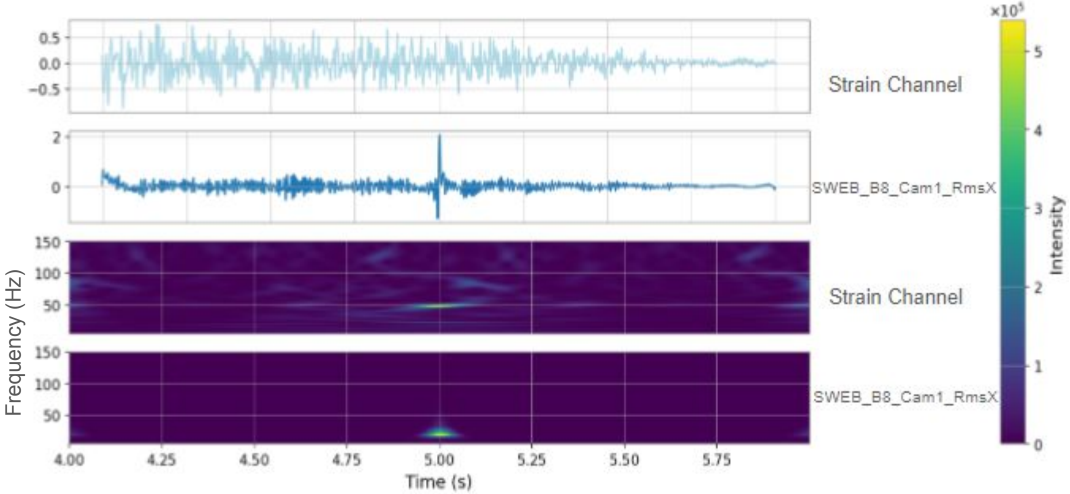
Outline and goal

- Non-Gaussian transient noise artifacts (aka **glitches**) are one of the most **challenging limitations** in the study of gravitational-wave interferometer
- We **map glitches from control channels** (uncorrelated with the physical signals) **to the main channel**, in order to **subtract the generated noise from the physically interesting data**
- We use **deep generative models** for the mapping in order to capture **non-linear** structure in the **data**

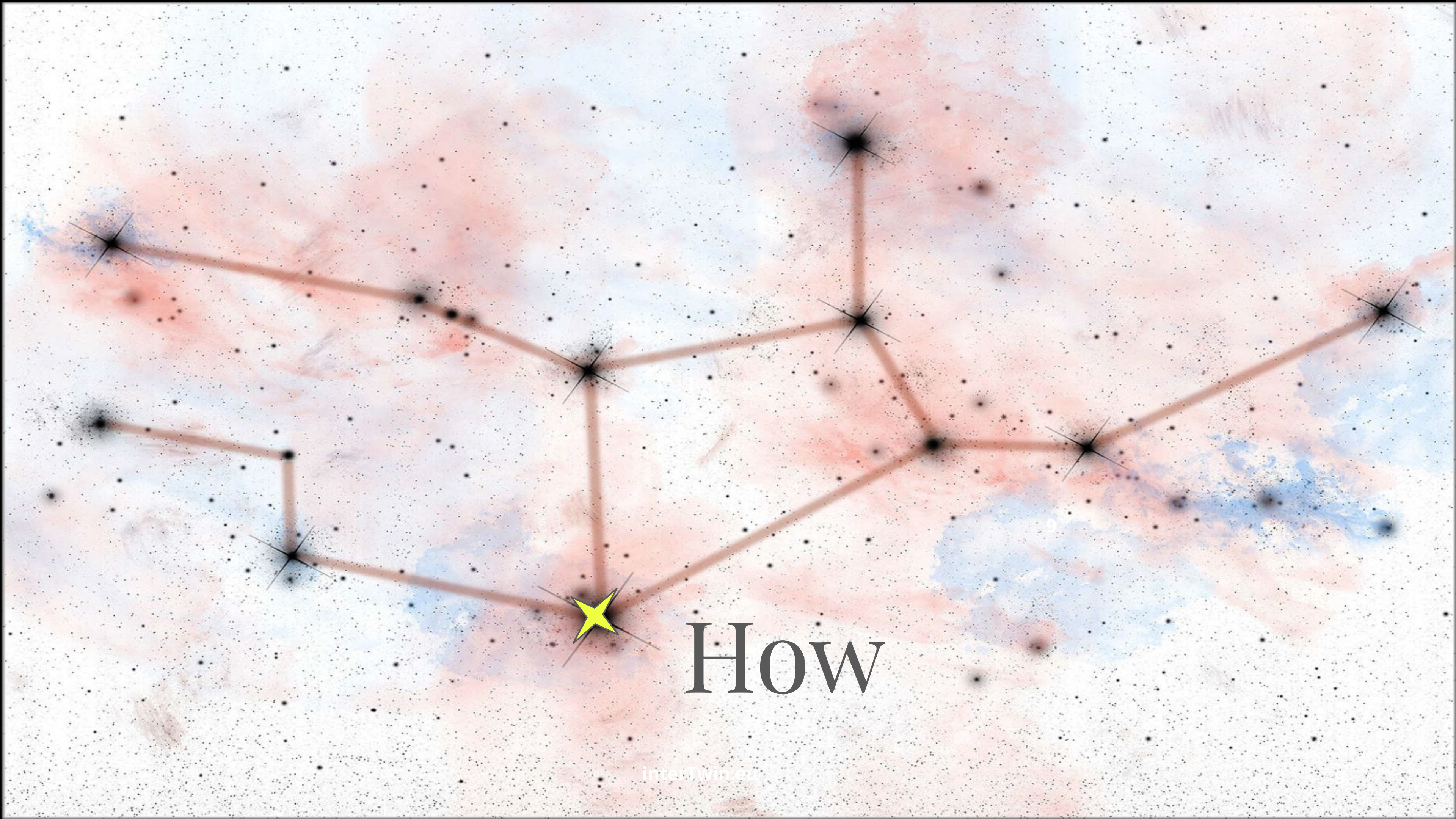


Gravitational Interferometers' data

- Single type of glitch: **Scattered Light** (unwanted light that is reflected or scattered off surfaces within the interferometer's optical components).
- **Physical motivation** in **channel selection** (Out or the 100K+ auxiliary channels, we select those whose units of measurement are position, velocity and acceleration)
- Consider only **safe channels** (i.e. channels that are not witness of any gravitational signal)
- Choose only **correlated channels** (i.e. control channels that show excess in energy at the same time as the main channel.
- Use **Qplots** (a particular type of spectrograms, that show the energy content of signals in both time and frequency domain)
- **Correlation** among channels can be **non-linear** (Synchronous excess in energy can occur at different frequencies)



Time series (top) and Qplots (bottom) for main channel and one channel monitoring movements of the optical benches



How

Dataset

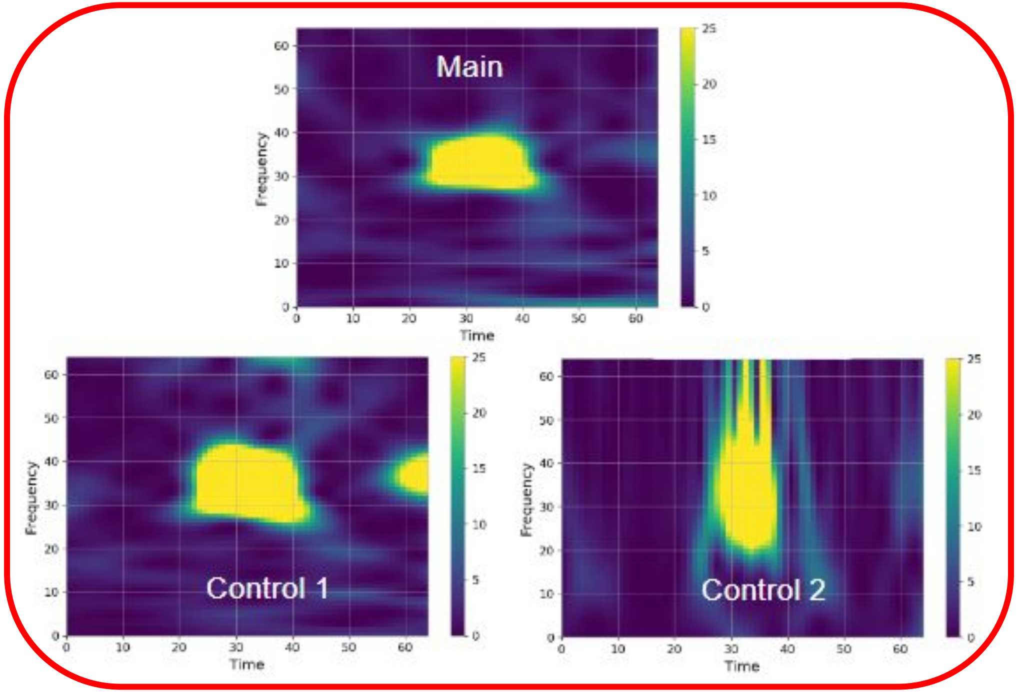
Time series

1. 12K Scattered Light events in O3a VIRGO
2. ± 8s around glitch
3. Resampling to 500Hz
4. Whitening
5. Normalization to [0,1] range

Qplots

6. Qplot and cropping around highest peak frequency.
7. Select spectrograms with SNR of at least 15
8. Normalization to [0,1] range

64x64 = (1s)x(64Hz)

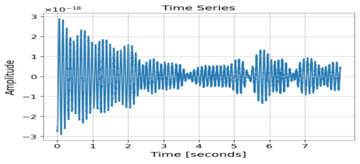


Neural Network architectures

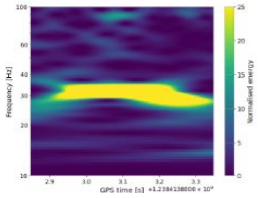
- **Code in Pytorch**
- **L1-Loss= Σ |Generated Output - Target Output|**

- **Input**

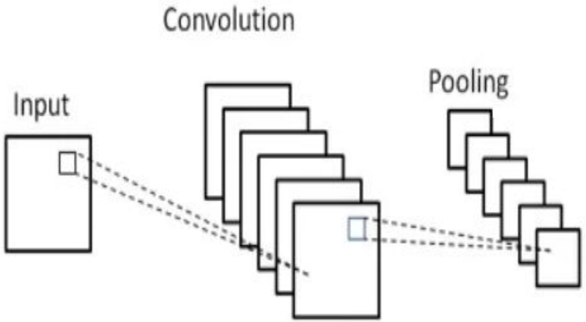
1D



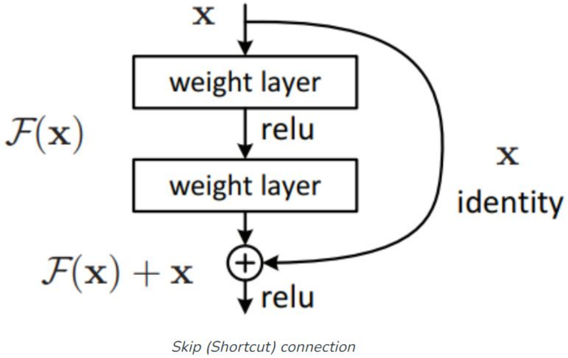
2D



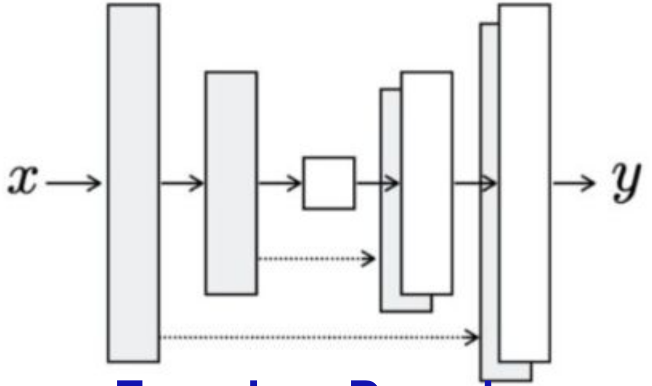
CNN



ResNet (residual block)

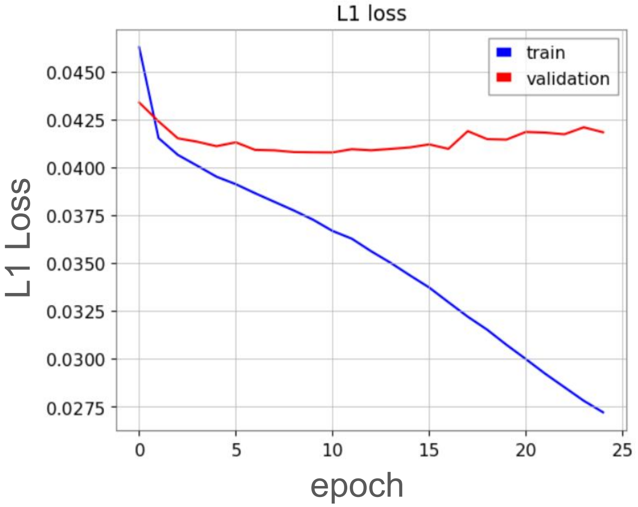


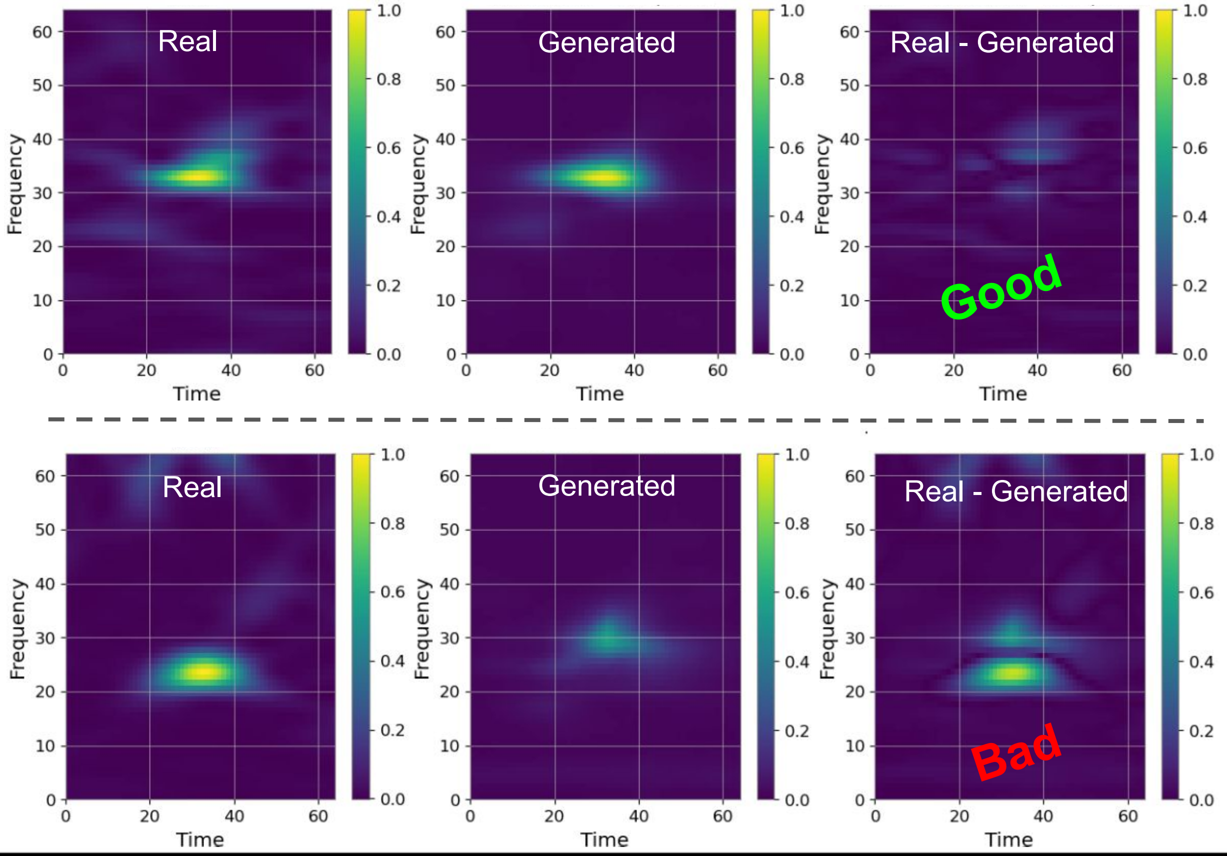
U-Net



Encoder - Decoder

QPlots: Resnet (12 blocks)





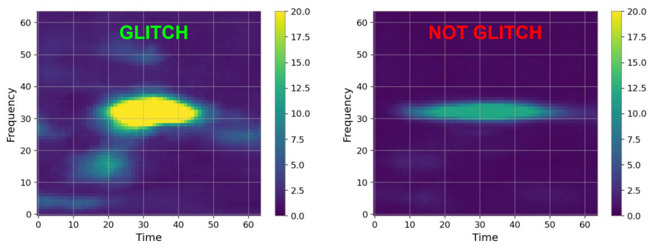
- **Mostly very good results**, but some bad ones as well in presence of messy or odd looking input
- Network seems to stop learning after very few epochs. **High Bias after very few epochs**

Performance Tests: Vetoing

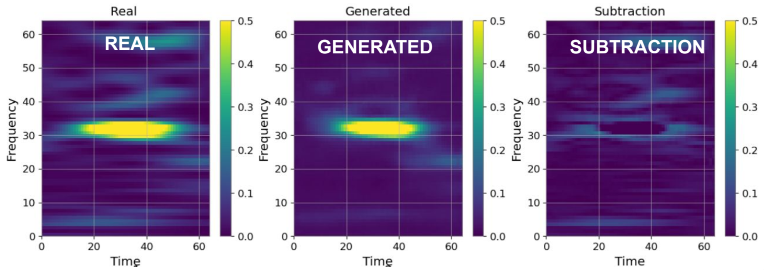
- ***Glitch definition***: **Cluster** of at least **10 pixels** with **SNR above threshold (15,10,8)**. This choice mimics actual alert mechanisms in use at Virgo (Omicron)
- Use **Clustering** mechanism as **Classifier** on **generated data**
- **Test set**: 1083 Glitches, 536 empty background

	SNR 8	SNR 10	SNR 15
Glitch generation Accuracy Model is able to correctly generate a glitch given control channels	99.0%	98.7%	97.8%
Glitch position Accuracy Model generates glitches with right time and frequency. Intensity is saturated at trigger SNR	25.9%	99.0%	99.6%

Glitch generation



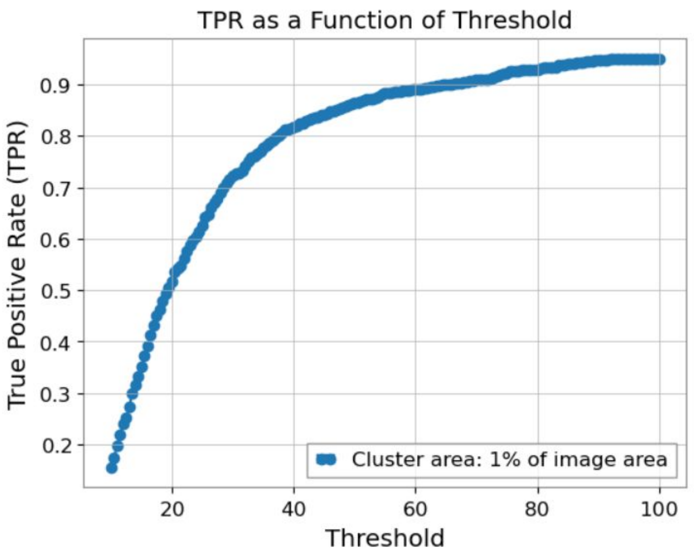
Glitch position



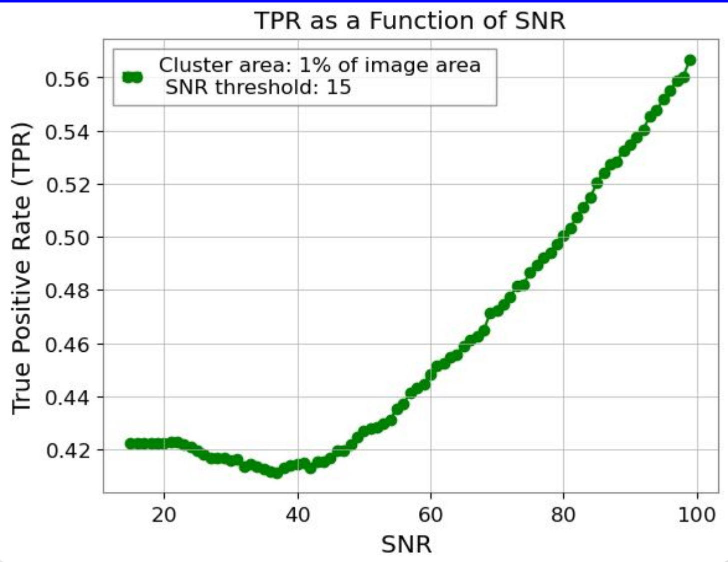
Performance Tests: Denoising

- ***Glitch definition***: **Cluster** of at least **41 pixels** (1% of total Qplot area) with **SNR above threshold**.
- Use **Clustering** mechanism as **Classifier** on **cleaned data** (i.e. real minus generated)
- **Test set**: 1083 Glitches, 536 empty background

Dataset fixed, varying Classifier Threshold



Classifier Threshold fixed, varying SNR in Dataset



Future Work

- Streamline data acquisition and transfer to DataLake
- Perform channel analysis on bigger dataset for all channels
- Refine and implement data analysis protocol into shareable tool
- Retrain networks with new and more channels
- Deploy more complex generative models like GANs, Pix2Pix,...
- Generate synthetic GWs signals
- Design denoising pipeline
- Explore ideas from audio processing (like MFCCs, and so on)
- … **Suggestions?**

Thank you! Questions?

www.intertwin.eu

info@intertwin.eu

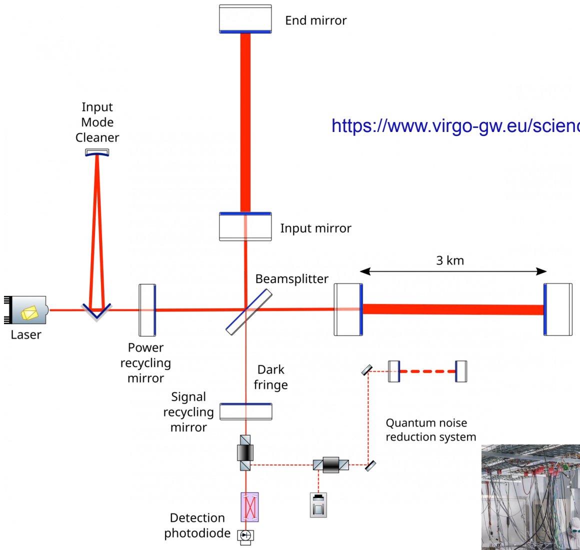
intertwin_eu

intertwin

Backup slides

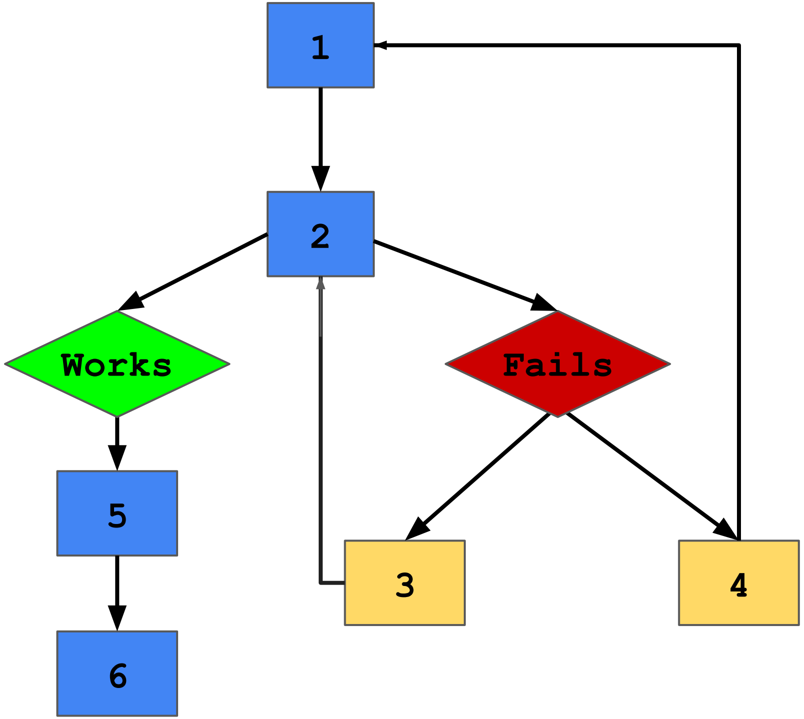
Gravitational Interferometers: very complex machines!

- **Extreme Precision**: They require incredibly precise measurement capabilities to detect tiny distortions in spacetime caused by gravitational waves.
- **Large Scale**: Interferometers such as LIGO and Virgo consist of multiple components, including long, L-shaped vacuum tubes (several kilometers in length) with mirrors suspended at their ends.
- **Isolation from Noise**: They need to isolate their sensitive instruments from various sources of noise, such as seismic activity, thermal fluctuations, and even quantum mechanical effects.
- **High Vacuum Environment**: To minimize interference from air molecules, interferometers operate in high vacuum environments
- **Advanced Laser Technology**: Interferometers rely on powerful lasers and intricate optical systems to precisely measure changes in the length of their arms.



Data Processing and Analysis: To keep track of every potential source of noise, Gravitational Interferometers have a very large [O(10^5)] number of control channels. Despite that, GWs signals are buried under a vast amount of noise.

Workflow



1. Identify *safe* auxiliary channels correlated with glitches in main channel ✓
2. Generate glitches from selected channels with a simple NN as a p.o.c. ✓
3. Increase complexity in NN architecture
4. Improve channel selection & Data preprocessing

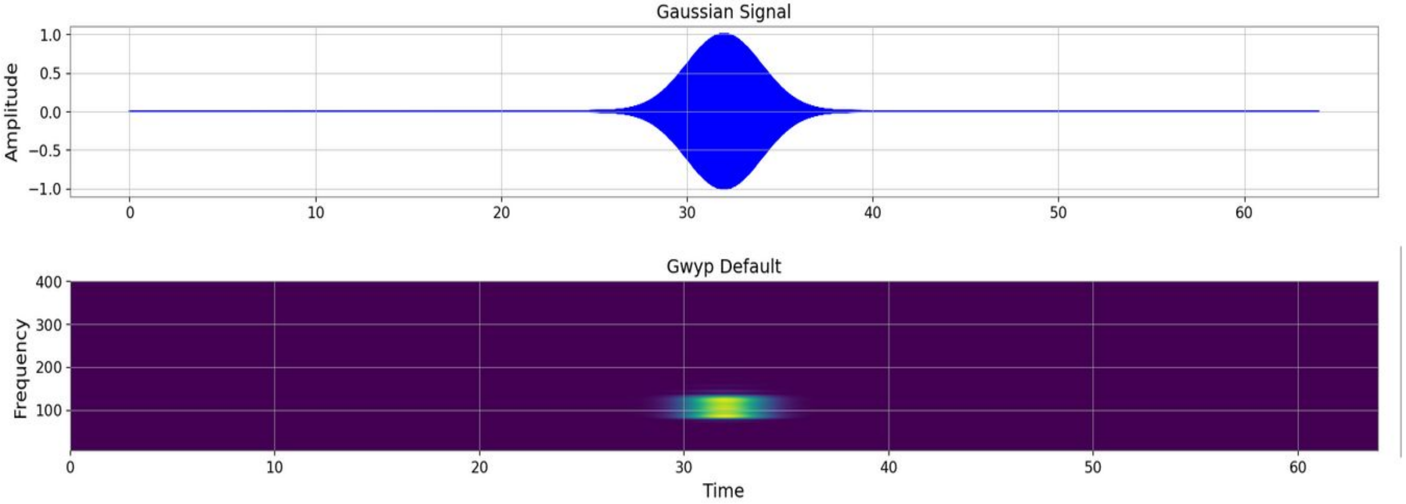
LOADING...

5. Generate synthetic GWs and test the glitch subtraction procedure
6. Test on real GW data

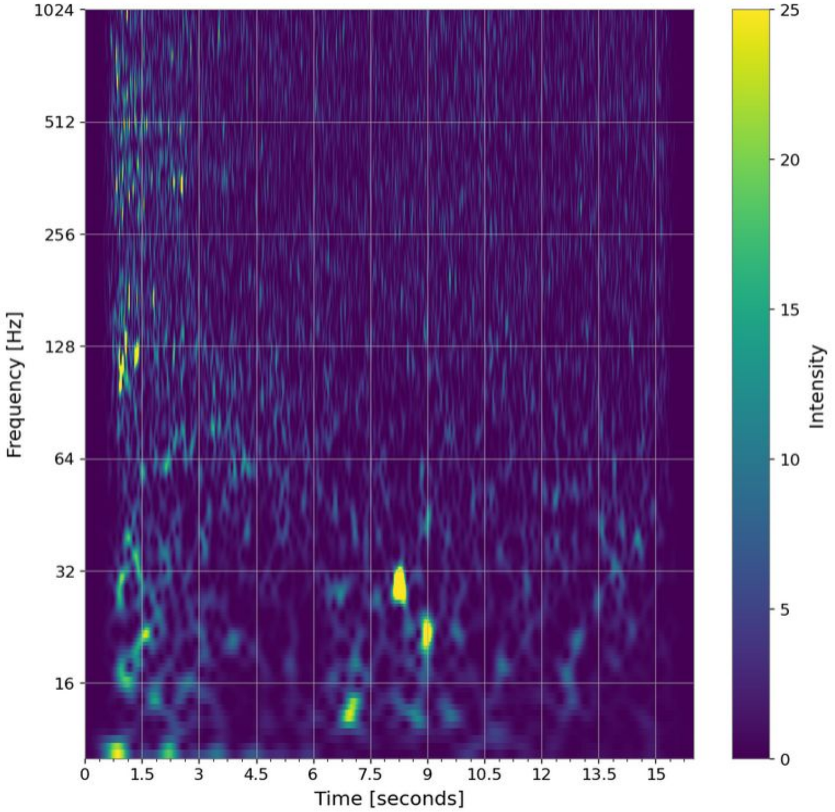
TO DO

What are Spectrograms?

Spectrograms are visual **representations** of the **spectrum** of **frequencies** of a **signal** as it **varies with time**



Q-Transforms



$$X(\tau,\phi,Q)=\int_{-\infty}^{+\infty}x(t)w(t-\tau,\phi,Q)e^{-i2\pi\phi t}dt$$

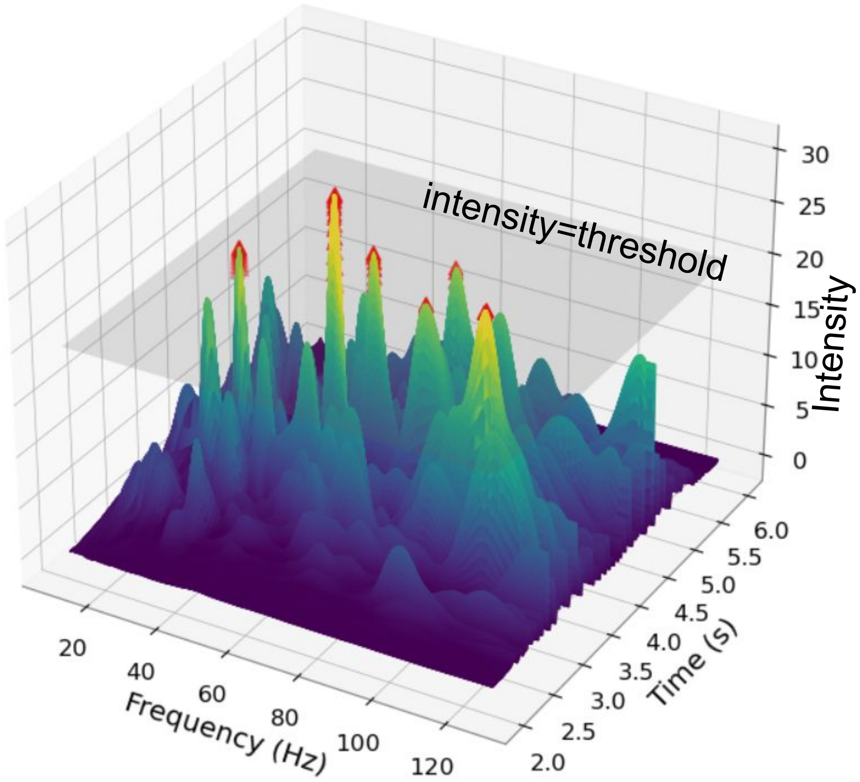
$$w(t-\tau,\phi,Q)=\frac{C}{\sqrt{2\pi}\sigma_t(Q,\phi)}\exp\left[-\frac{(t-\tau)^2}{2\sigma_t(Q,\phi)^2}\right]$$

$$\sigma_t(Q,f)^2=\frac{Q^2}{8\pi^2f^2}$$

In practice, few lines of code thanks to gwpy!

```
from gwpy.timeseries import TimeSeries

filename='/data/datasets/intertwin-dati-aux/gspy_O3a_Scattered_Light_V1/DdncGnx3ph.h5'

fout=h5.File(filename)

t_strain=TimeSeries(fout['DdncGnx3ph.h5']['V1:Hrec_hoft_16384Hz'])

t_strain=t_strain.whiten()

hq=t_strain.q_transform(frange=f_range)
```

Correlations in Q-Plots: Peak frequencies

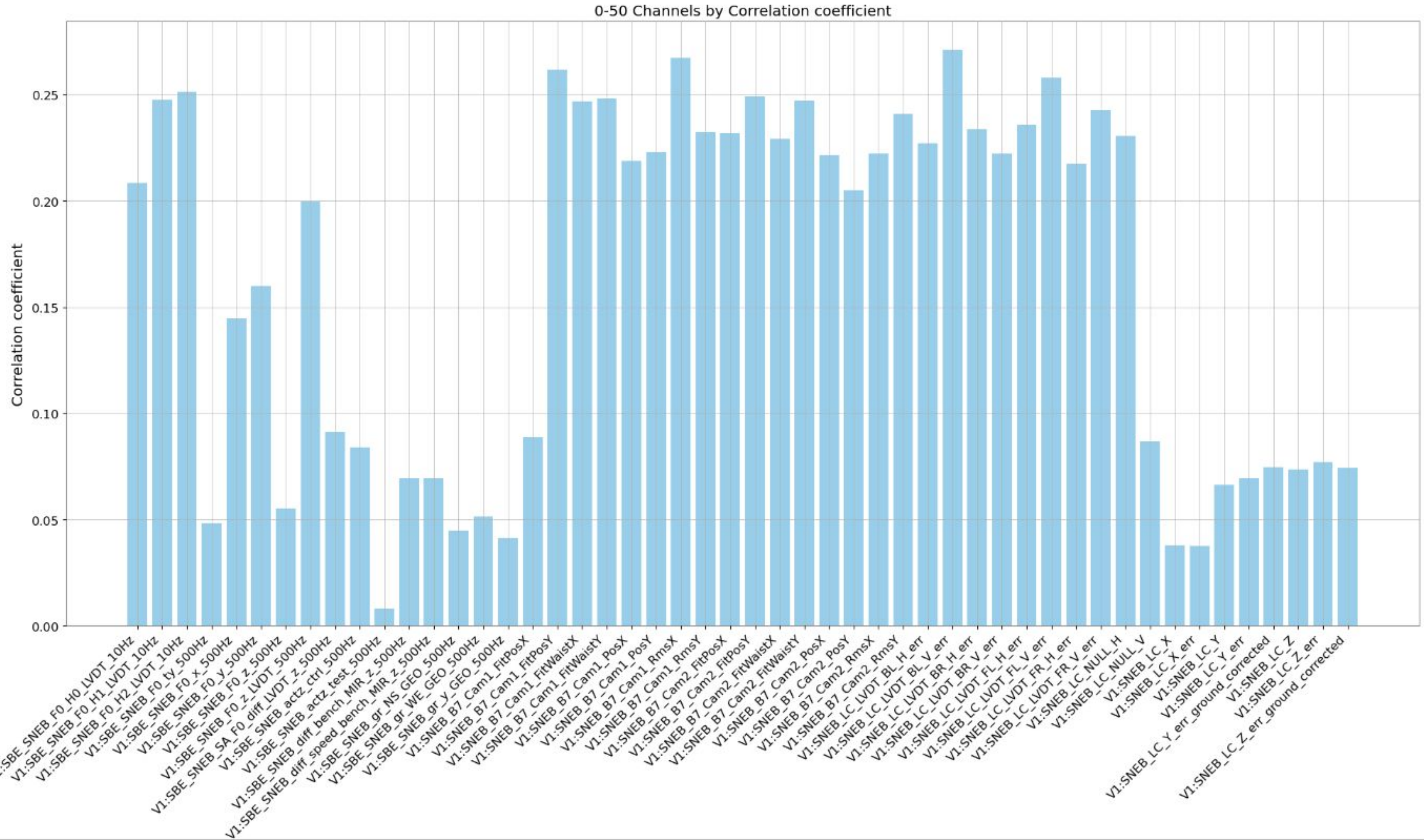


- ***Common peaks***: peaks above intensity threshold within a given tolerance distance (Δt,Δf) in Q-Plot
- Apply analysis to each aux channel paired with strain channel
- Calculate correlation coefficient:

$$\text{Corr\_coeff} = \frac{\textbf{common peaks}}{\textbf{strain peaks}}$$

- Average over 1000 different “events” classified as Scattered Light with confidence ≧86%.
- Data source: VIRGO O3a
- Python: gwpy, lalframe, pandas

1



0-50 Channels by Correlation coefficient
Correlation coefficient
0.00
0.05
0.10
0.15
0.20
0.25
V1:SBE_SNEB_F0_H0_LVDT_10Hz
V1:SBE_SNEB_F0_H1_LVDT_10Hz
V1:SBE_SNEB_F0_H2_LVDT_10Hz
V1:SBE_SNEB_F0_ty_500Hz
V1:SBE_SNEB_F0_x_500Hz
V1:SBE_SNEB_F0_y_500Hz
V1:SBE_SNEB_F0_z_500Hz
V1:SBE_SNEB_F0_z_LVDT_500Hz
V1:SBE_SNEB_SA_F0_diff_LVDT_z_500Hz
V1:SBE_SNEB_actz_ctrl_500Hz
V1:SBE_SNEB_actz_test_500Hz
V1:SBE_SNEB_diff_bench_MIR_z_500Hz
V1:SBE_SNEB_diff_speed_bench_MIR_z_500Hz
V1:SBE_SNEB_gr_NS_GEO_500Hz
V1:SBE_SNEB_gr_WE_GEO_500Hz
V1:SBE_SNEB_gr_y_GEO_500Hz
V1:SNEB_B7_Cam1_FitPosX
V1:SNEB_B7_Cam1_FitPosY
V1:SNEB_B7_Cam1_FitWaistX
V1:SNEB_B7_Cam1_FitWaistY
V1:SNEB_B7_Cam1_PosX
V1:SNEB_B7_Cam1_PosY
V1:SNEB_B7_Cam1_RmsX
V1:SNEB_B7_Cam1_RmsY
V1:SNEB_B7_Cam2_FitPosX
V1:SNEB_B7_Cam2_FitPosY
V1:SNEB_B7_Cam2_FitWaistX
V1:SNEB_B7_Cam2_FitWaistY
V1:SNEB_B7_Cam2_PosX
V1:SNEB_B7_Cam2_PosY
V1:SNEB_B7_Cam2_RmsX
V1:SNEB_B7_Cam2_RmsY
V1:SNEB_LC_LVDT_BL_H_err
V1:SNEB_LC_LVDT_BL_V_err
V1:SNEB_LC_LVDT_BR_H_err
V1:SNEB_LC_LVDT_BR_V_err
V1:SNEB_LC_LVDT_FL_H_err
V1:SNEB_LC_LVDT_FL_V_err
V1:SNEB_LC_LVDT_FR_H_err
V1:SNEB_LC_LVDT_FR_V_err
V1:SNEB_LC_NULL_H
V1:SNEB_LC_NULL_V
V1:SNEB_LC_X
V1:SNEB_LC_X_err
V1:SNEB_LC_Y
V1:SNEB_LC_Y_err
V1:SNEB_LC_Y_err_ground_corrected
V1:SNEB_LC_Z
V1:SNEB_LC_Z_err
V1:SNEB_LC_Z_err_ground_corrected

Dataset

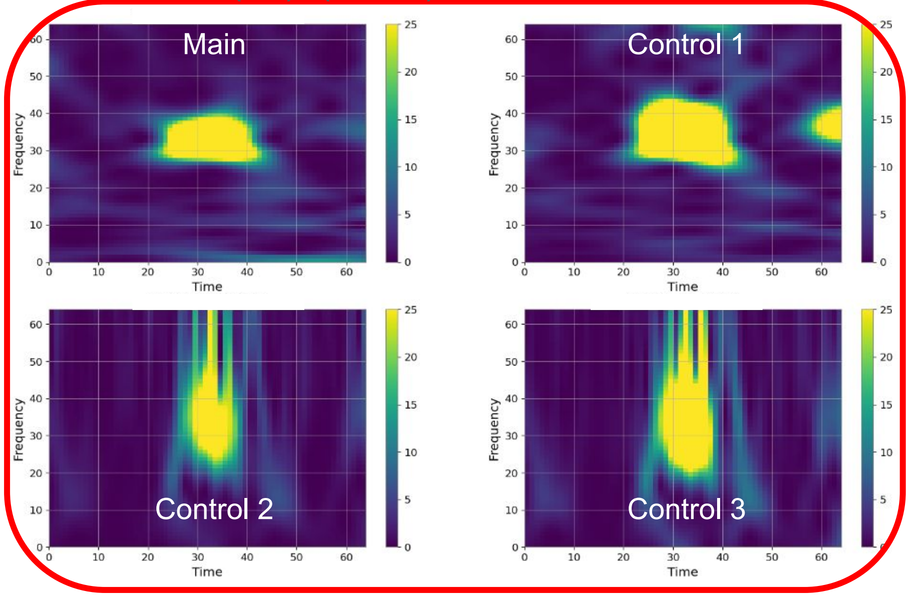
Time series

1. 12K Scattered Light events in O3a VIRGO
2. ± 3s around glitch
3. Resampling to 500Hz and whitening
4. Normalization to [0,1] range

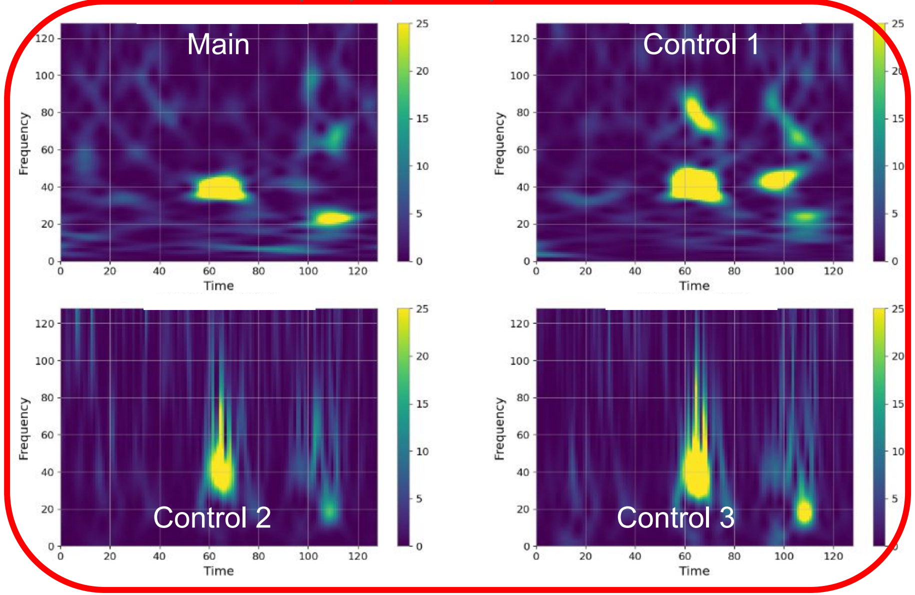
Qplots

1. ⟶ 4.
5. Qplot and cropping around highest peak frequency
6. Normalization to [0,1] range

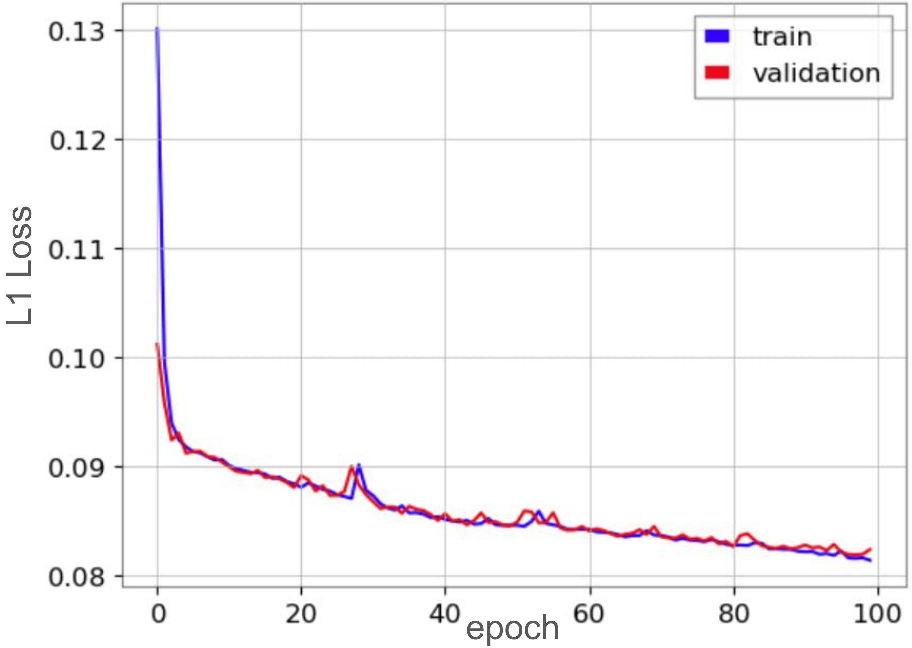
64x64 ~ (1s)x(35Hz)

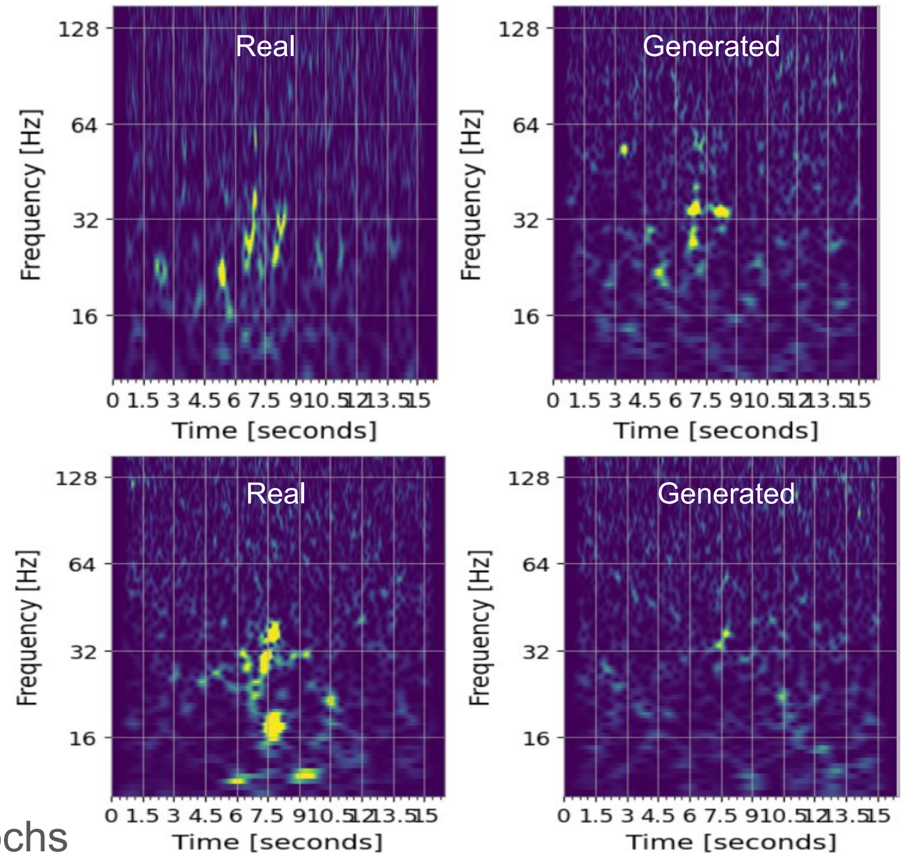


128x128 ~ (2s)x(70Hz)



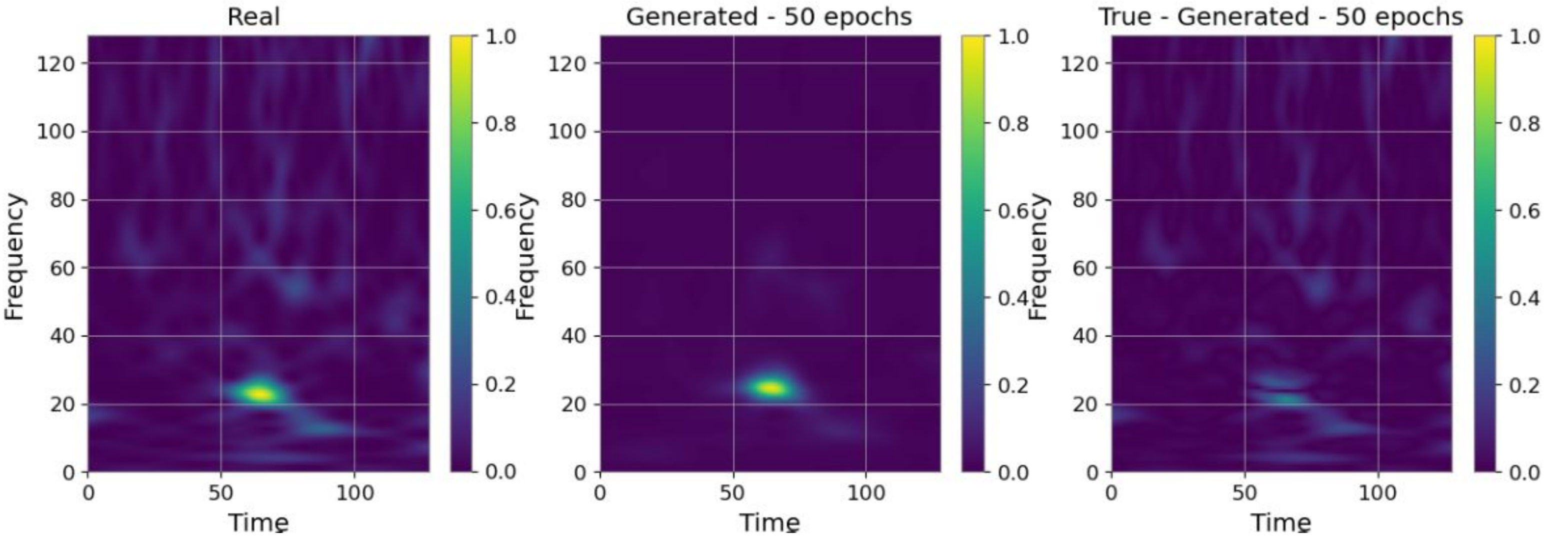
Time series: U-Net





- Some good results, but a lot of bad ones…
- Network seems to stop learning after few epochs

QPlots: Decoder CNN+ Resnet

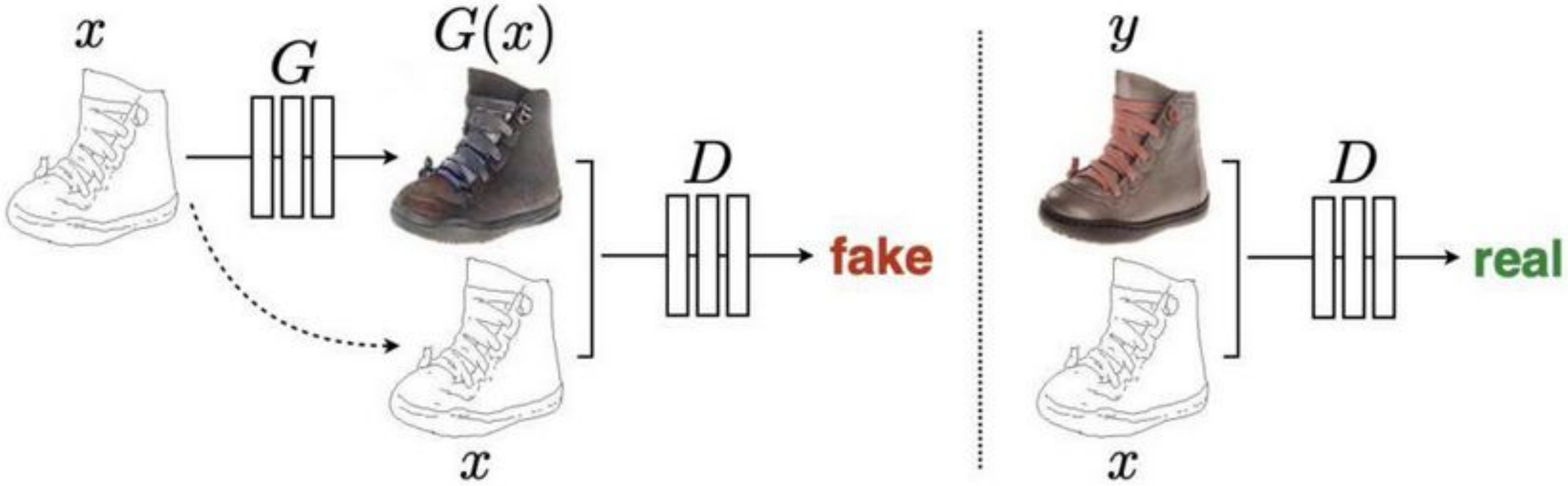


- Using 64x64 model as pre-training, only one epoch of training is required!
- High Bias after very few epochs
- Network does not seem to notice light blue/green background (good for noise stability?)
- Bad results for messy (or odd looking) input
- Very similar loss and results for U-Net

Pix2Pix GAN

Paired Image-to-Image Gan maps the data from three auxiliary channels to the data in the strain.

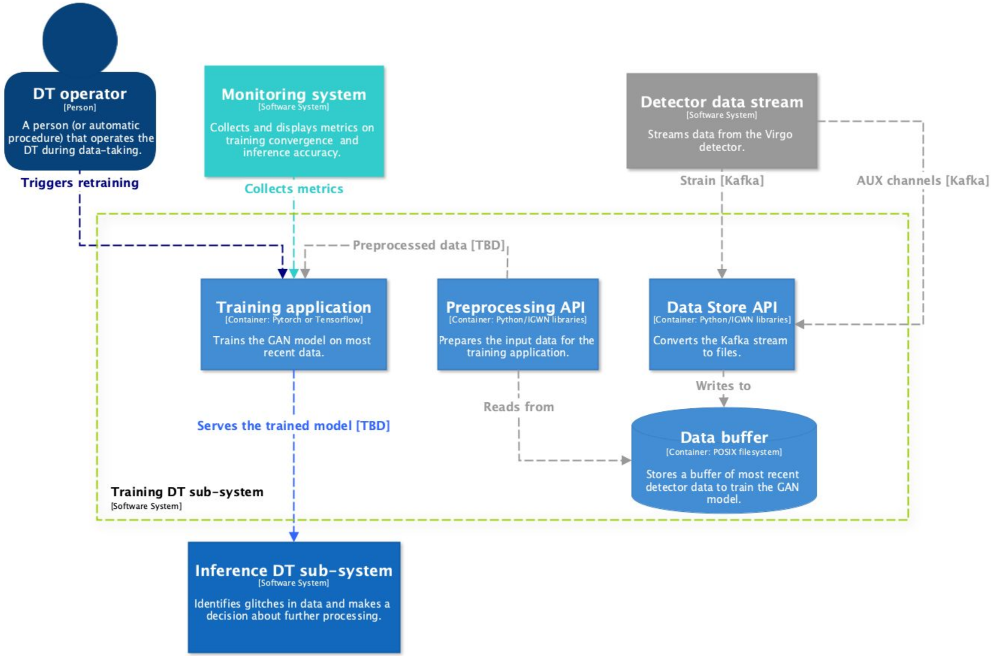
This is a conditional adversarial network which learns the mapping from input to output images.



Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

Status of the WP7 activities

- Internal architecture of the training subsystem defined
- Started first implementation of the training pipeline focusing on:
    - workflow execution (Airflow)
    - interface definition
    - data transfer (Rucio/Kafka)



Activities on the K8s demo cluster at CNAF (WP5)

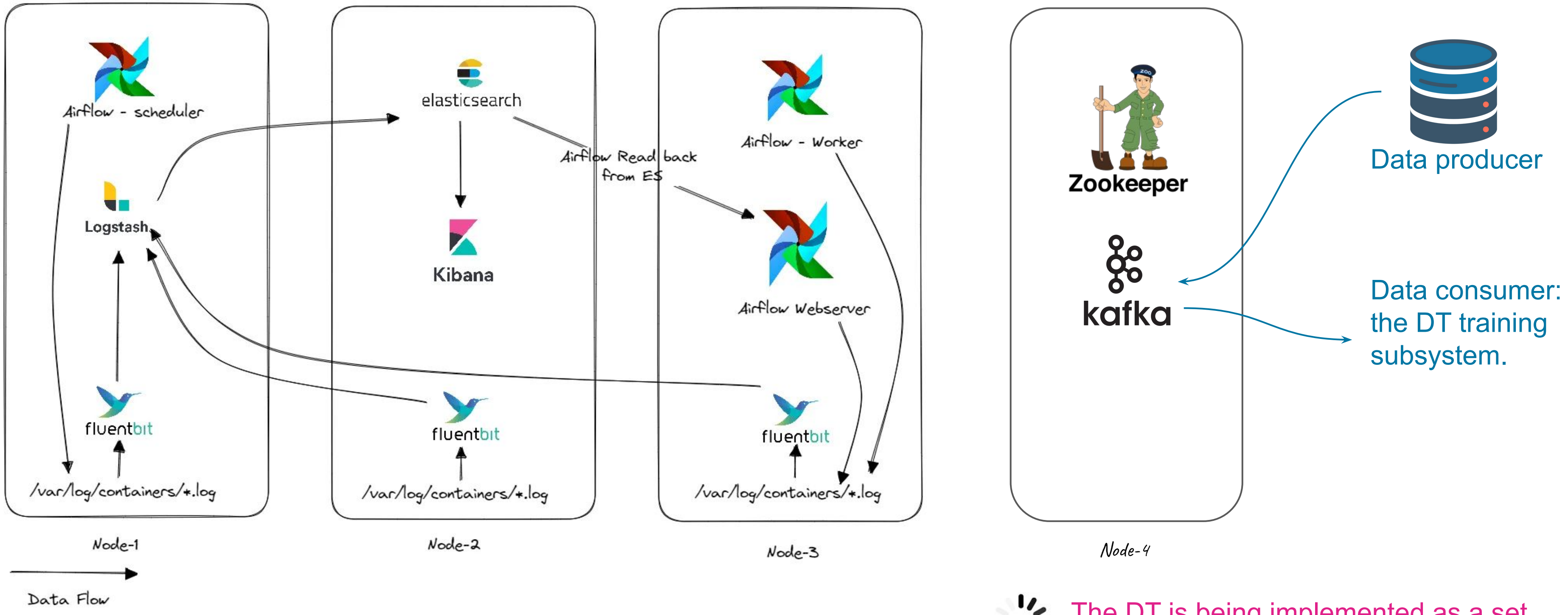
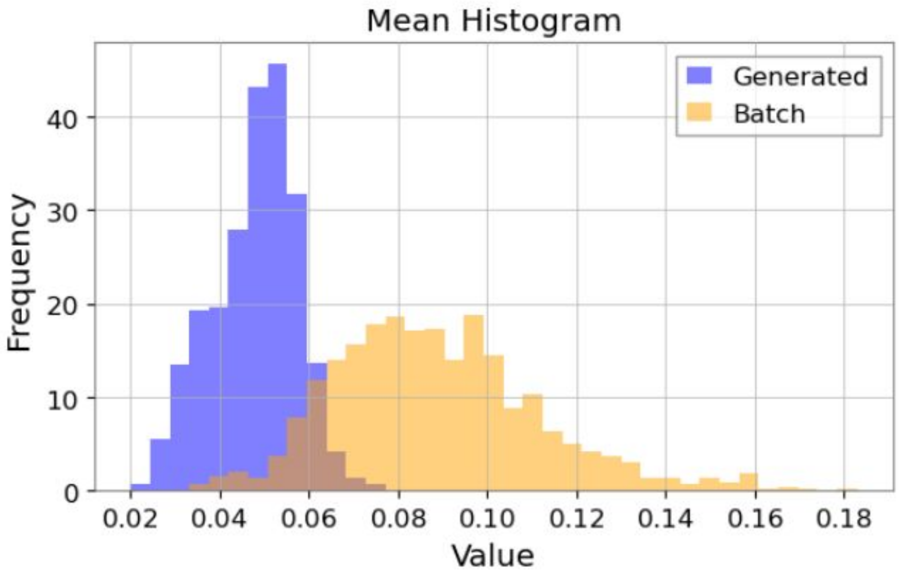


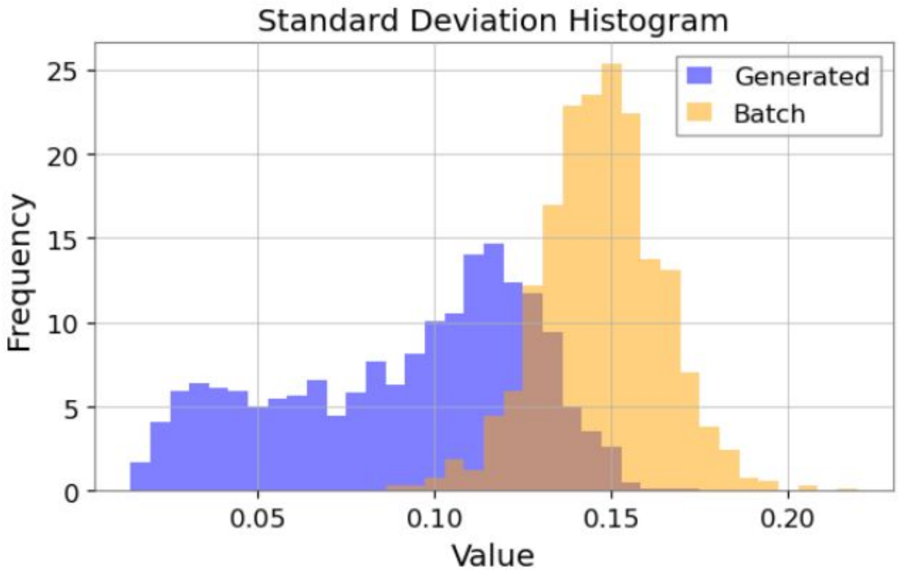
Image from:
https://medium.com/@dulshanr12/airflow-log-integration-with-fluent-bit-elk-stack-kubernetes-f2afa3a6ff00

LOADING...

The DT is being implemented as a set of **interdependent Airflow DAGs**.

Generated vs Real glitch distribution

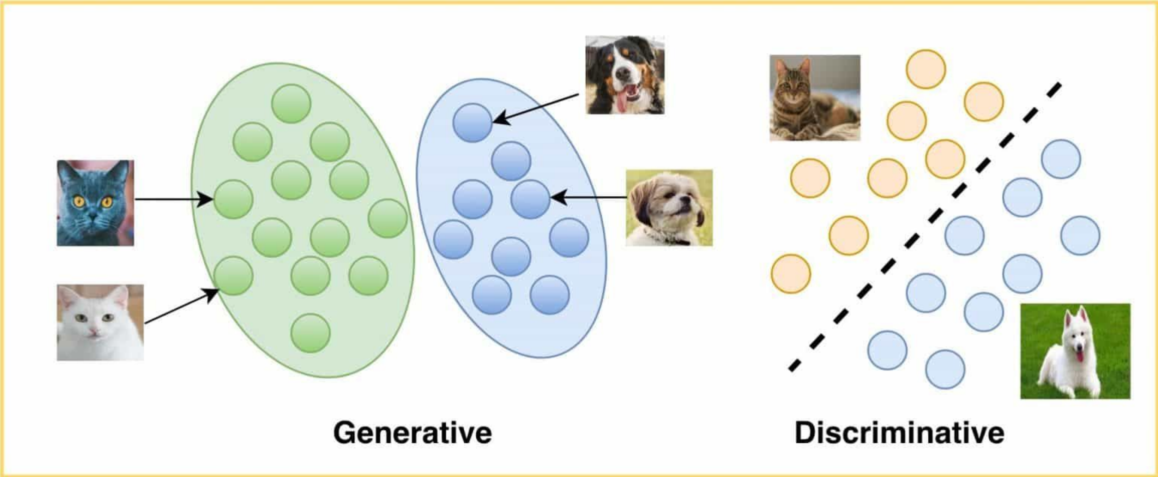




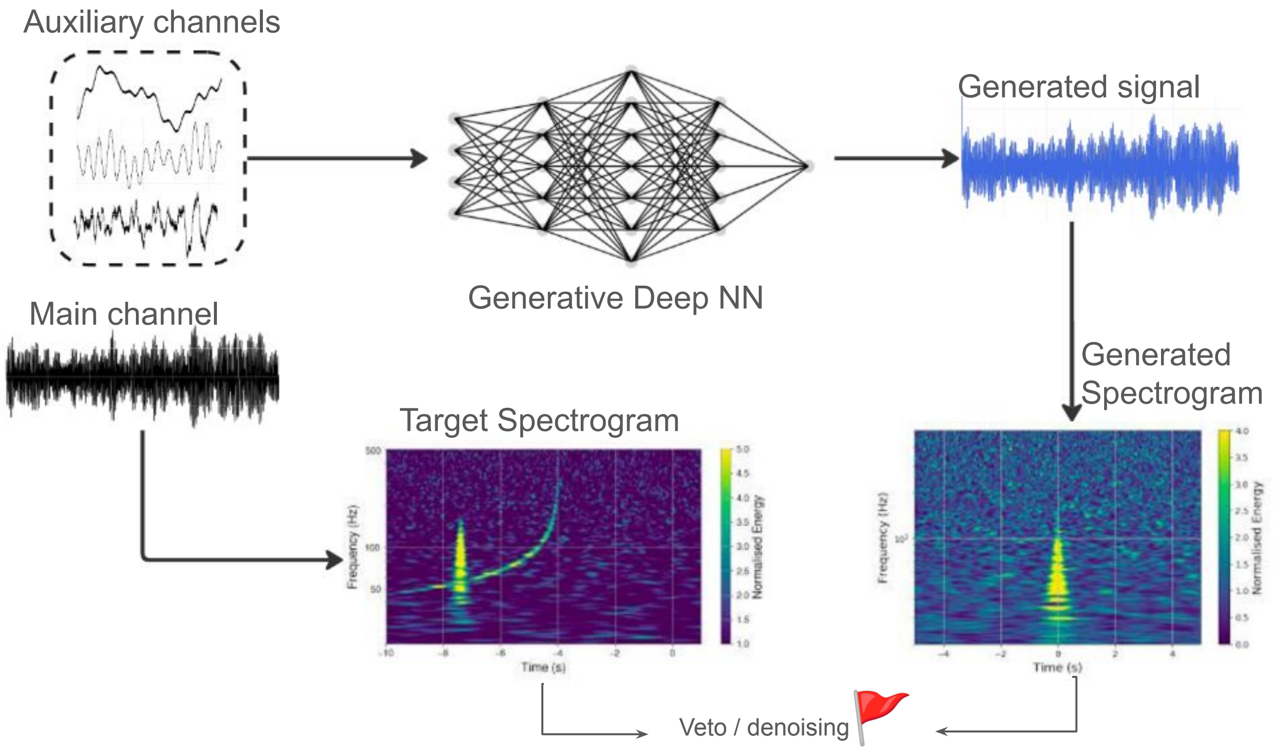
What are generative models?

- **Generative models** are **algorithms** that learn to **generate new data** that is **similar** to the **data** they were **trained on**

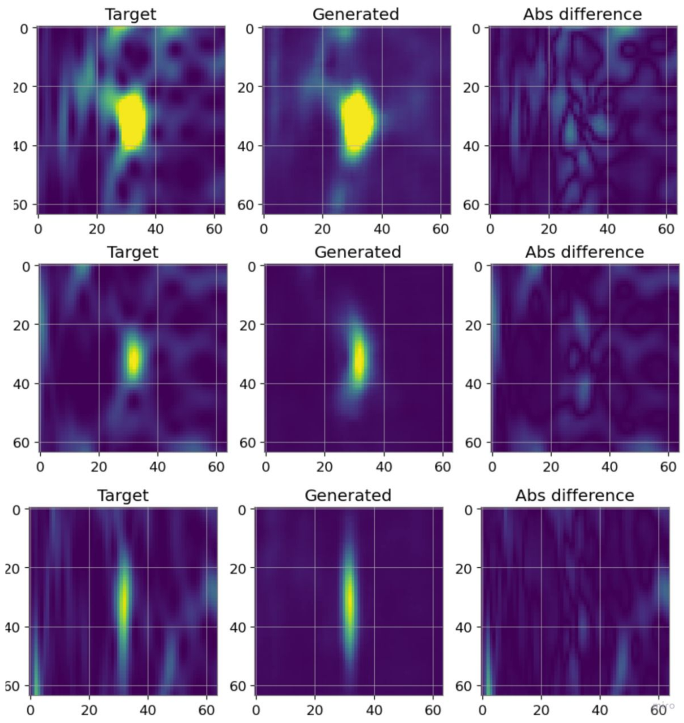
- They learn to **map data from** the **original high-dimensional space to** the l**ower-dimensional latent space** and then **generate new data** by **sampling** from this **latent space**.



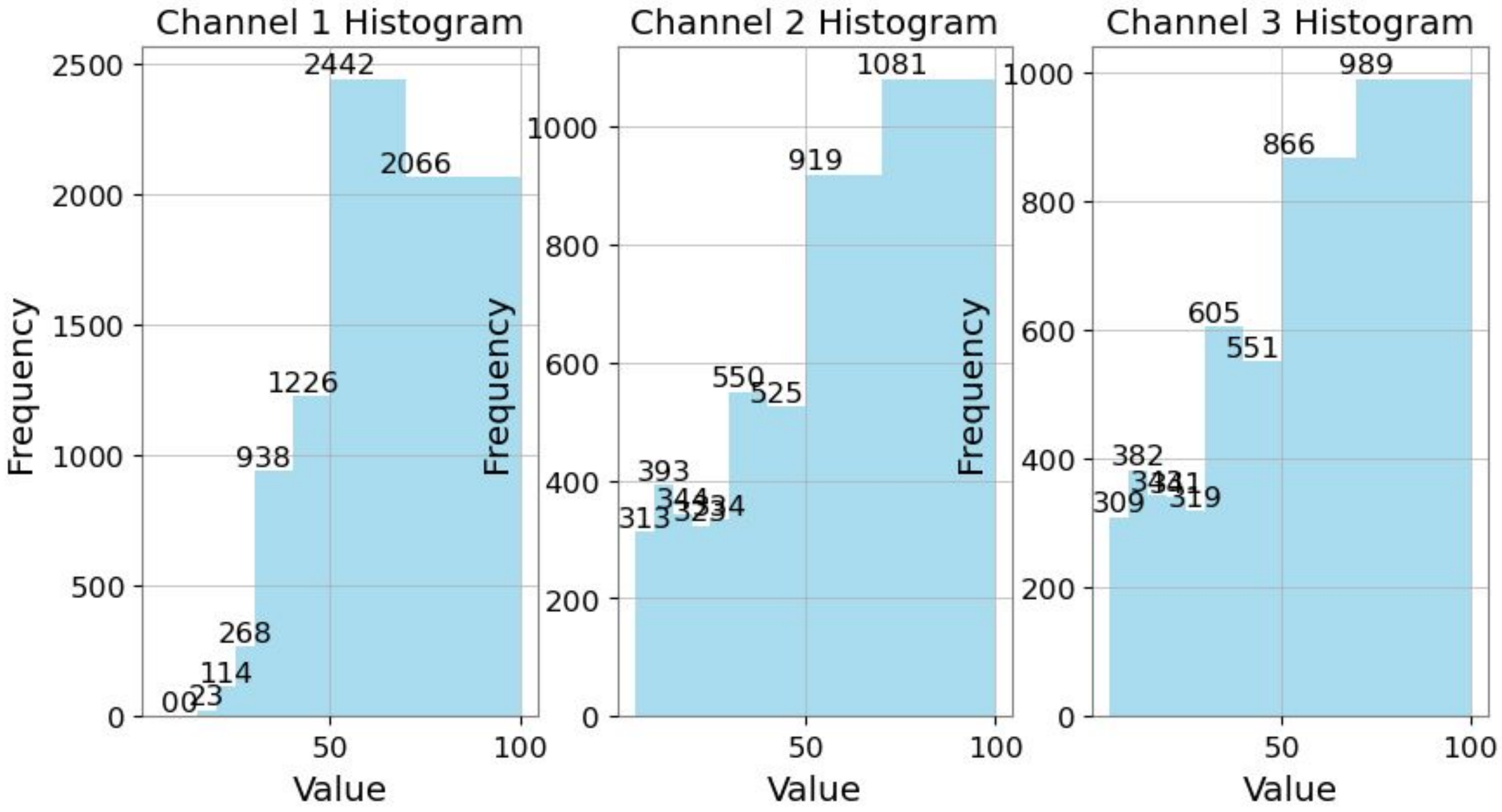
Glitch Flow Pipeline



Glitch Flow Inference



SNR distribution of data



Model Performance

- The model **correctly predicts** the **presence of noise** in signal from only looking at **aux channels** in **100%** of cases
- The model **removes** the **noise from the signal** in **59,7%** of cases

How to improve

- Use more complex NN architectures
- Data augmentation
- Use more and more appropriate auxiliary channels
- Build more sophisticated tools for channel analysis
- Much more …

The two auxiliary channels which were used in the analysis are:

V1:LSC_MICH_ERR

Deviation in the Michelson interferometer signal (sampling rate:10000 Hz, measured in Ampere).

V1:LSC_PR_CORR

Correction on the voltage in the Power Recycling cavity (sampling rate:10000 Hz, measured in Volts).

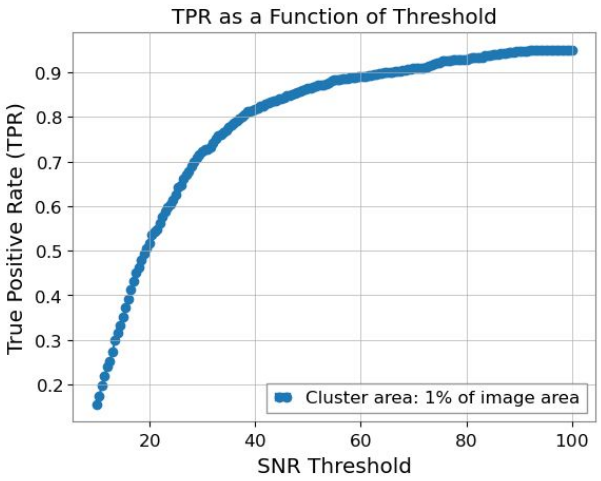
These channels are both safe, and they are used in the linear denoising in the strain channel.

Performance Tests: Denoising

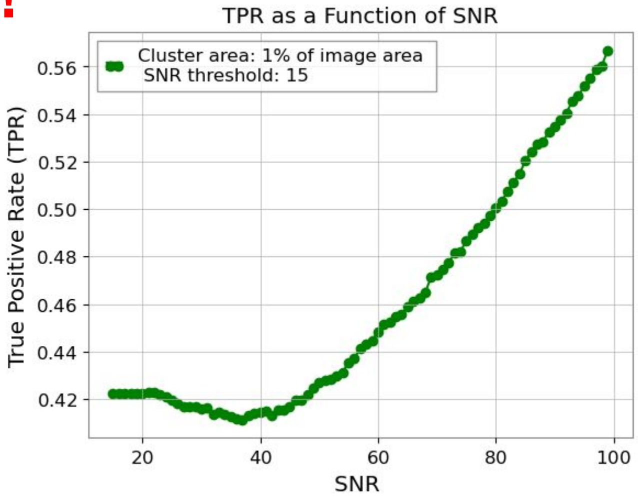
We pass the subtraction of the real signals minus the generated ones to the classifier. For each threshold value of the SNR, we construct a confusion matrix. We plot the True Positive Rate (TPR) for two different studies.

Needs Improving!

1. Varying the SNR in the classifier for the definition of a glitch



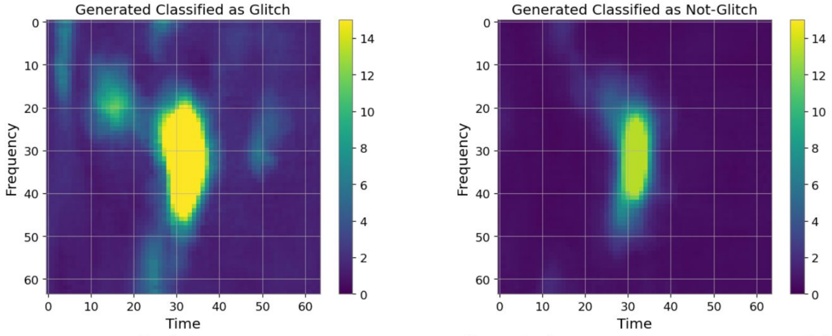
1. Keeping SNR=15 in the classifier but dropping the data below a given threshold



Performance Tests: Vetoing

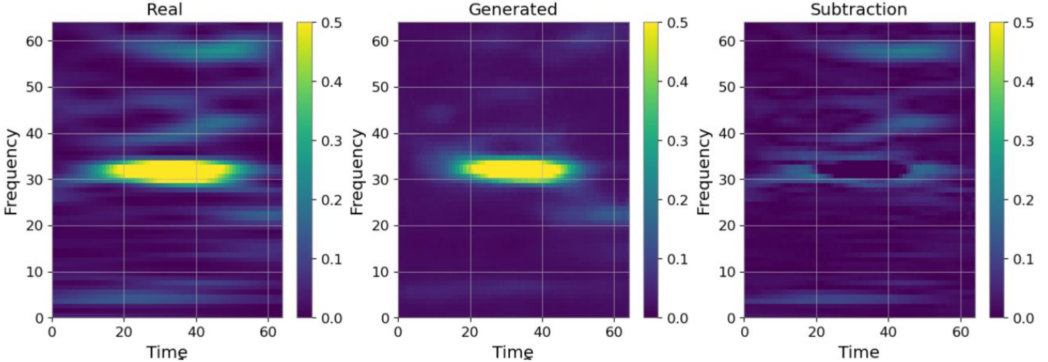
We construct a classifier which labels Qplots as glitches/not-glitches. It is based on a clustering algorithm: a glitch is defined as an area made of at least 10 pixels which have SNR equal or above 15.

1. **Accuracy of glitch generation**: 97.8 % of the generated data is identified as glitches by our classification algorithm (tested on a sample of 1083 generated signals plus 536 injected with background noise)



2. **Accuracy of glitch positions**: the generated glitch has the correct time-frequency coordinates in 99.6% of the cases

Works Well!



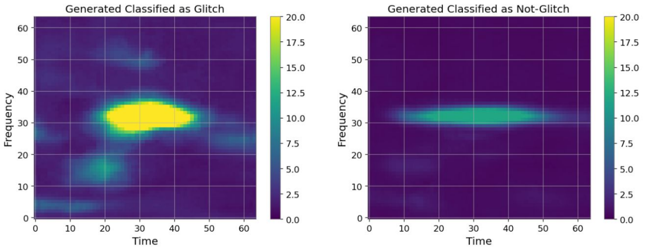
Performance Tests: Vetoing

- ***Glitch definition***: **Cluster** of at least **10 pixels** with **SNR above threshold (15,10,8)**. This choice mimics actual alert mechanisms in use at Virgo (Omicron)
- Use **Clustering** mechanism as **Classifier**
- **Test set**: 1083 Glitches, 536 empty background

Accuracy of glitch generation

Model correctly predicts the presence of a glitch given control channels

Accuracy		
SNR=8	SNR=10	SNR=15
15%	22%	35%



Accuracy of glitch positions

Model correctly predicts time-frequency coordinates of glitch given control channels