GlitchFlow, a Digital Twin for transient noise in Gravitational Wave Interferometers

L. Asprea[1,2], E. Cellini[3], **F. Legger[1,2]**, A. Romano[2,3], F. Sarandrea[1,2], S. Vallero[1,2]

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

CHEP 2024 - 24/10/2024

Funded by the European Union

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

The InterTwin project

- **EU funded**
- **Aim:** design and build a prototype of an interdisciplinary **Digital Twin (DT)** Engine, based on a co-designed Blueprint Architecture

36 months
from
Sep 22
To
Aug 25

Budget

12 M Euros

30 Partners
with
1500 PMs
Coordinated by
EGI

2 + 2 + 3 DT
Use Cases

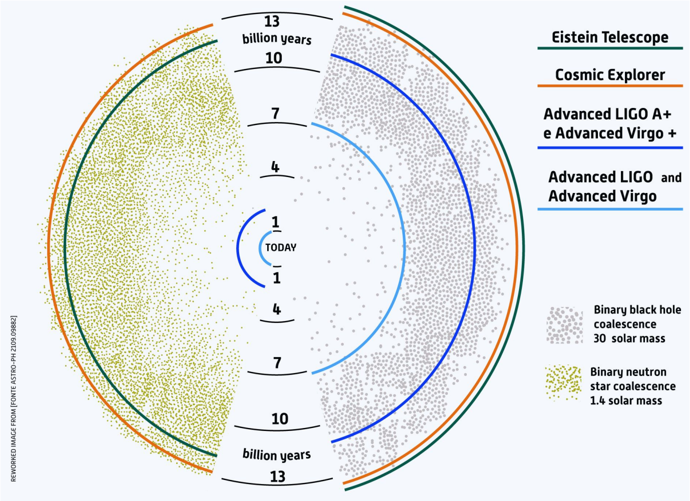
from

HEP
Astro
Climate
Environment

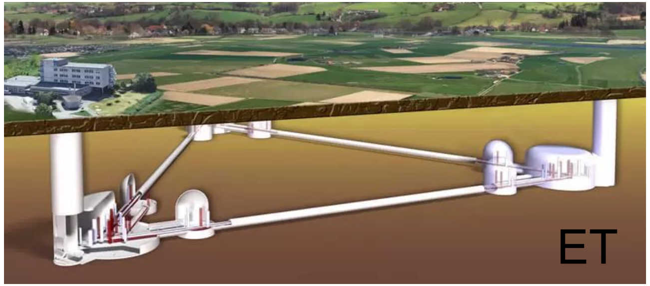
Transient noise simulation for Virgo and Einstein Telescope

Gravitational Waves and Interferometers

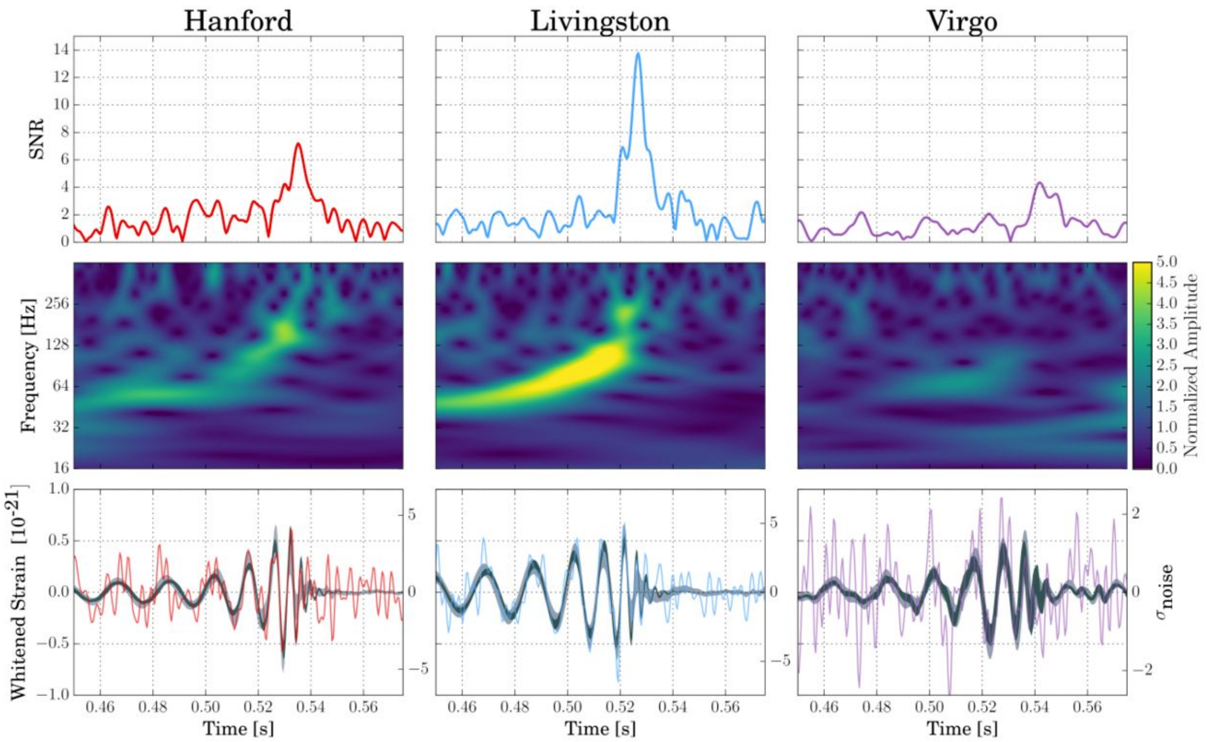
- **Gravitational waves (GWs)** for:
  - cosmology and early universe
  - fundamental physics
  - **Multi-Messenger Astronomy**



- **GW Interferometers**
  - may detect GWs produced by the acceleration of massive objects, such as **black holes** or **neutron stars**
  - Current generation: LIGO, **Virgo**, KAGRA
  - Next generation: **Einstein Telescope (ET)**, Cosmic Explorer



GW Detection



GW170814, PRL 119, 141101 (2017)

- based on the **strain** measurement (deformation of the interferometer arms)
- interferometer status and environmental conditions are monitored in the **auxiliary channels**

Goal of the Digital Twin

- sensitivity of GW interferometers is limited by **noise**
- DT aims to realistically simulate and detect **transient noise (glitches) quasi-real time**
- Final goal: **veto** and (later) **de-noise**

- GW signal → strain
- transient noise (**glitches**) → auxiliary channels and strain

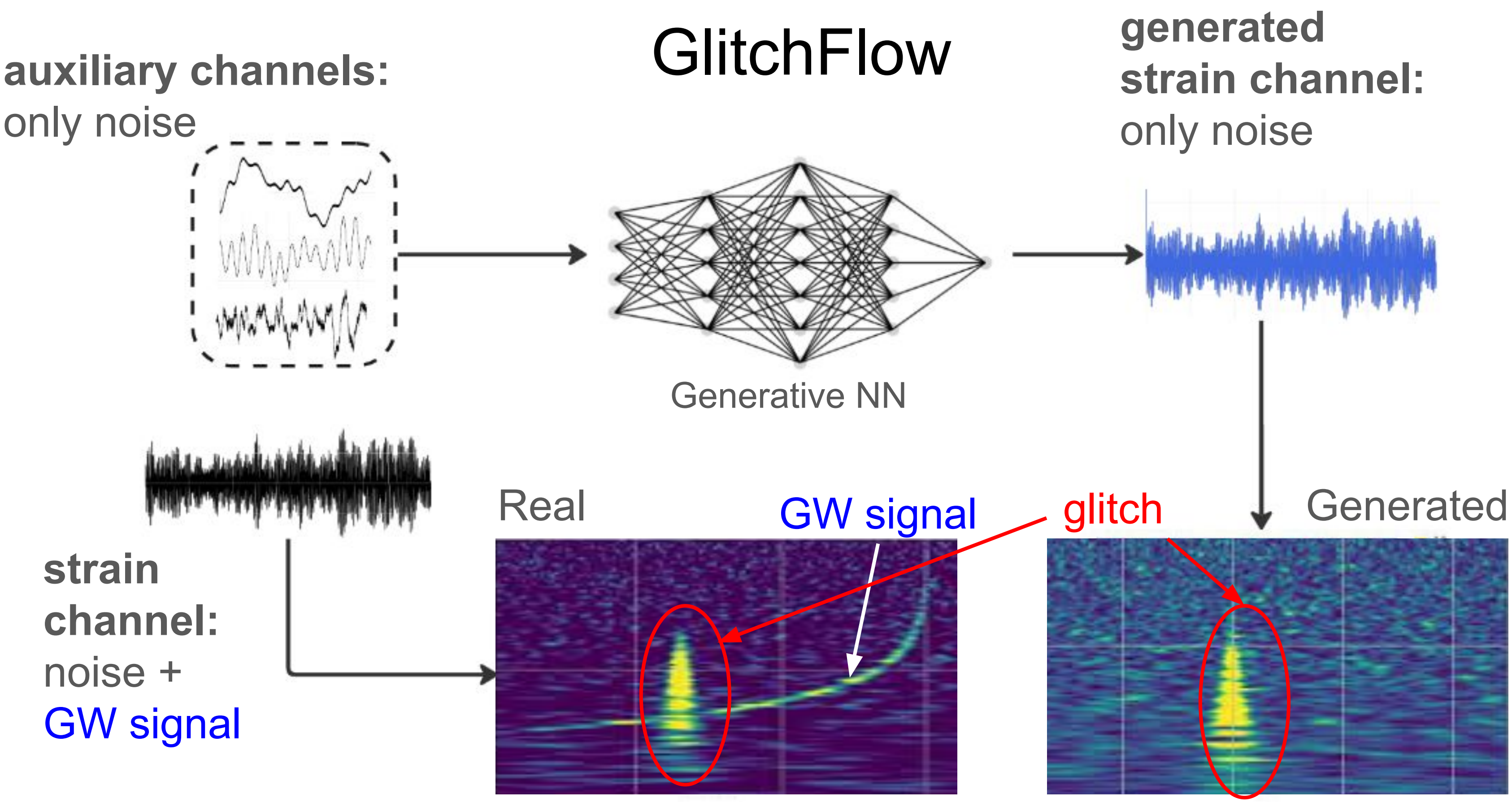
→

- map glitches from auxiliary channels to strain
- **deep generative models** to capture non-linear structures in the data

→

- **Veto** events containing glitches
- **Noise subtraction** from the strain channel

GlitchFlow



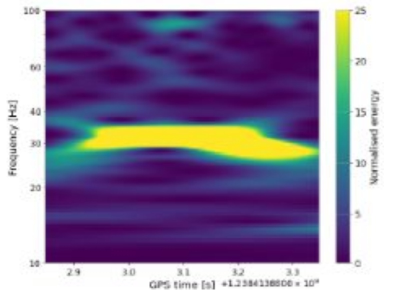
auxiliary channels:
only noise
Generative NN
generated strain channel:
only noise
strain channel:
noise +
GW signal
Real
GW signal
glitch
Generated

Neural Network architectures

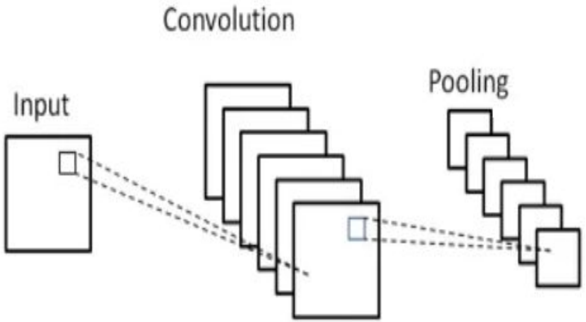
- Code in Pytorch
- **L1-Loss= Σ |Generated Output - Target Output|**

- **Input:** 2 aux channels

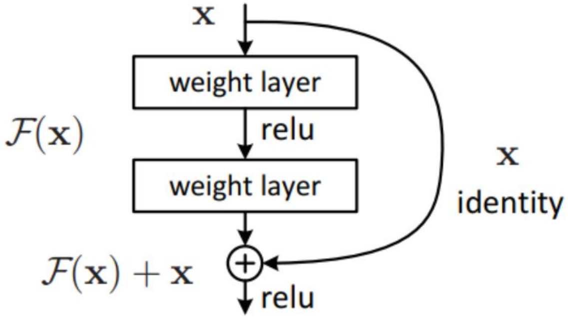
2D



CNN

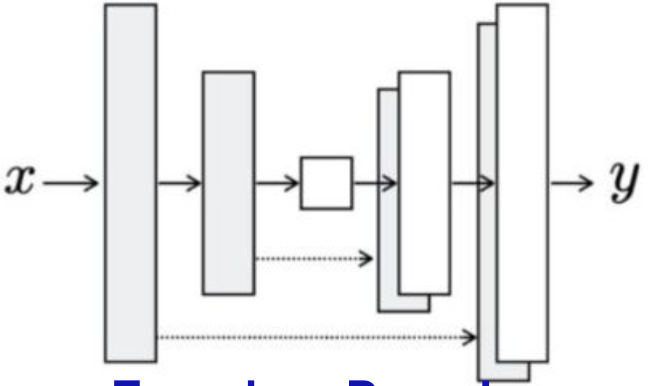


ResNet (residual block)



Skip (Shortcut) connection

U-Net



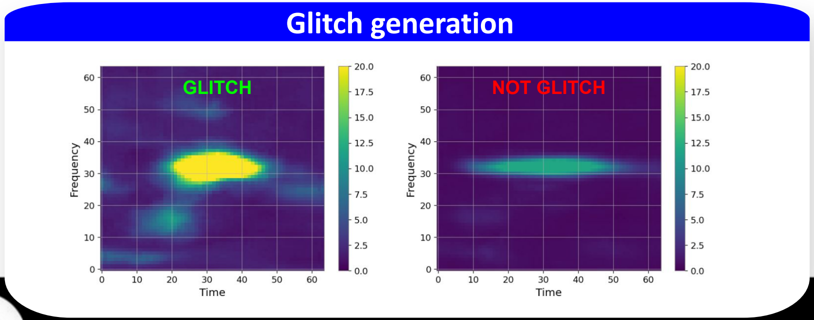
Decoder

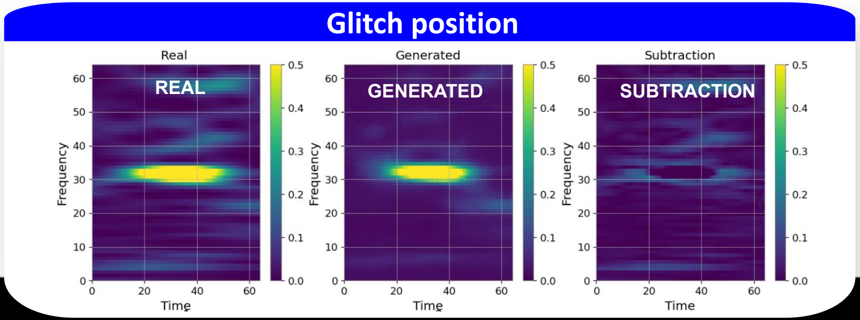
Encoder - Decoder

Performance Tests: Vetoing

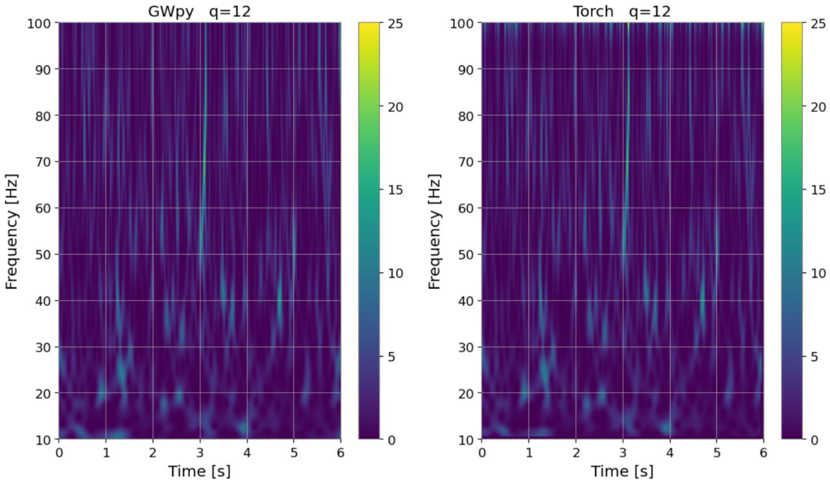
- ***Glitch definition*: Cluster** of at least **10 pixels** with **SNR above threshold**
  - This choice mimics actual alert mechanisms used by Virgo (Omicron)
- Use **Clustering** mechanism as **Classifier** on **generated data**
  - **Test set**: 1083 Glitches, 536 empty background

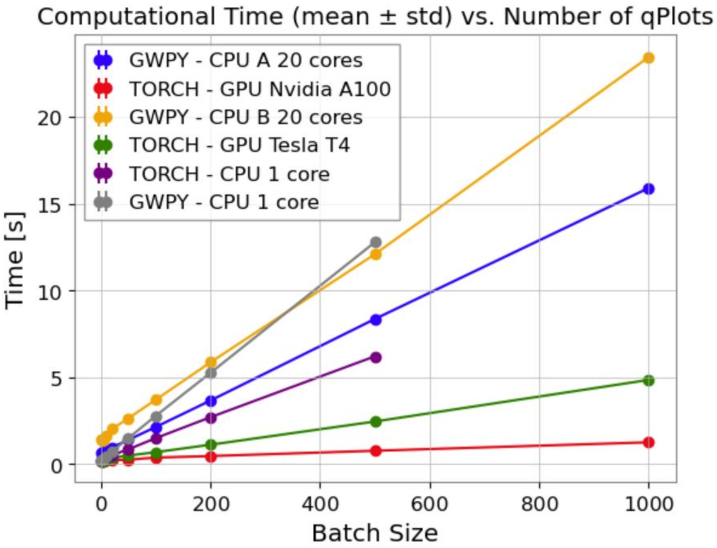
	SNR 8	SNR 10	SNR 15
Glitch generation Accuracy Model is able to correctly generate a glitch given control channels	99.0%	98.7%	97.8%
Glitch position Accuracy Model generates glitches with right time and frequency. Intensity is saturated at trigger SNR	25.9%	99.0%	99.6%





Pytorch based Q-transform on GPU





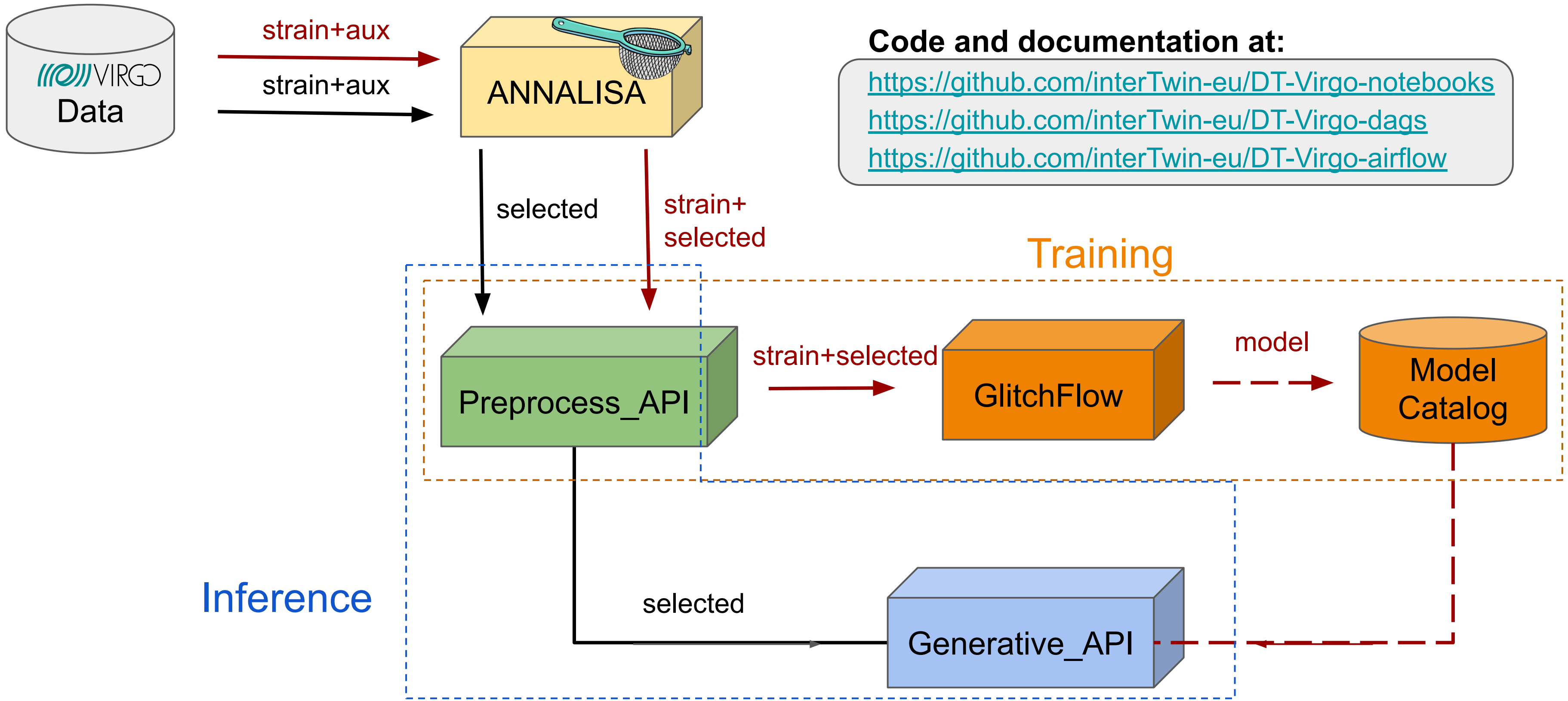
Enables:

- Broader channel scan with more statistics
- Low latency data preprocessing and inference
- Time series as NN input, as q-transform can be used in the loss (autograd).

⇨ Easier normalization of data and NN convergence

NEW

Digital Twin Engine: Thematic modules

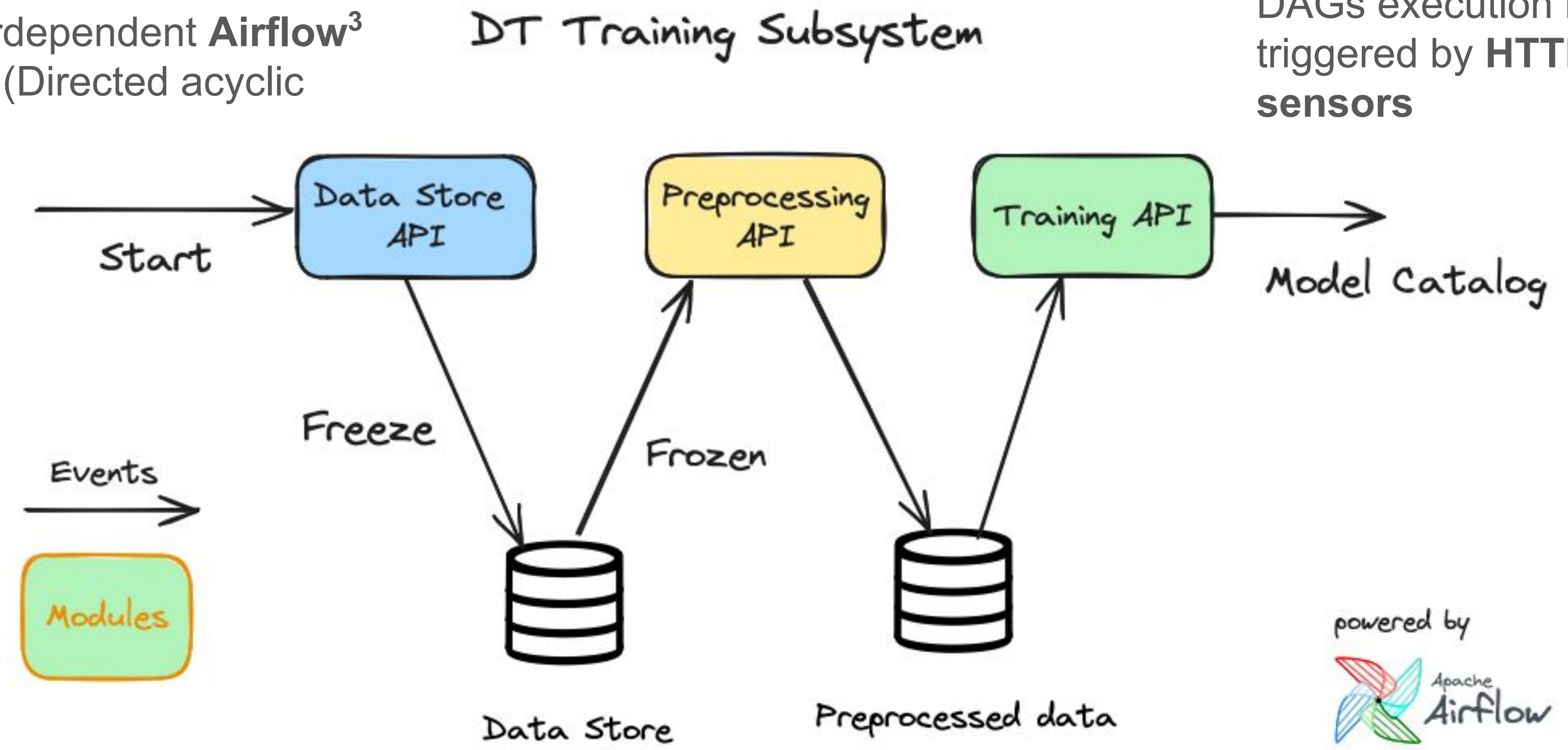


Code and documentation at:

https://github.com/interTwin-eu/DT-Virgo-notebooks
https://github.com/interTwin-eu/DT-Virgo-dags
https://github.com/interTwin-eu/DT-Virgo-airflow

DT implemented as a set of interdependent **Airflow**[3] **DAGs** (Directed acyclic graph)

DAGs execution is triggered by **HTTP sensors**

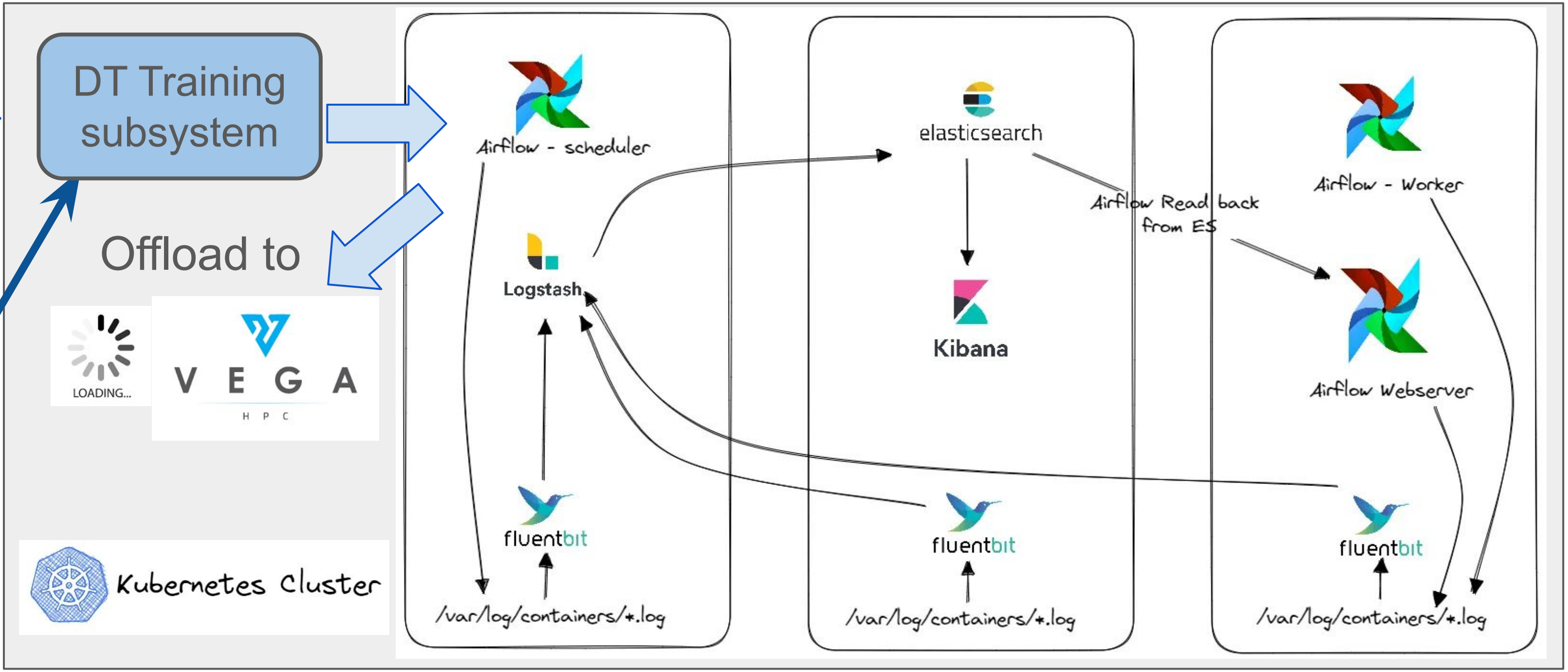


[3] https://airflow.apache.org/

DT training subsystem at CNAF

Data consumer

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



Data producer

Currently implemented as HelloWorld modules

Outlook

- Development of a DT for transient noise simulation for GW interferometers - ongoing
- **Training subsystem**:
  - **GlitchFlow**:
    - Larger dataset and more auxiliary channels
    - Use time series as input instead of spectrogram
    - Develop denoising algorithm
  - **Analysis pipeline**:
    - Implement offload to VEGA
    - Set up DAGs with real modules
- **Inference subsystem**: to be developed

Thank you!

Questions?

www.intertwin.eu

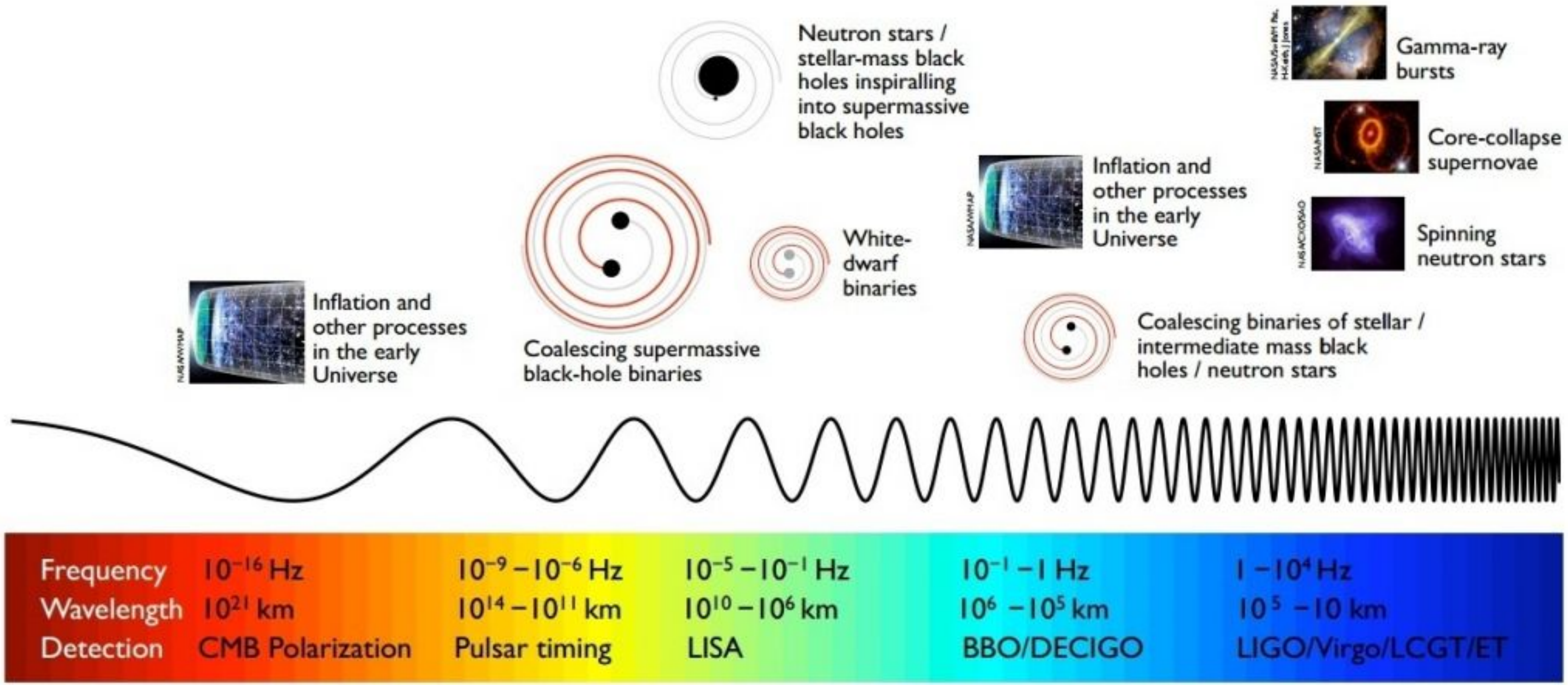
info@intertwin.eu

intertwin_eu

intertwin

Backup slides

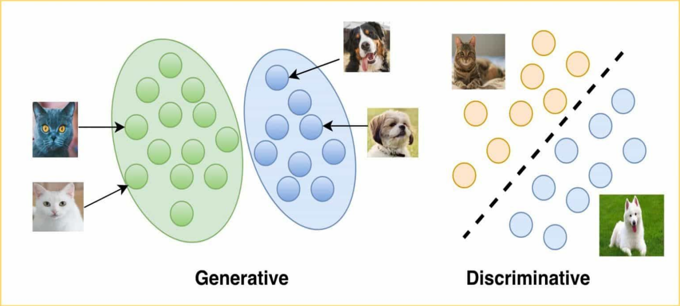
The Gravitational Wave Spectrum



What are generative models?

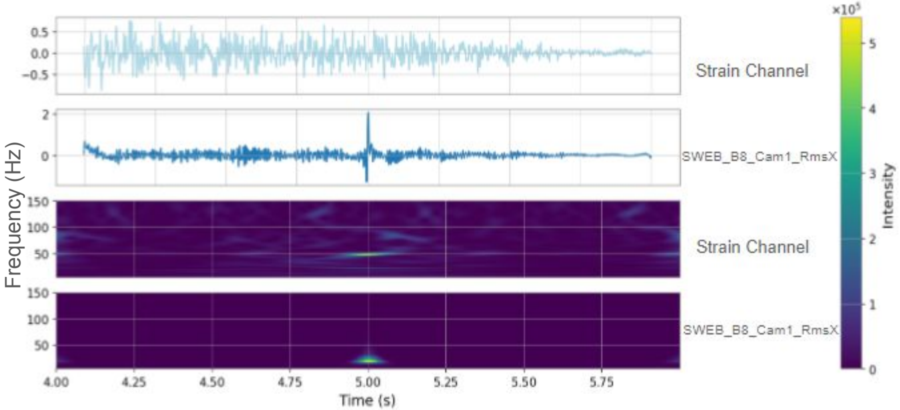
Generative models are **algorithms** that learn to **generate new data** that is **similar** to the **data** they were **trained on**

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



What are Spectrograms?

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



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

What are Safe Channels?

Channels that are not witness of any sort of astrophysical signal, i.e. channels measuring quantities unaffected by the passage of GWs.

Dataset

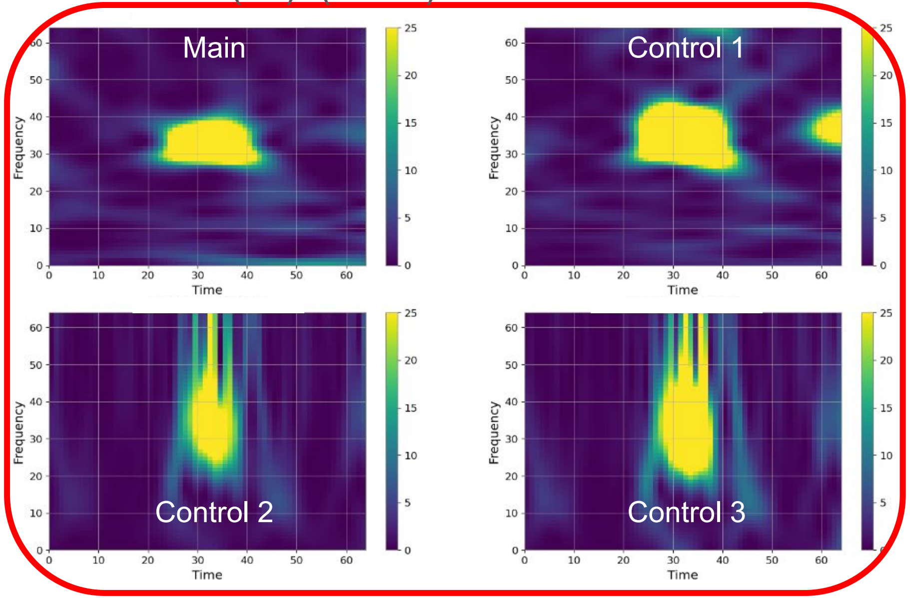
Time series

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

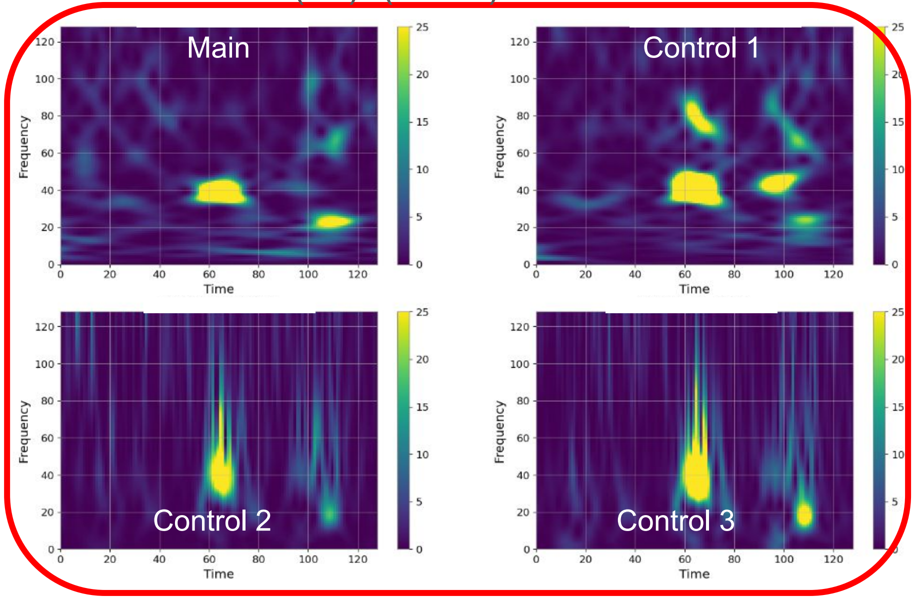
Qplots

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

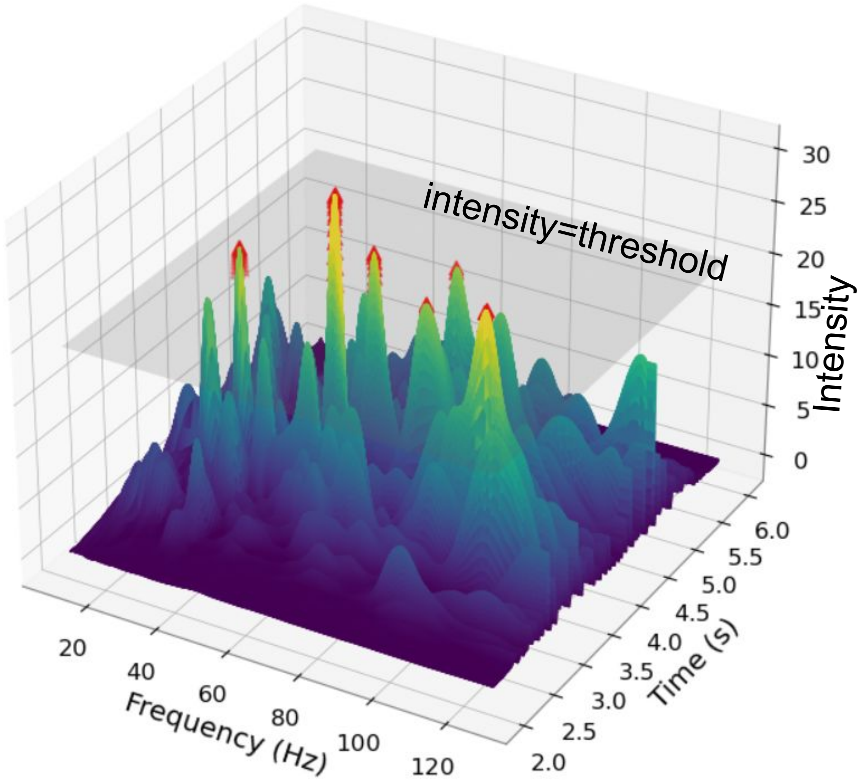
64x64 ~ (1s)x(35Hz)



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



ANNALISA: finding the right channels

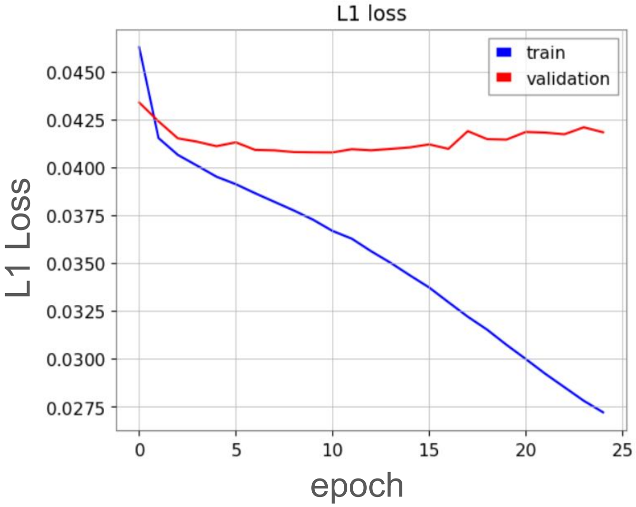


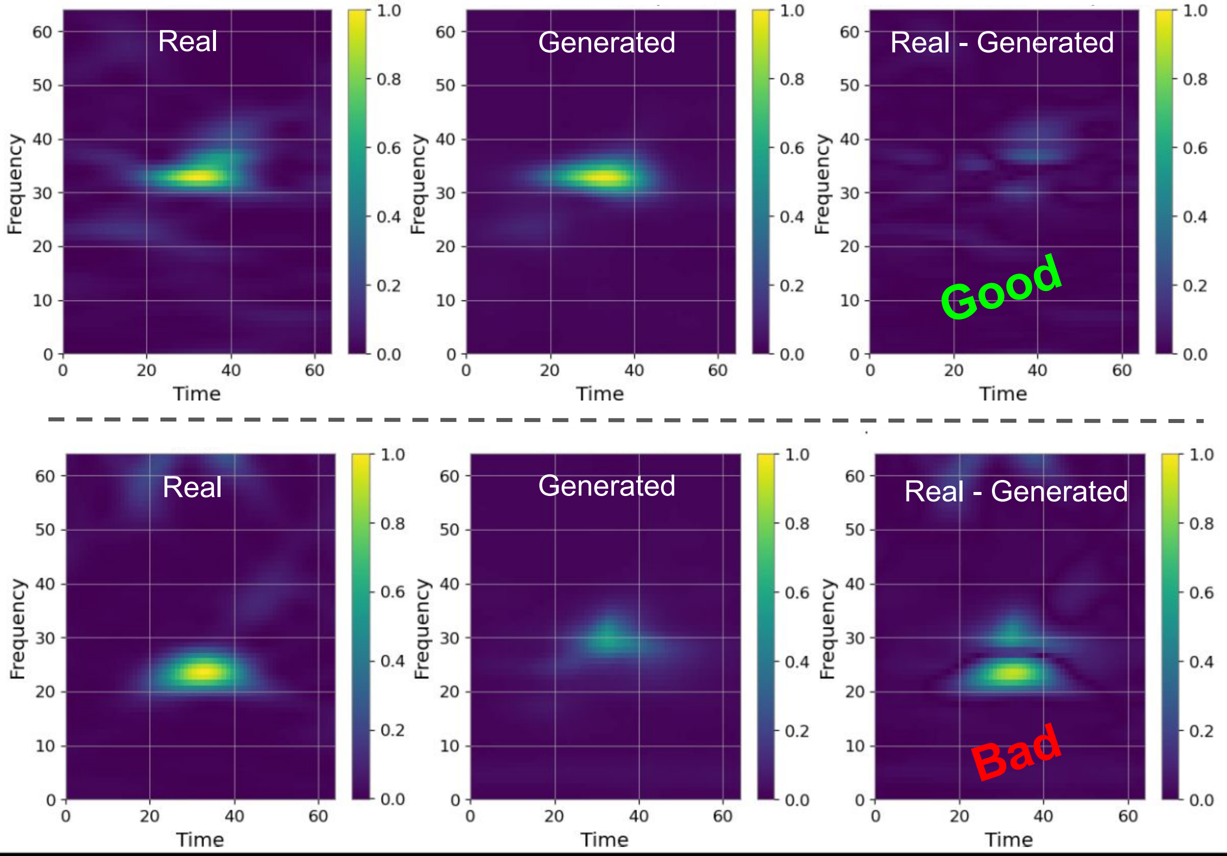
- ***Common peaks***: peaks above intensity threshold at same time (allows for non linear correlations)
- Apply analysis to each aux channel paired with strain channel
- Calculate correlation coefficient:

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

- Data source: VIRGO O3a
- https://github.com/interTwin-eu/DT-Virgo-notebooks/tree/main/WP_4_4/Annalisa-0.1.tar.gz

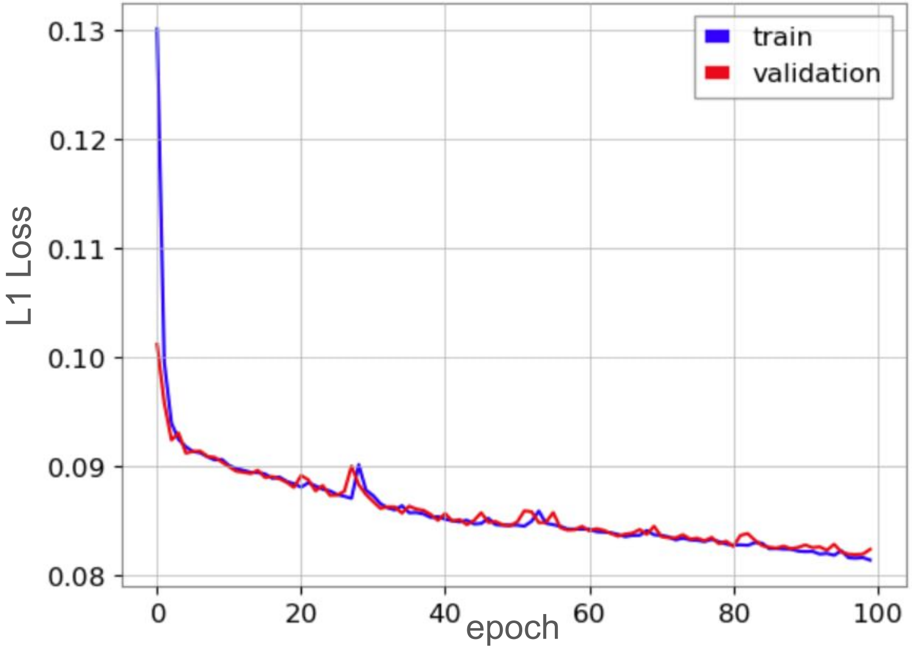
QPlots: Resnet (12 blocks)

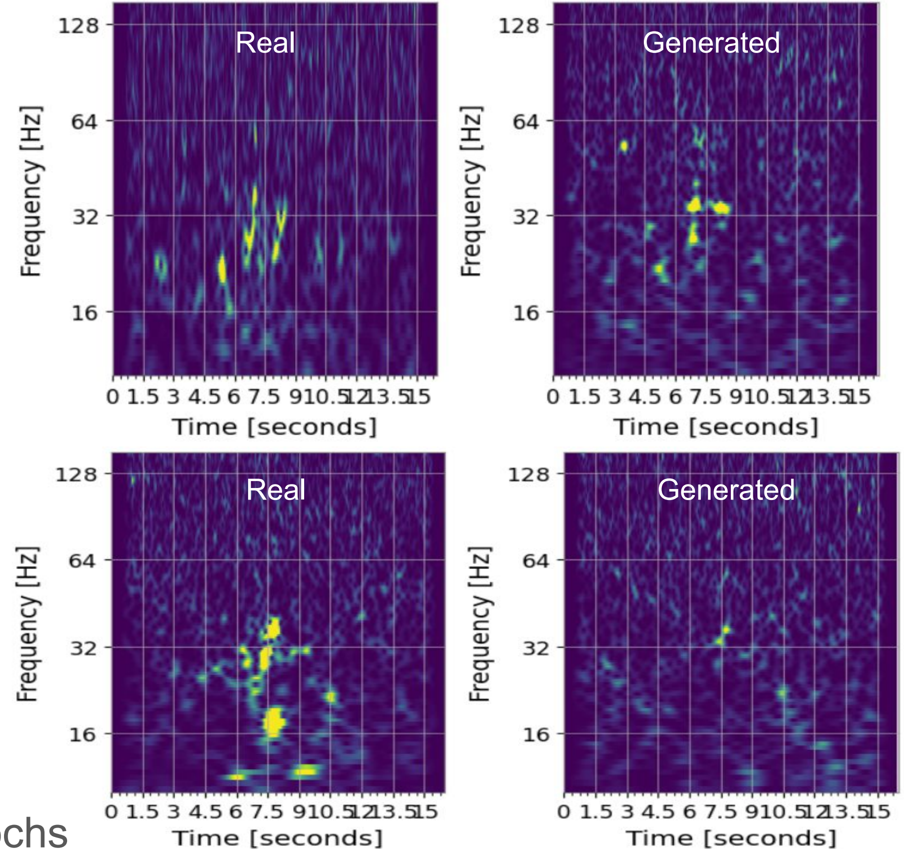




- **Mostly very good results**, but some bad ones as well in presence of messy or odd looking input
- Network seems to stop learning after very few epochs. **High Bias after very few epochs**

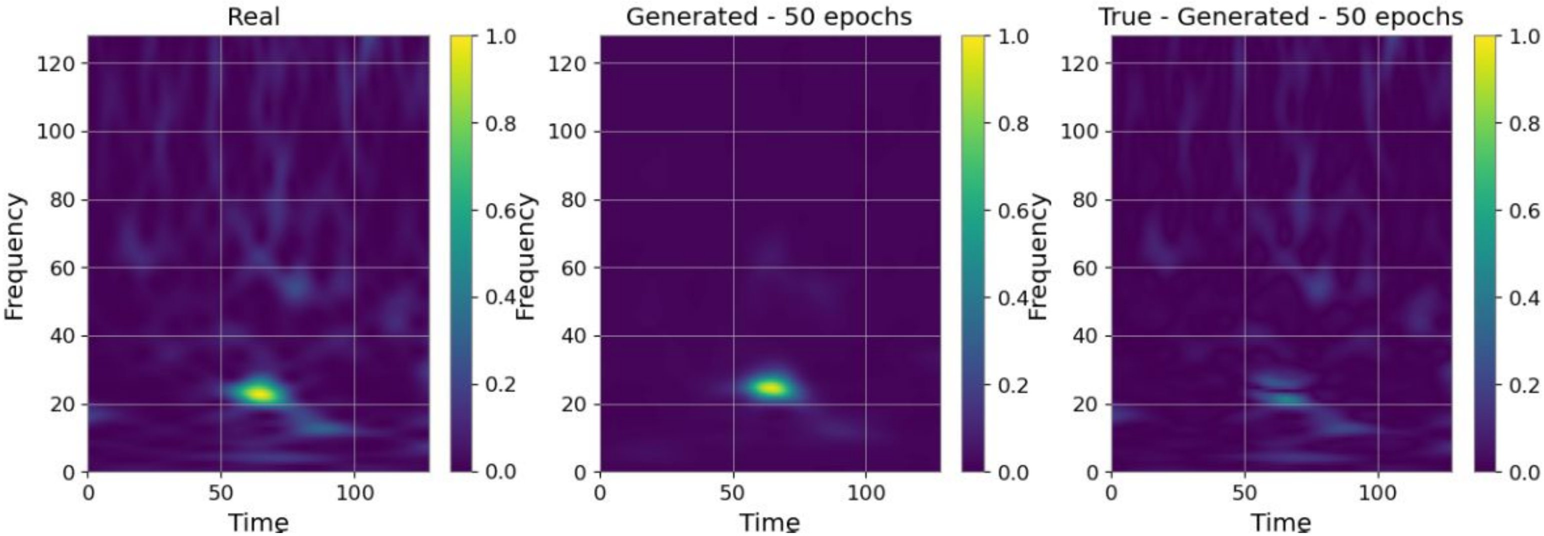
Time series: U-Net





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

QPlots: Decoder CNN+ Resnet



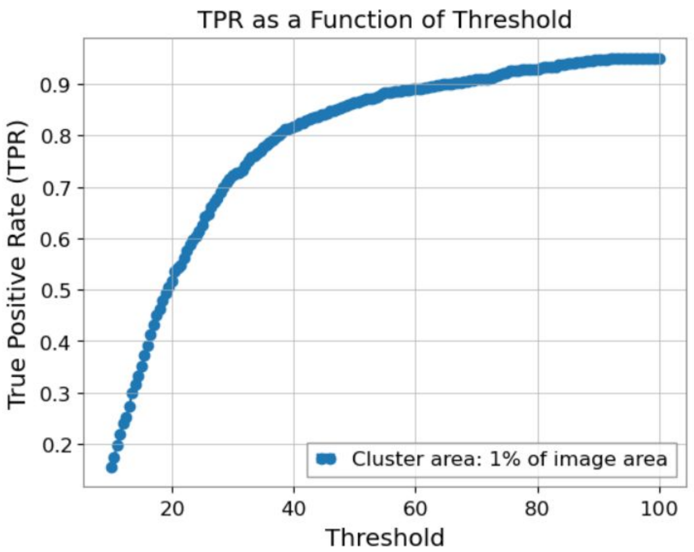
- Using 64x64 model as pre-training, only one epoch of training is required!
- High Bias after very few epochs
- Network does not seem to notice light blue/green background (good for noise stability?)
- Bad results for messy (or odd looking) input
- Very similar loss and results for U-Net

Performance Test: Denoising

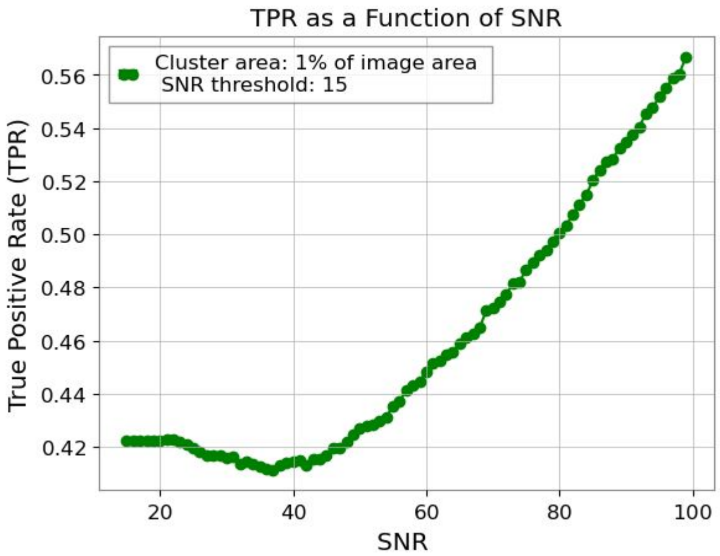
- ***Glitch definition***: **Cluster** of at least **41 pixels** (1% of total Qplot area) with **SNR above threshold**.
- Use **Clustering** mechanism as **Classifier** on **cleaned data** (i.e. real minus generated)
- **Test set**: 1083 Glitches, 536 empty background

WORK in PROGRESS

Dataset fixed, varying Classifier Threshold



Classifier Threshold fixed, varying SNR in Dataset



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.