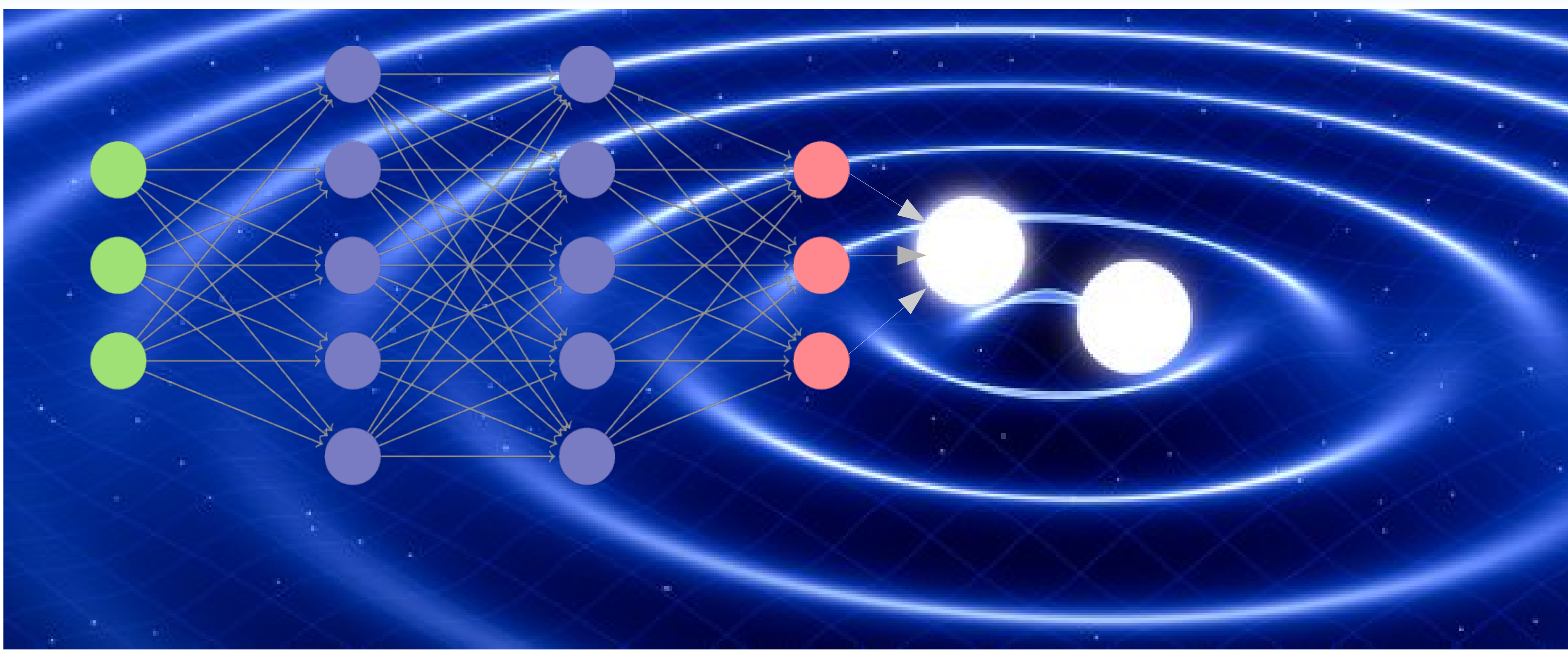
CNN for early alert: Going down in frequency

Grégory Baltus[1]
J. Janquart[2,3], M. Lopez[2,3],
H. B. Narola[2,3], J.R. Cudell[1]

Gw1-g2net: Valence 2022



Goals and motivations

Goal : Detect the early inspiral phase of a BNS with convolutional neural network (CNN)

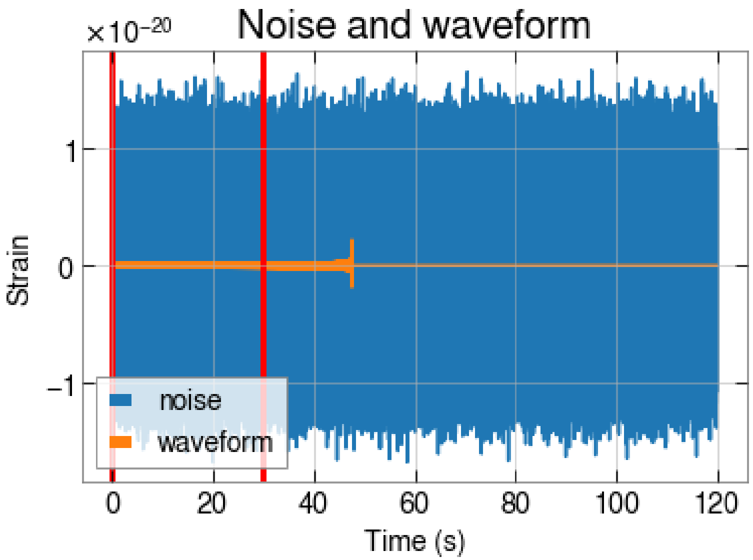
Motivation:
Multi-messenger astronomy
early alert system

Why deep learning ?

- Computationally cheap after the training
- Fast data frames analysis
- Does not require a template bank during the search

Building on previous work

Convolutional neural networks
for the detection of the
early inspiral of a gravitational-wave signal
arXiv:2104.00594

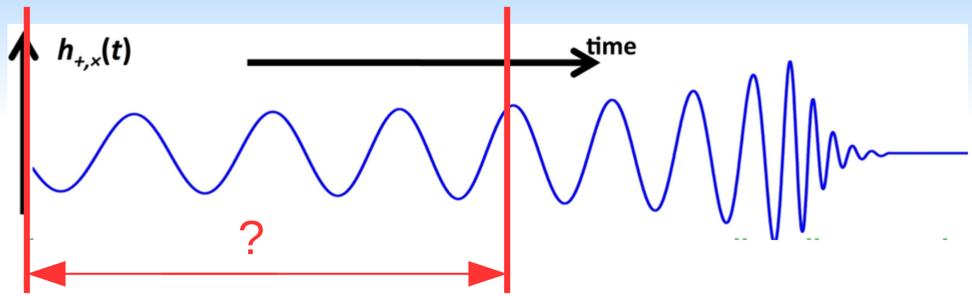


CNN

Output : [0,1]
If output > Threshold
→ Signal
If output < Threshold
→ Only noise

Proof of concept:
CNN to classify between noise and noise + early inspiral

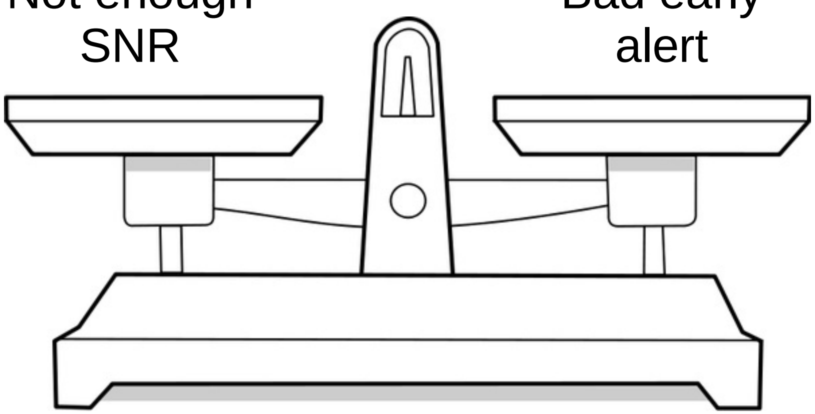
Observable time window



Which inspiral duration to choose?

Too short:
Not enough SNR

Too long:
Bad early alert



Problem : the length of the signal depends on the masses !

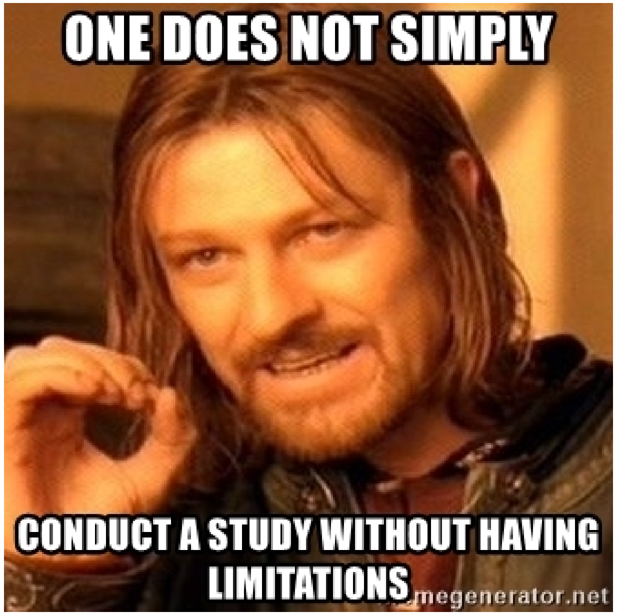
Solution : Multiples categories of BNS

Object	light BNS	intermediate BNS	heavy BNS
$\mathcal{M}_c$ $(M_\odot)$	1.21 - 1.56	1.56 - 2.09	2.09 - 2.61
$m_1 = m_2$ mass range $(M_\odot)$	1.4 - 1.8	1.8 - 2.4	2.4 - 3
f_{low} (Hz)	20	20	20
Duration (s)	160 - 100	100 - 65	65 - 45
OTW (s)	80	50	30
Fraction of signal	0.5 - 0.8	0.5 - 0.77	0.46 - 0.66
Early alert before merger (s)	80 - 20	50 - 15	35 - 15

Each category has its own CNN

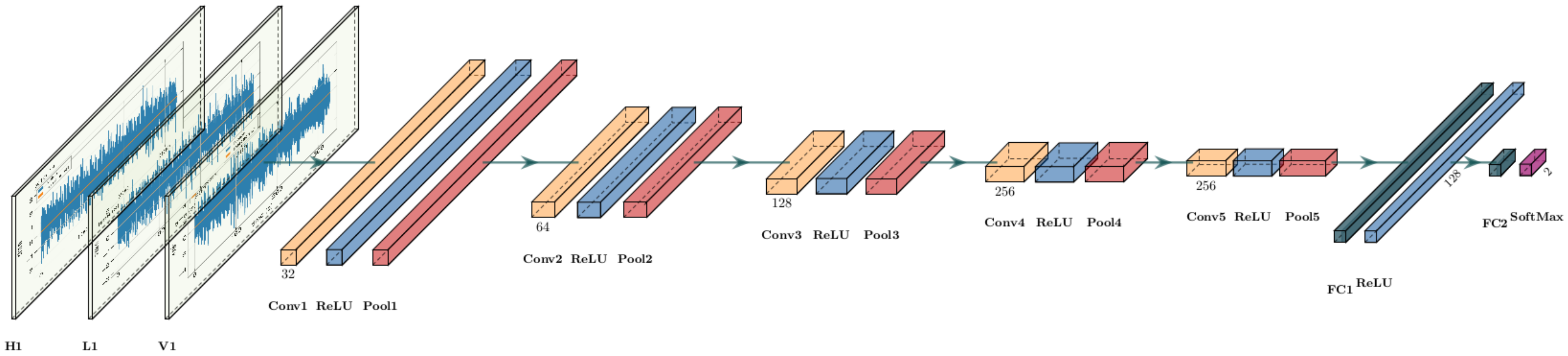
Limitations of the previous work

- Three BNS categories depending on masses with one network trained for each category
- Only one intereferometer with optimal sky position
- Colored Gaussian noise from design sensitivity PSD
- Data duration: 30, 50, or 80 s
- Minimum frequency: 20 Hz

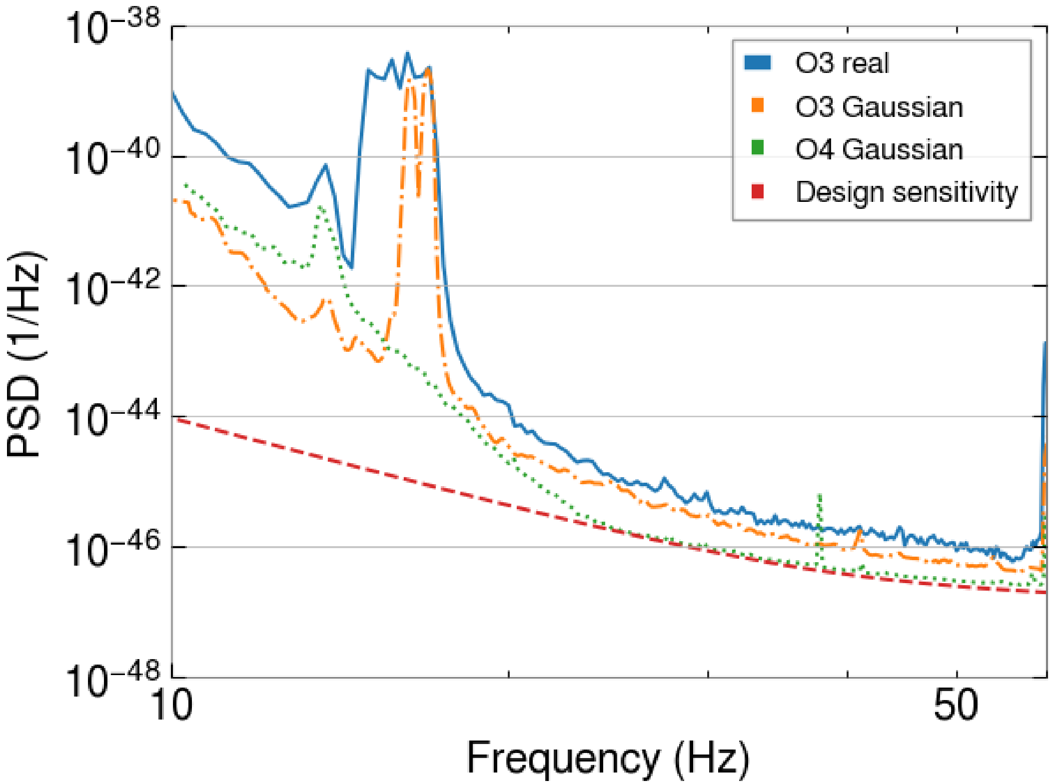


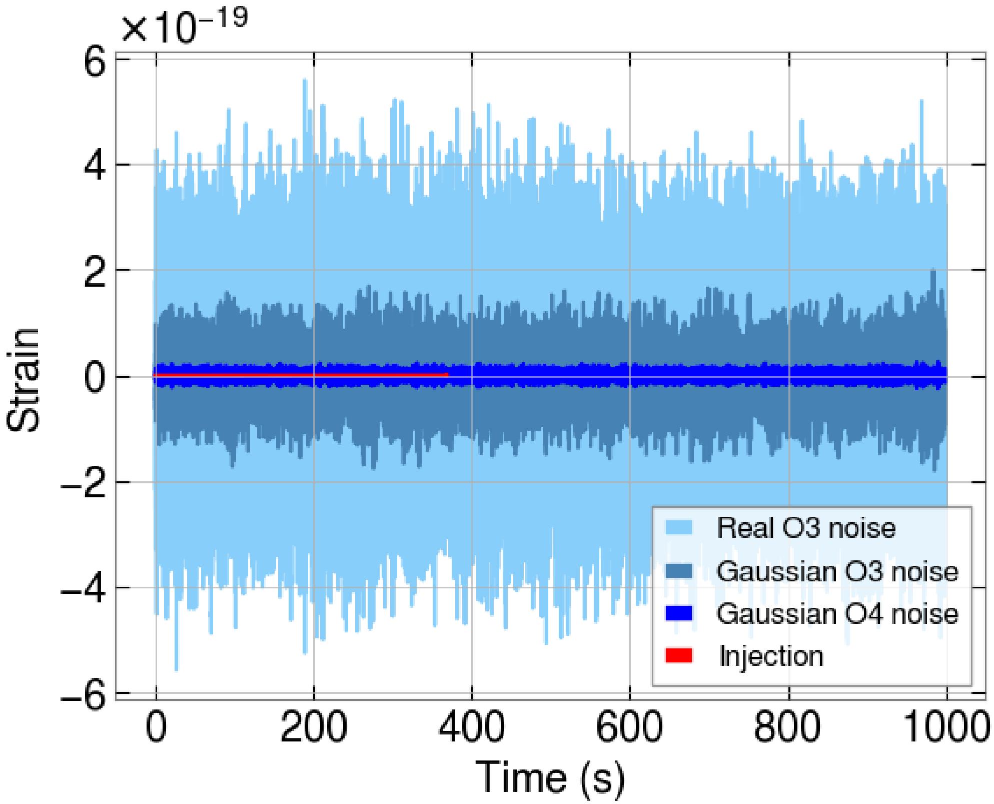
Upgrades

- Single CNN for all the BNS's, regardless of the masses
- One network taking into account the three detectors (H1, L1, V1)
- BNS's uniformly distributed in the sky
- Fixed input duration of 300 s
- Minimum frequency of 10 Hz



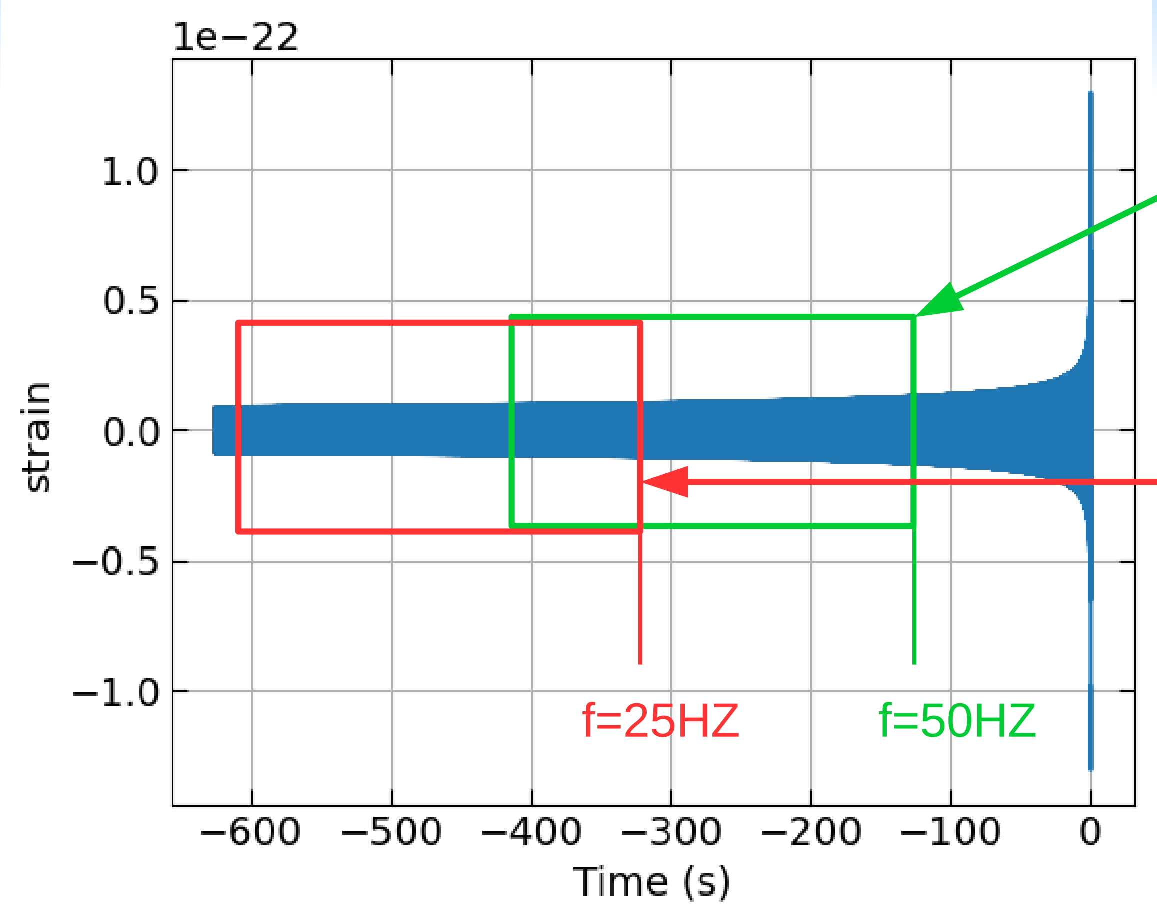
Three different types of noise





- **Gaussian O3 noise:** coming from pycbc
- **Real O3 noise:** channel: H1:GDS-CALIB_STRAIN frame type: H1_llhoft
- **Real O4 noise:** coming from MDC

Going down in frequency



The network has a good efficiency

The network efficiency degrades

Training with a uniform distribution of maximum frequency leads to bad performance when the maximum frequency starts to decrease
→ To overcome this problem: Curriculum learning on the frequency

Training details

<u>For each type of noise the training parameters were similar</u>

Curriculum learning data set:

Data set	Max Freq	Min Freq	Min TBM	Max TBM
Data set 1	40 Hz	10 Hz	7 s	44 s
Data set 2	35 Hz	10 Hz	10 s	63 s
Data set 3	30 Hz	10 Hz	15 s	95 s
Data set 4	25 Hz	10 Hz	24 s	115 s
Data set 5	20 Hz	10 Hz	45 s	280 s

*

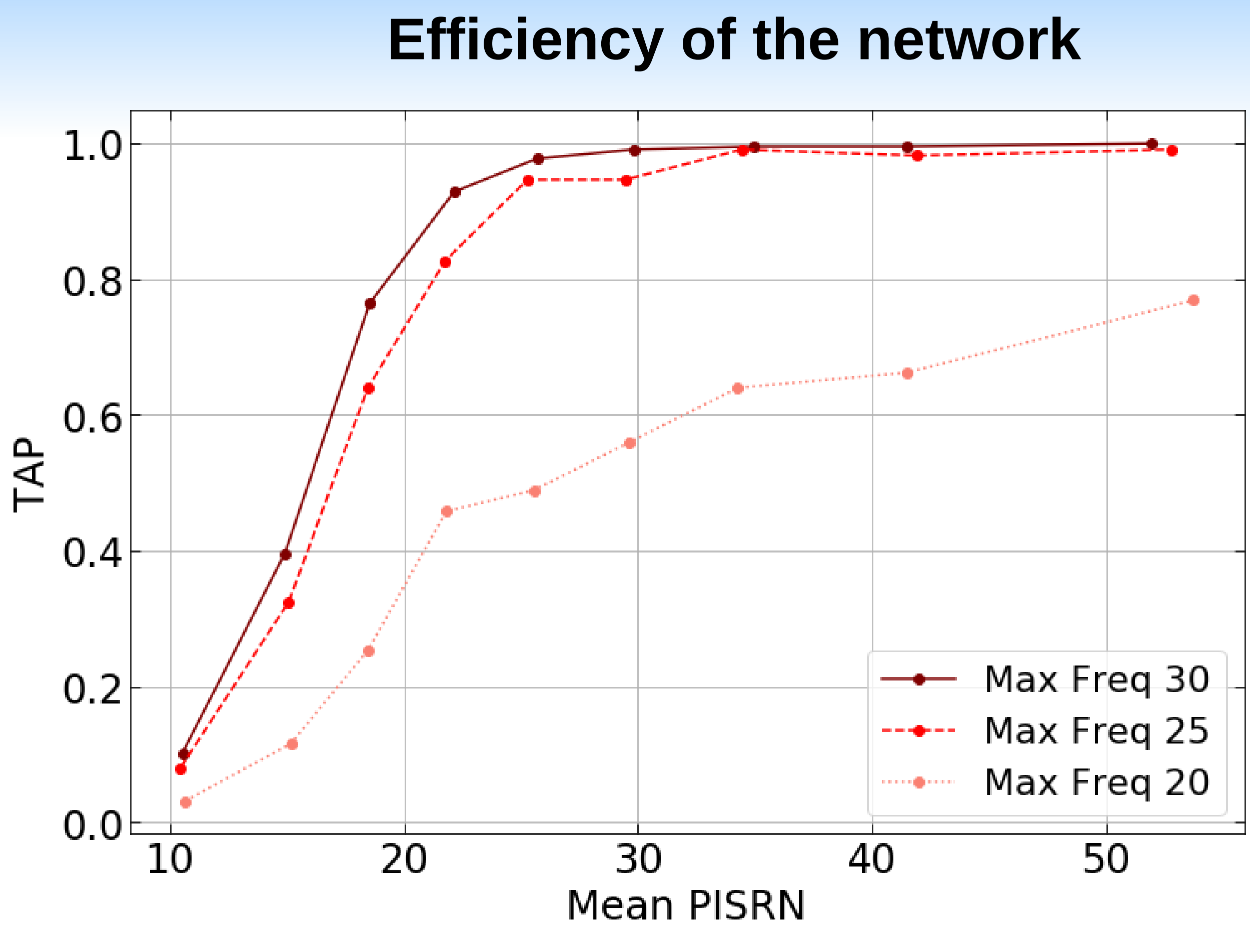
Number of sample in each data set :

- 16000 for training
- 4000 for validation
- 4400 for testing

* TBM = Time Before Merger

For Real O3 noise: training on O3a and testing on O3b

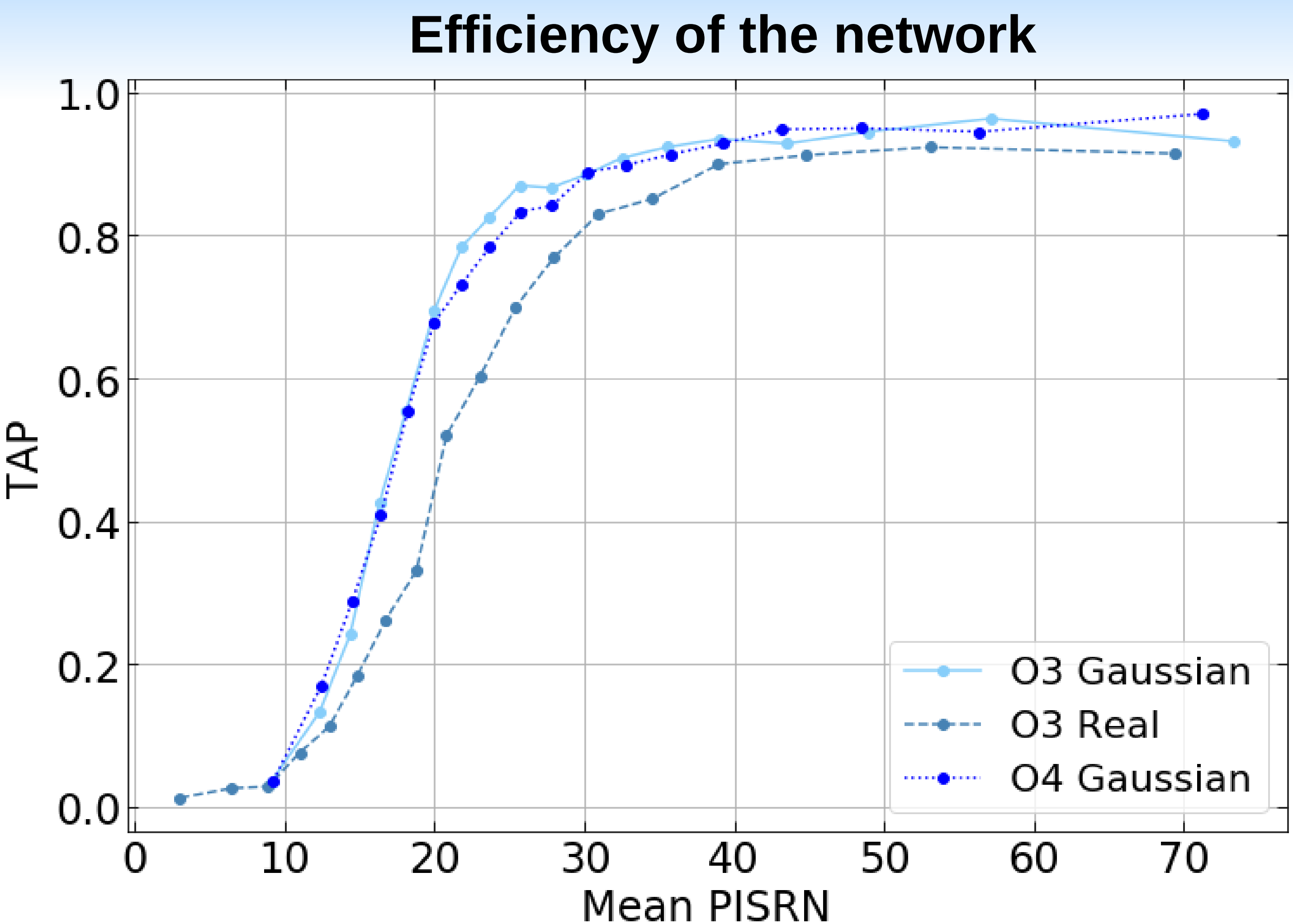
Results for O3 Gaussian noise



- For high max frequency, all the results are similar
- For a maximum frequency of 20 Hz, results degrade → Peak in the PSD
- FAP fixed at 1%
- PISNR = Partial Inspiral Signal to Noise Ratio

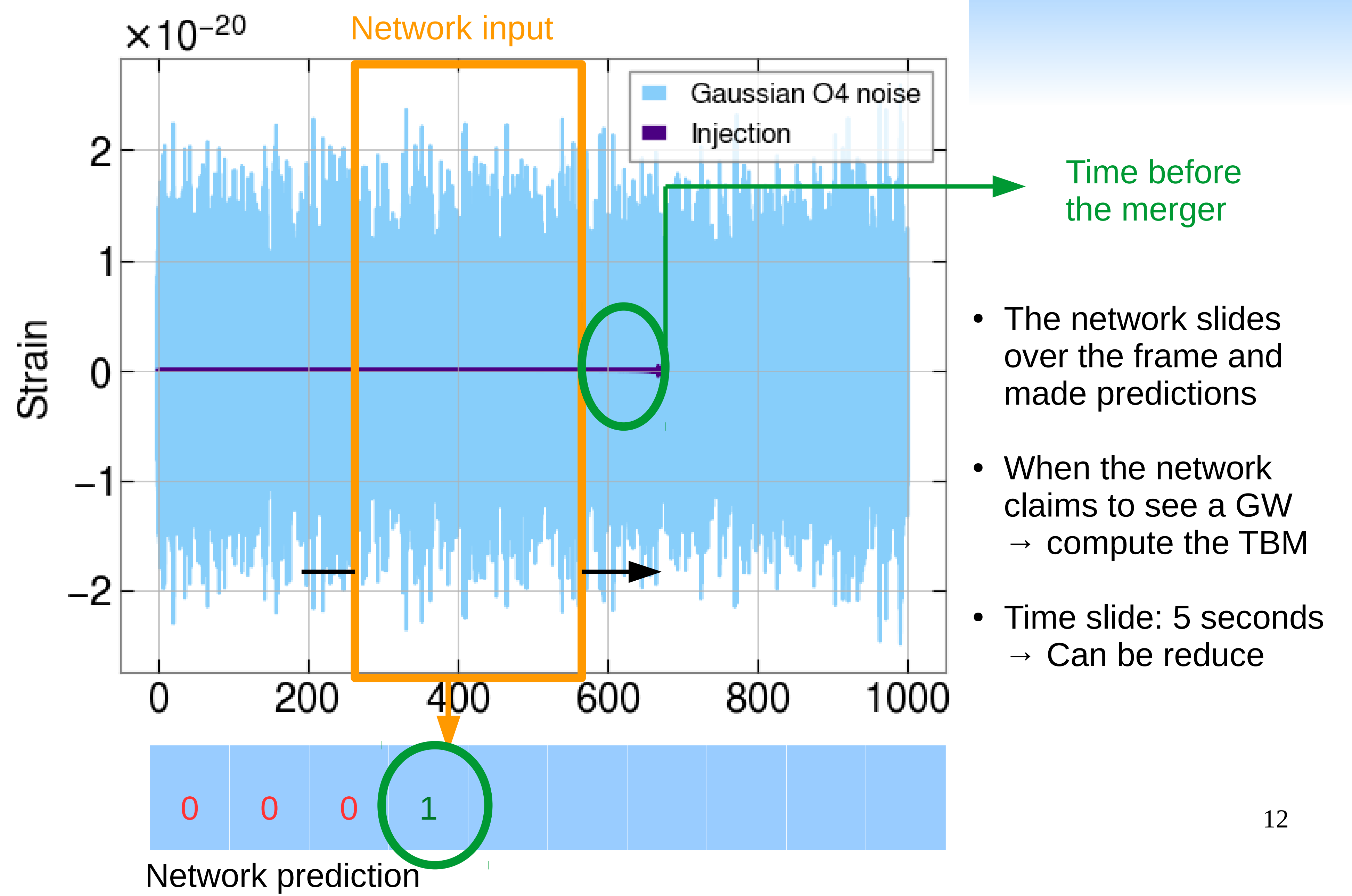
The limitation is not the network but the PSD (peak at 20 Hz)

Comparison between different noise types



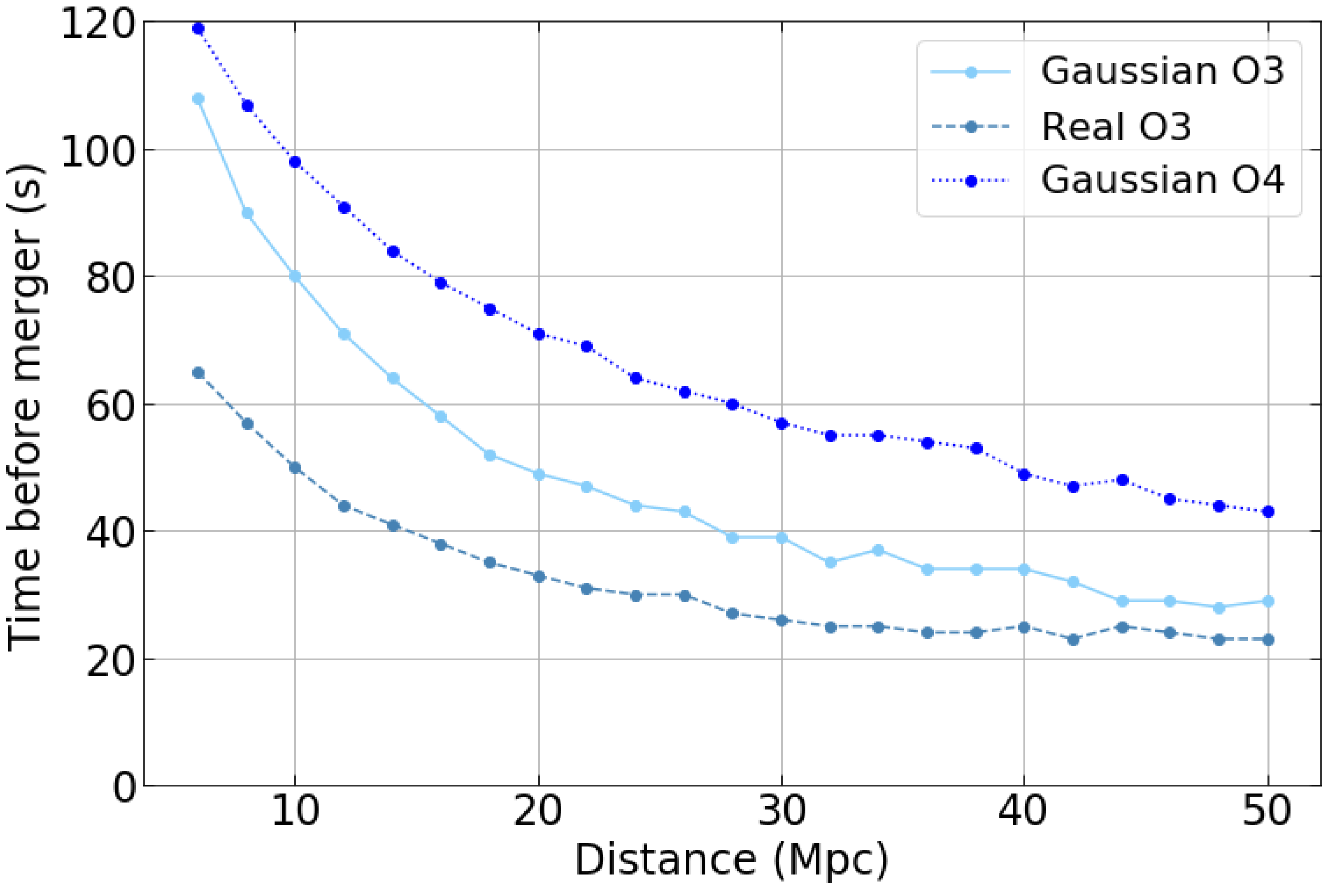
- The performances for the three noise types are similar.
- For real noise, the performance are slightly worse. This could be an effect of the non-Gaussianities in the noise

Time before merger: Sliding the network



- The network slides over the frame and made predictions
- When the network claims to see a GW → compute the TBM
- Time slide: 5 seconds → Can be reduce

Comparison between different noise types : TBM



Time before merger for the detection of a BNS at 40 Mpc (like GW170817):

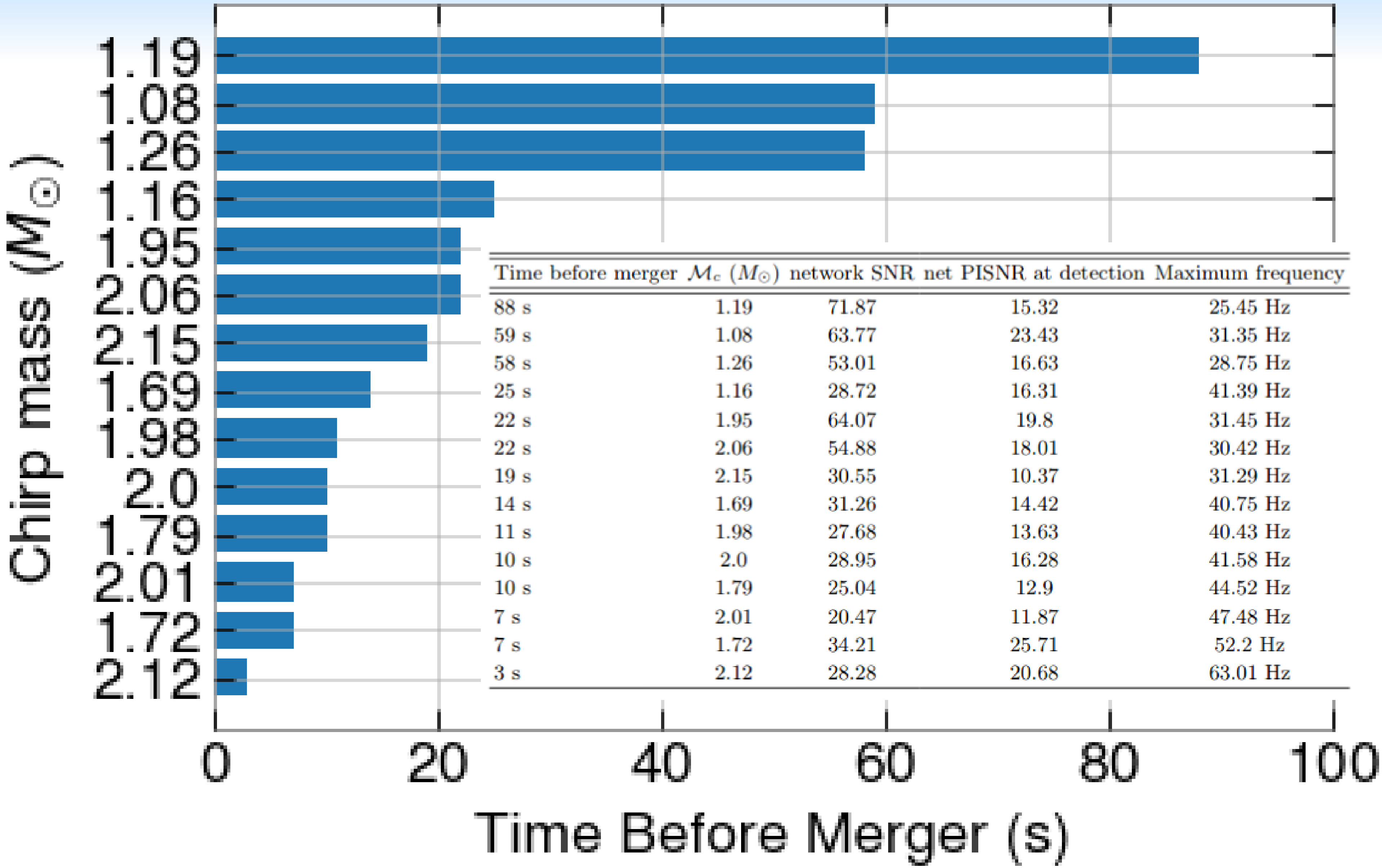
O3 Gaussian : 35 s before merger

O3 real : 25 s before merger

O4 Gaussian : 50 s before merger

Realistic O4 population

Simulation of a population of BNS for 5 years.
We injected them into O4 noise and run our pipeline.



Time before merger	$\mathcal{M}_c$ ($M_\odot$)	network SNR	net PISNR at detection	Maximum frequency
88 s	1.19	71.87	15.32	25.45 Hz
59 s	1.08	63.77	23.43	31.35 Hz
58 s	1.26	53.01	16.63	28.75 Hz
25 s	1.16	28.72	16.31	41.39 Hz
22 s	1.95	64.07	19.8	31.45 Hz
22 s	2.06	54.88	18.01	30.42 Hz
19 s	2.15	30.55	10.37	31.29 Hz
14 s	1.69	31.26	14.42	40.75 Hz
11 s	1.98	27.68	13.63	40.43 Hz
10 s	2.0	28.95	16.28	41.58 Hz
10 s	1.79	25.04	12.9	44.52 Hz
7 s	2.01	20.47	11.87	47.48 Hz
7 s	1.72	34.21	25.71	52.2 Hz
3 s	2.12	28.28	20.68	63.01 Hz

Next steps

- FAP still relatively high.
  →To reduce it we can train a CNN for each detector, and use consistency between triggers

- Test our method on the MDC

- Investigate architecture of full convolution with more complex output

- Produce a sky location map with the detected early inspiral
  → Adapt the method of A. Kolmus: arXiv:2111.00833

Conclusion

- CNN for the detection of early inspiral of BNS
- Build on our previous work: only one network, three detectors, 300 seconds input, going down in frequency
- Able to perform almost as well in real noise than in Gaussian noise
- Expected to detect some BNS in O4
- Can detect some BNS more than 1 minute before the merger
- Continue to investigate how to upgrade it

DCC of the paper: LIGO-P2200119