Rapid Identification and Classification of Eccentric Binary Black-hole mergers using Machine Learning

Yuvraj Sharma[1,2] *Collaborators:* Prayush Kumar[1] Adhrit Ravichandran[3]

[1] *International Centre for Theoretical Sciences, Bengaluru*
[2] *Indian Institute of Science Education and Research, Bhopal*
[3] *University of Massachusetts, Dartmouth*

Introduction

- Quasi-circular or eccentric binary black hole mergers typically arise in dense stellar environments like AGNs or core collapse globular clusters, featuring high mass ratios. They fall within the frequency bandwidth of ground-based detectors.
- GW190521 marked the first detection of an eccentric (quasi-circular) binary black hole merger involving massive black holes with masses of 85$M_{\odot}$ and 66$M_{\odot}$ resulting in a remnant 142$M_{\odot}$, akin to an intermediate mass black hole (IMBH).
- Upgraded detectors for future runs demonstrate an increased likelihood of detecting eccentric mergers, expanding the sensitivity band.

Motivation and Research Problem

- Bayesian Parameter Estimation (PE) faces challenges due to the absence of waveform models covering the entire Eccentricity range, hindering likelihood sampling.
- Including Eccentricity as a free parameter in PE becomes computationally intensive.

We explore Machine Learning models to identify non-eccentric events and classify the rest into three categories: non-eccentric ($e = 0$), moderately eccentric ($0 < e \leq 0.2$), and highly eccentric ($0.2 < e \leq 0.5$).

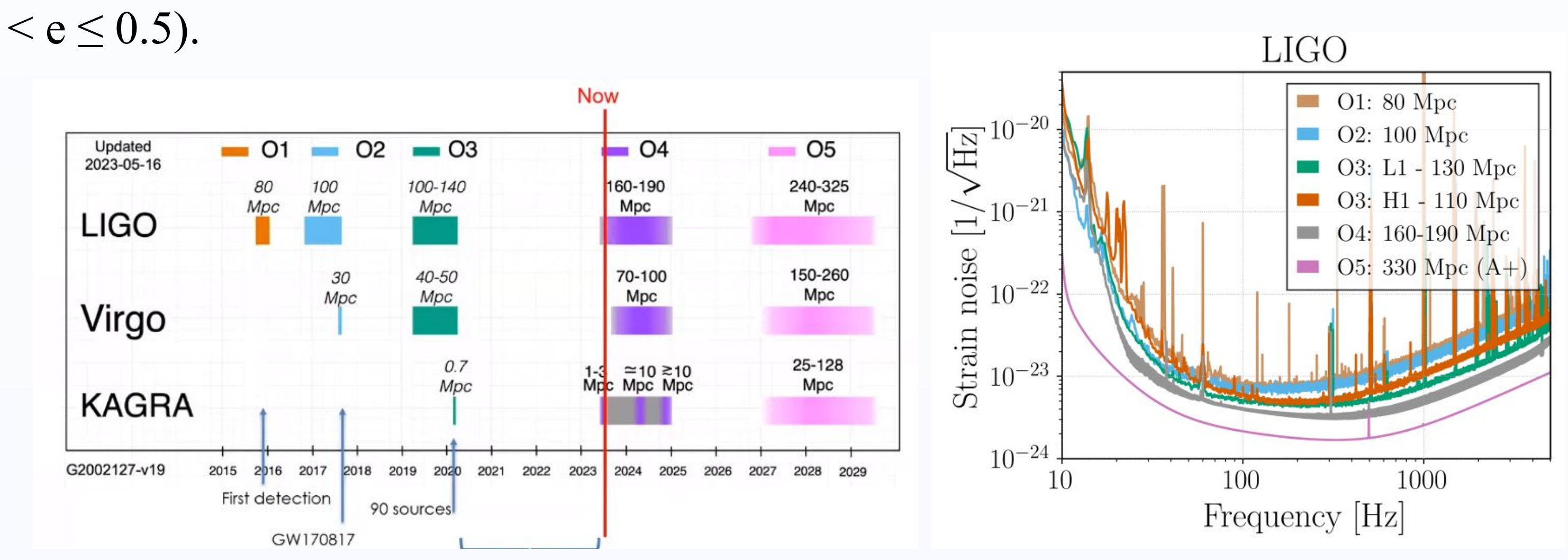


Fig. 1: Luminosity and Frequency Sensitivity range for various LIGO runs

Machine Learning Model

We use a Separable Convolutional Neural Network (SCNN) as our learning model, which is trained on time-frequency representations of the detector strain, called Q-transform spectrograms (Q-Scans). The model expects Eccentricity to modulate the shape within these Q-Scans in a detectable way for identification and classification.

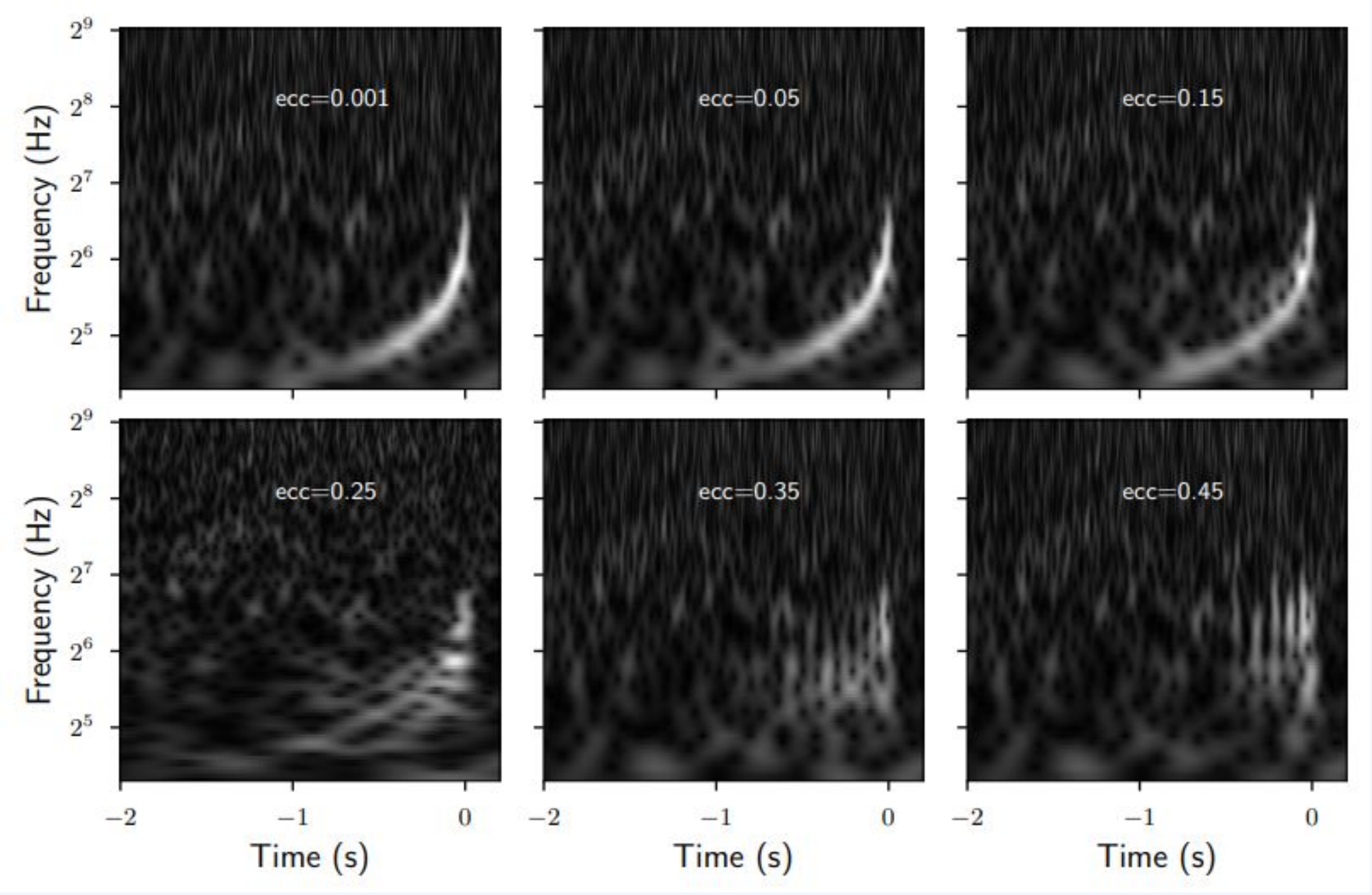


Fig. 2: Q-Scans for various Eccentricities generated by Adhrit

Dataset

The model is trained on Q-Scans of detector strain, generated from simulated Eccentric waveforms injected with whitened Gaussian noise. Dataset specifications include:

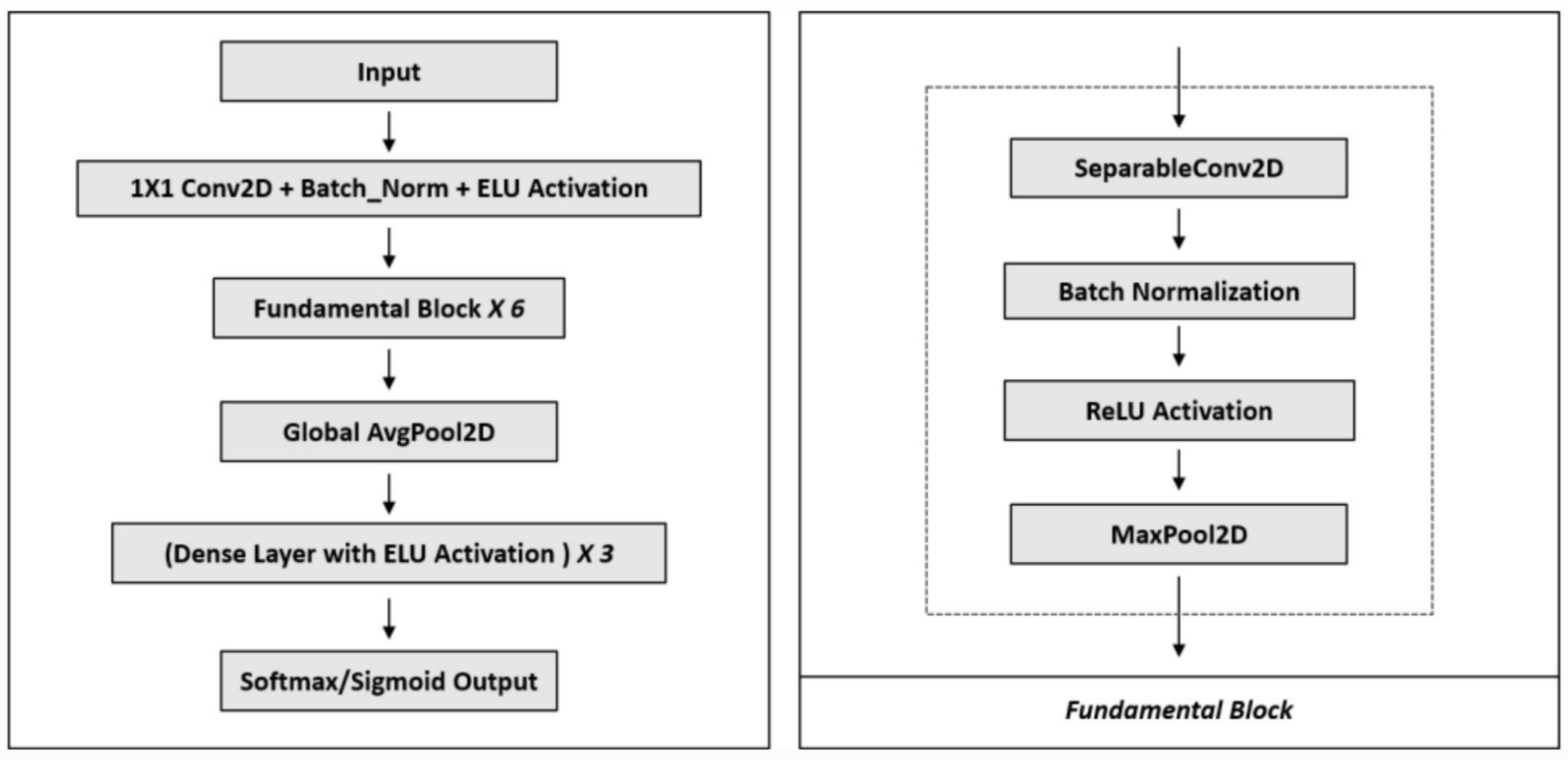
- EccentricTD approximant for template generation.
- Uniform eccentricity sampling: [0.001, 0.01] for non-eccentric, [0.01, 0.2] for moderately eccentric, [0.2, 0.5] for highly eccentric.
- Binaries' masses uniformly sampled from 10-40$M_{\odot}$
- Sky locations sampled uniformly on a two-sphere, and inclination sampled from a cosine distribution.
- Luminosity distance implicitly sampled through three-detector SNR (injection SNR) in the range 15-85 with a Universal distribution ($P(\rho) \propto \rho^{-4}$) for astrophysical sources (real) and a uniform distribution ($P(\rho) \propto$ constant) (uniform)
- Waveform and Gaussian noise sampled at 4096Hz.
- Gaussian Noise generated using the Zero-Detuning High-Power design sensitivity curve for LIGO detectors.
- Signal tapered, embedded into noise, bandpass-filtered (20Hz-512Hz), and whitened.
- Q-transforms cropped, resulting in fixed image size 256x256x3 (3rd index representing 3 detectors).

Network Architecture

- The model architecture uses Separable-Convolutional (SeparableConv2d) layers instead of the usual Convolutional layers.
- The depthwise-separable convolution factorizes a regular convolution, and this factorization drastically reduces computation and model size while maintaining accuracy
- Moving down the network, the number of channels in layers continues to increase while the height and width of the layer decrease. All convolutional layers are followed by batch normalization layers.
- The input layer takes in the 256×256×3 image, where 3 channels correspond to LIGO-Hanford (H1), LIGO-Livingston (L1) and Virgo (V1) detectors.
- To connect the SeparableConv2D to Dense layers, a Global Average Pooling is implemented. 3 Dense layers are connected, with decreasing number of nodes in the network. Dropout is implemented before the dense layer with high number of parameters to prevent any potential overfitting.

For the detection problem, the output layer is a 1 unit Softmax, where 1 corresponds to an eccentric signal (eccentricity ∈ (0.001, 0.5)) and 0 to a non-eccentric signal.

For the classification problem, the output layer is a 3 unit Softmax, where the labels are the one-hot-encodings for the 3 classes.


Input
1X1 Conv2D + Batch_Norm + ELU Activation
Fundamental Block X 6
Global AvgPool2D
(Dense Layer with ELU Activation) X 3
Softmax/Sigmoid Output

SeparableConv2D
Batch Normalization
ReLU Activation
MaxPool2D
Fundamental Block

Fig. 3: Previous Neural Network Model Architecture

Results with Gaussian Noise

- Training done for 30 epochs (detection) and 100 epochs (classification) with accuracies converging to 1 and 0.98 respectively.
- Testing resulted in showing real distribution produced better performing NNs.
- Testing the SCNN model on waveforms generated from TEOBResumS, to check if the patterns learned by the model were generic to eccentric waveform models as well or just for EccentricTD.
- For the detection problem, the accuracy for the best performing NN (real trained for 20 epochs) was 0.97, the loss and accuracy for both converged after around 15 epochs.
- For the classification problem, the accuracy for the best performing NN (real trained for 94 epochs) was 0.92, the loss and accuracy for both converged after around 50 epochs.

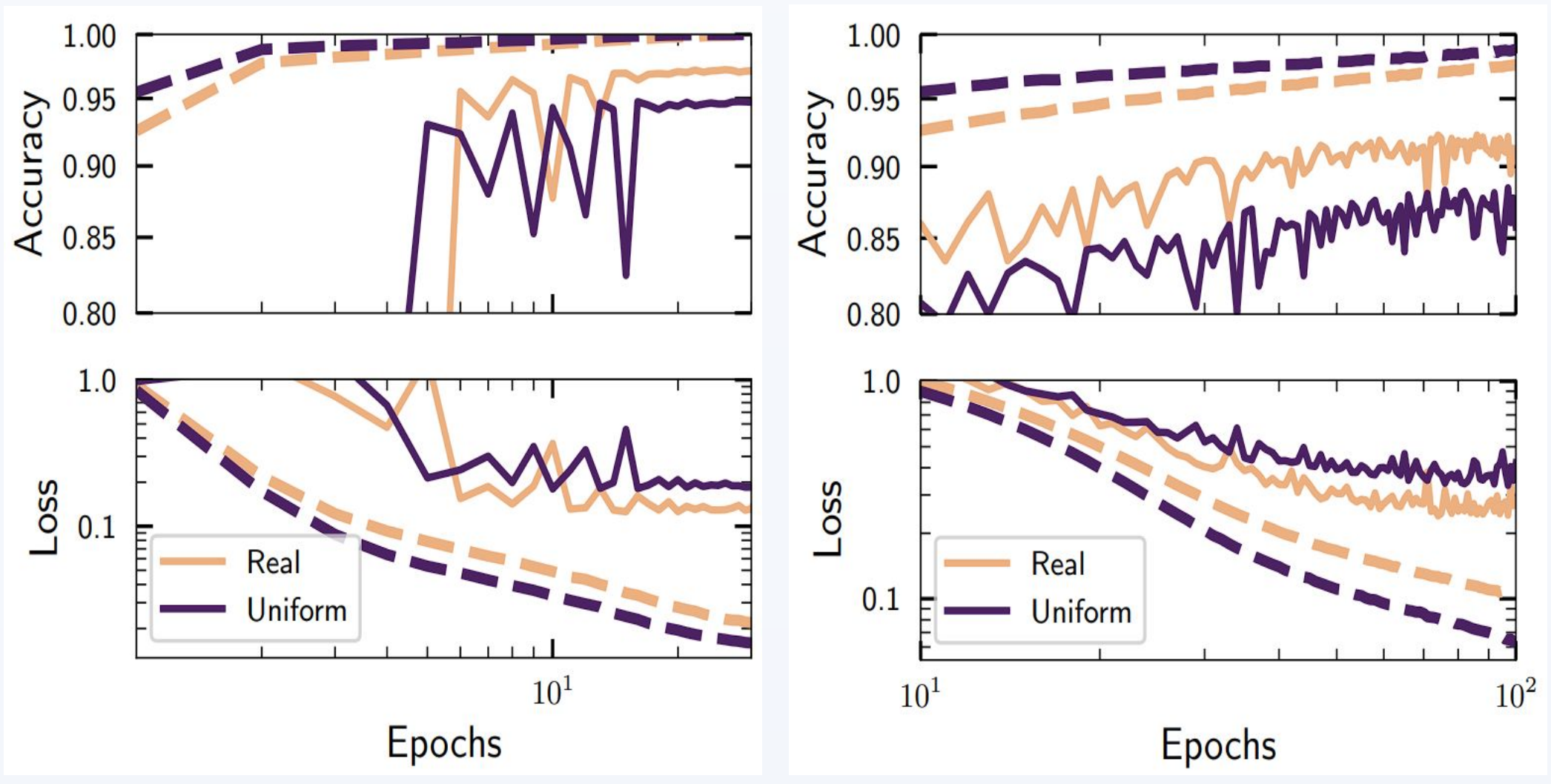


Fig. 4: Accuracy and Loss curves for Testing data

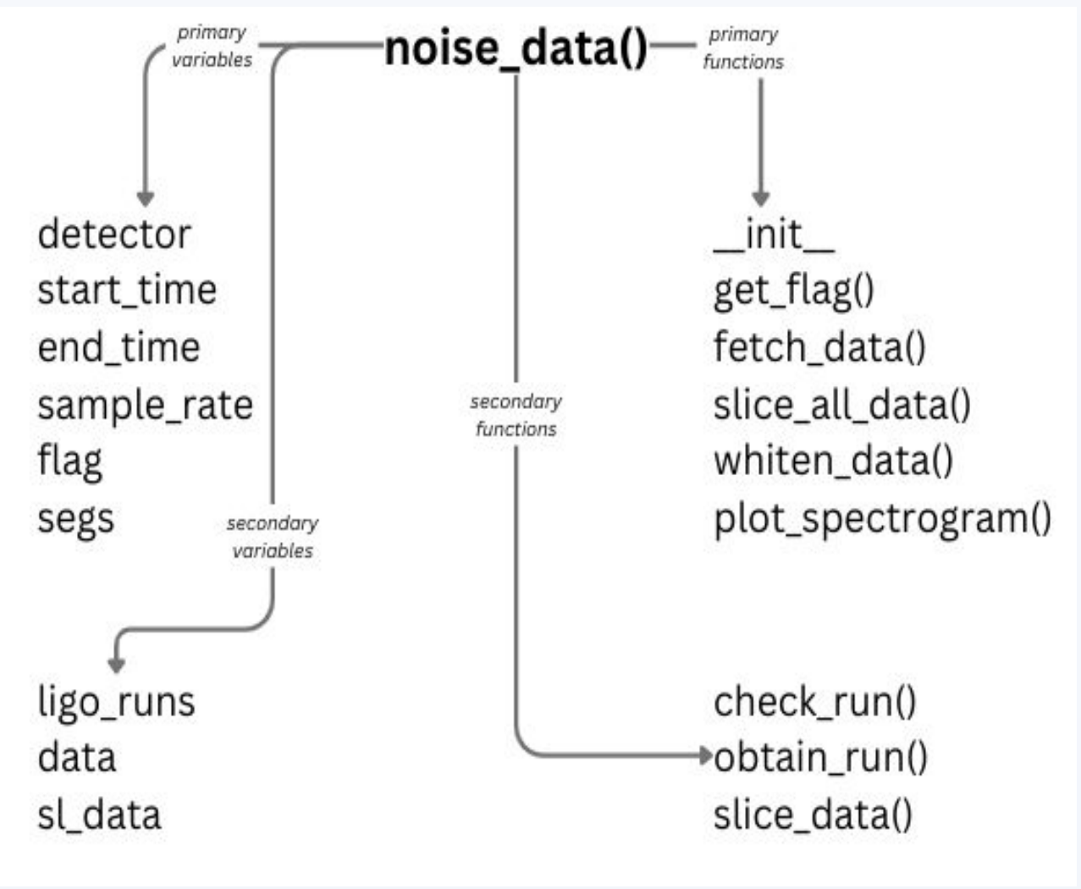

noise_data()
primary variables
primary functions
secondary variables
secondary functions
detector
start_time
end_time
sample_rate
flag
segs
ligo_runs
data
sl_data
__init__
get_flag()
fetch_data()
slice_all_data()
whiten_data()
plot_spectrogram()
check_run()
obtain_run()
slice_data()

Fig. 5: Outline for the current pipeline

Training with Real-detector Noise

- Currently training the model and re-tune it to work on a dataset injected with real-detector noise from the LIGO-Virgo detectors. To obtain noise data and further manipulate it easily I built a new data generation pipeline gw_data.
- The pipeline currently employs classes and its functions additionally from GWPy.
- The main aim of the data generation pipeline is to automate the process and make it more flexible for anyone to use the LIGO-Virgo noise data, for further analysis, recoloring, injection, etc.
- Additional functionalities to handle data for multiple detectors at once.
- Currently most of the studies being conducted have been on simulated and colored gaussian noise but this will help obtain real detector noise.
- The current model when ran with real detector noise, experienced a drop in its accuracy to 0.81 for the classification problem, shows the architecture needs to be restructured.

Future Work

- Using other model architectures like Transformers or Normalizing flows to build a better Machine Learning model for the detection and classification while training on real noise.
- Adding more parameters for the detection problem like Black-hole spin as currently we are assuming them as non-spinning Black-holes and Orbital precession.

References:

1. Ravichandran, A., Vijaykumar, A., Kapadia, S. J., Kumar, P. (2023). Rapid Identification and Classification of Eccentric Gravitational Wave Inspirals with Machine Learning. ArXiv./abs/2302.00666
2. Luc Blanchet, "Gravitational Radiation from PostNewtonian Sources and Inspiralling Compact Binaries," Living Rev. Rel. 17, 2 (2014), arXiv:1310.1528 [gr-qc].
3. Fran¸cois Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition (2017) pp. 1251–1258.
4. B. P. Abbott et al. (LIGO Scientific, Virgo), "Search for Eccentric Binary Black Hole Mergers with Advanced LIGO and Advanced Virgo during their First and Second Observing Runs," Astrophys. J. 883, 149 (2019), arXiv:1907.09384 [astro-ph.HE].
5. Sashwat Tanay, Maria Haney, and Achamveedu Gopakumar, "Frequency and time-domain inspiral templates for comparable mass compact binaries in eccentric orbits," Physical Review D 93, 064031 (2016).
6. D. M. Macleod et al, SoftwareX, 13, 100657 (2021) doi.org/10.5281/zenodo.8409995

Acknowledgements: I would like to thank ICTS-TIFR for providing this platform and computational resources for this project.

GitHub: YuviAndBeyond

10th International Conference on Gravitation and Cosmology

yuvrajasharma.acads@gmail.com