Machine Learning for the Tune Estimation in the LHC

Data Science and Machine Learning Workshop
ICALEPCS 2021

Leander Grech

15th October, 2021

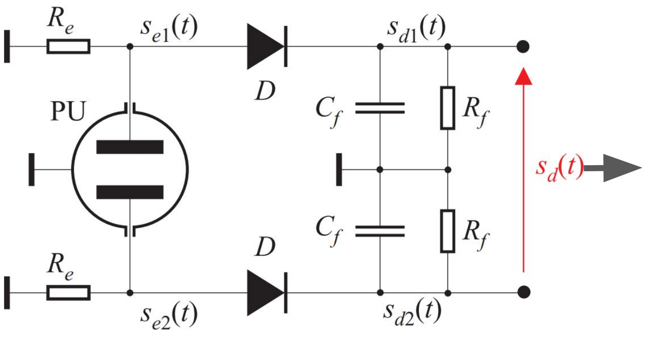
L-Università ta' Malta

CERN

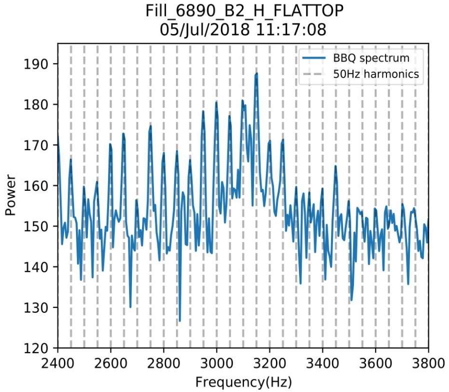
How is tune estimation done?

- **Tune (Q)** is the frequency of betatron oscillations
- The **Base-Band Q (BBQ)** system is responsible for estimating the tune in the LHC
  - BBQ systems obtains time series data sampled at revolution frequency → **ACQ data**
  - Fast Fourier Transform (FFT) 2048 samples of ACQ data → **FFT spectrum** (contains 1024 frequency bins)
  - EMA per frequency bin in FFT spectrum over time → **EMA spectrum**
  - Median and moving average filters over EMA spectrum → **Filtered spectrum**
  - Estimate peak position → **Q-coarse**
  - Check Q-coarse on EMA spectrum → **Q-refined**
  - Perform 3-point Gaussian fit centered around Q-refined → **Q**

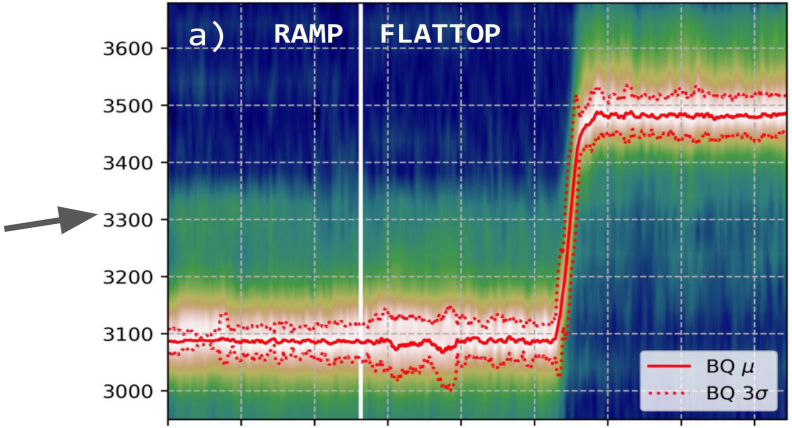
BQ Algorithm



BBQ samples beam with Pick-Up (PU) to get $s_e(t)$
Applies Direct Diode Detection (DDD) principle to get **ACQ data**



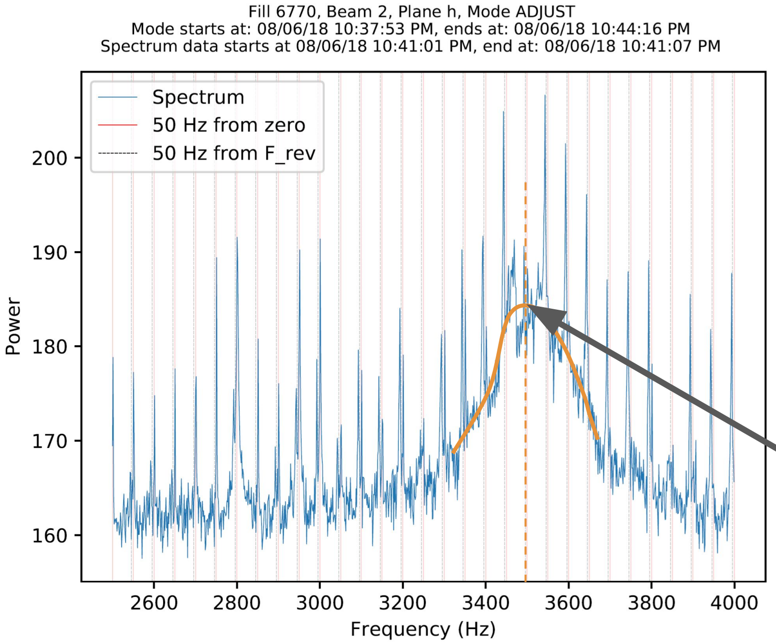
FFT on ACQ data to get **FFT spectrum**



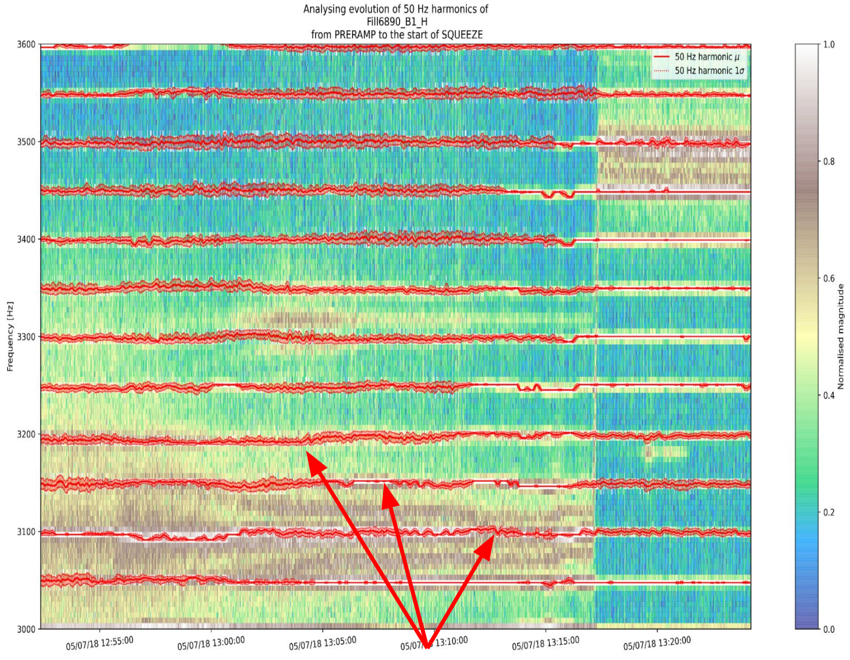
X-axis is time and Y--axis is frequency [Hz]
BQ tune estimation evolution (red) superimposed on smoothed EMA spectra.

Problems from the start

- Since 1st LHC operation, 50 Hz noise harmonics were observed in BBQ spectrum
- Creates spectra like this one:



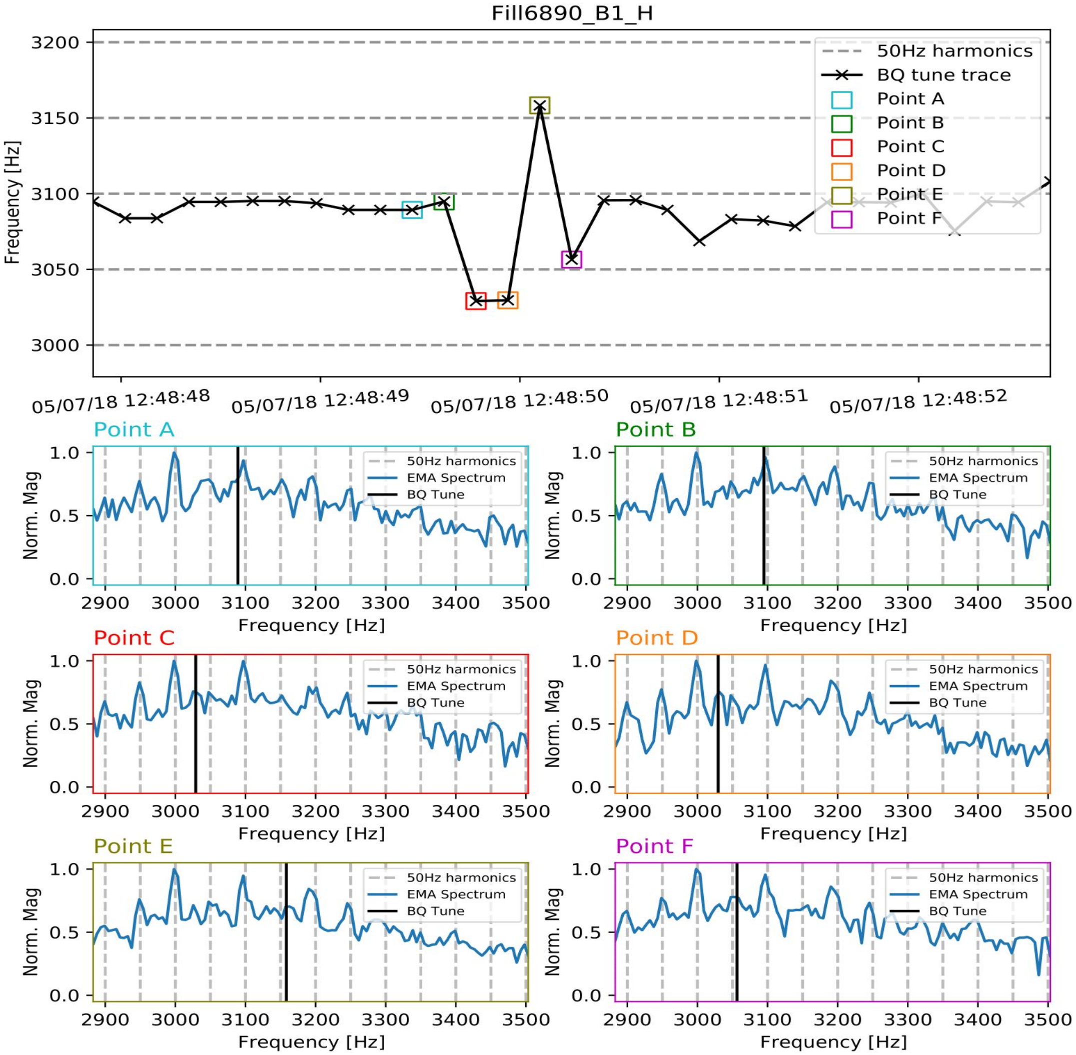
Main peak's center frequency is the tune



Tracking the evolution of 50 Hz harmonics

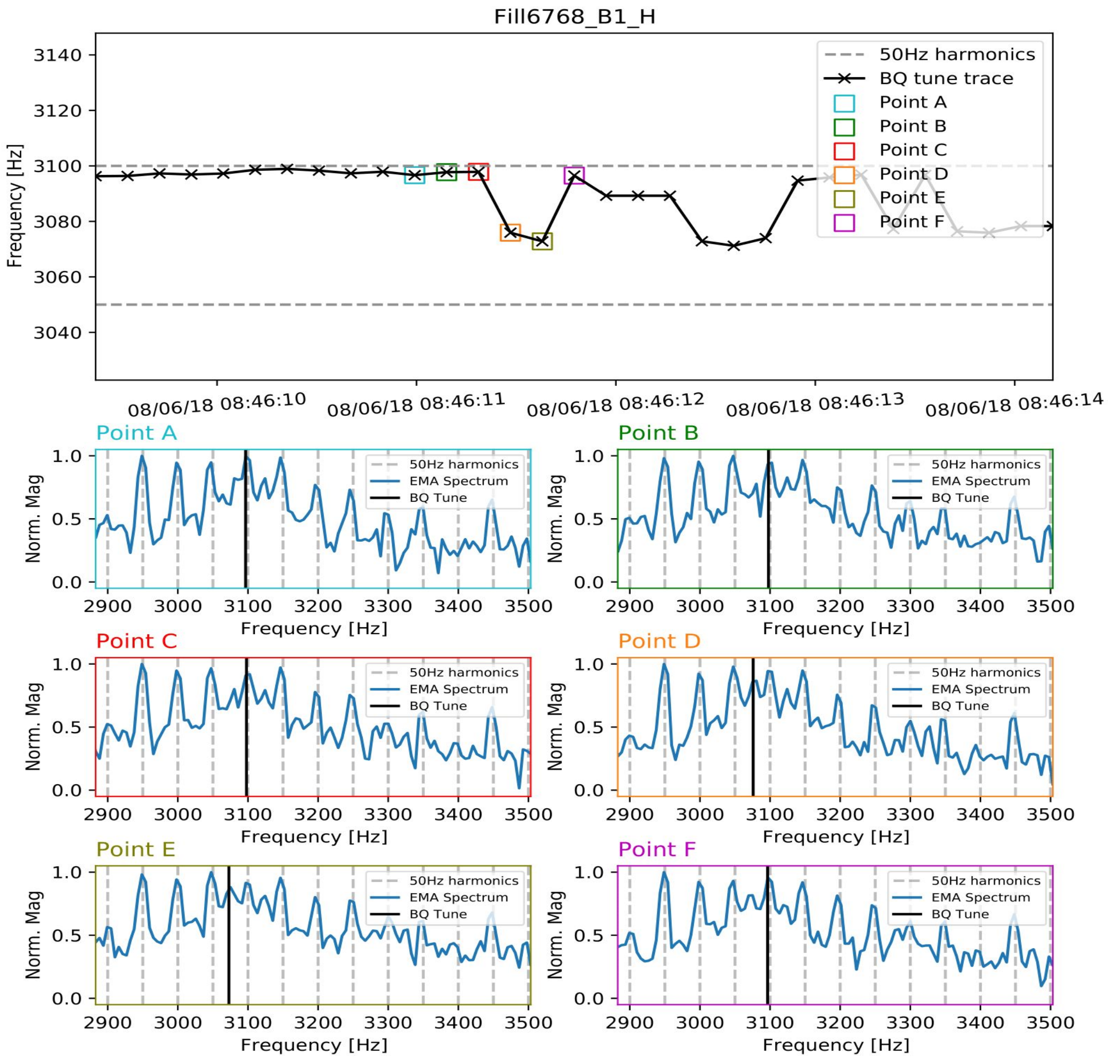
So, what was happening with BQ algorithm?

Tune estimates exhibited large variance



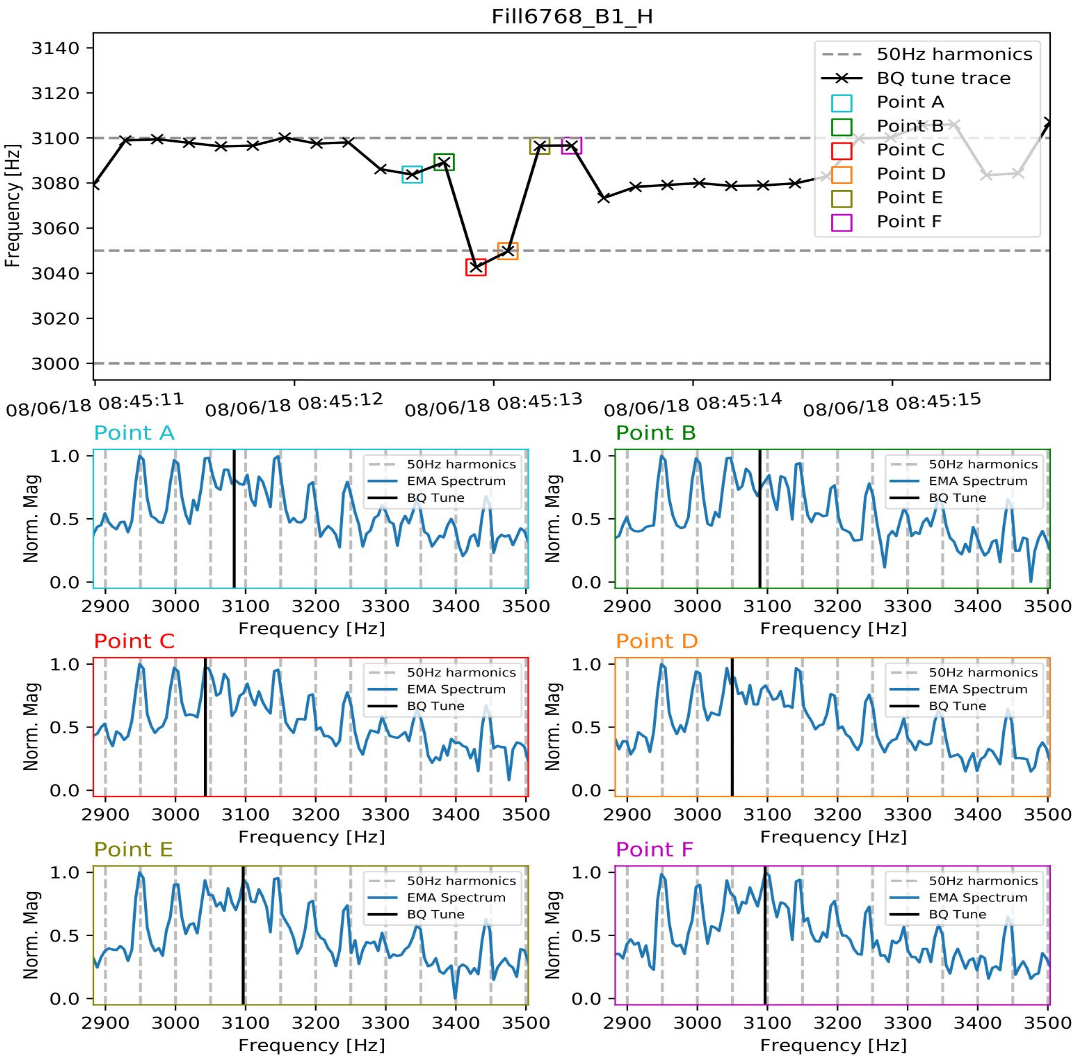
So, what was happening with BQ algorithm?

Tune estimates had a tendency to get stuck to harmonic frequencies



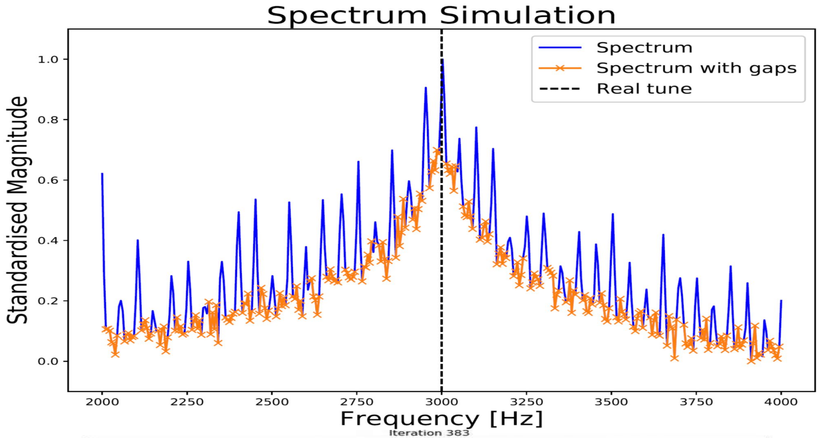
So, what was happening with BQ algorithm?

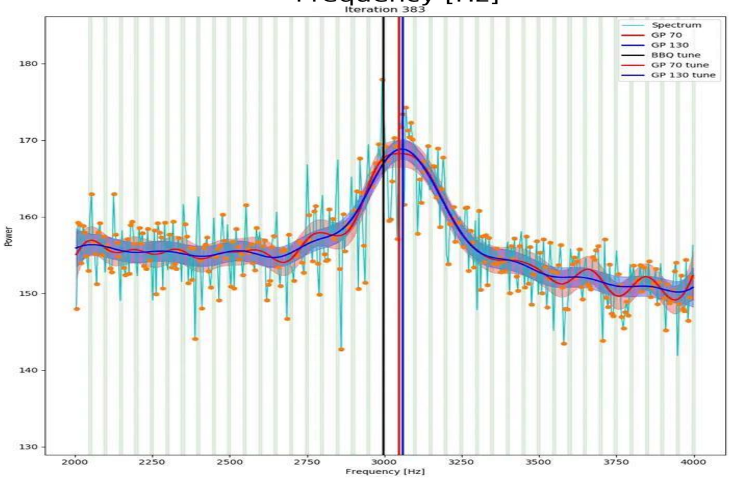
Tune estimates also jumped from one harmonic to the other



Can different algorithmic approaches achieve better results?

- A series of alternative algorithms were attempted that take into consideration the 50 Hz harmonics[1]
  - Since their frequency location is quite stable, we can explicitly remove them
  - This created spectra with gaps
  - Gaussian Processes (GP), Weighted Moving Average (WMA) and Polynomial fitting (POLY) were used on spectra with gaps to obtain tune
- Performance was somewhat improved but…
  - WMA and POLY were sensitive to hyperparameter tuning
  - GP was very computationally expensive (inverting 100x100 matrix per tune estimate)
  - Tuning harmonic removal is also tricky!`

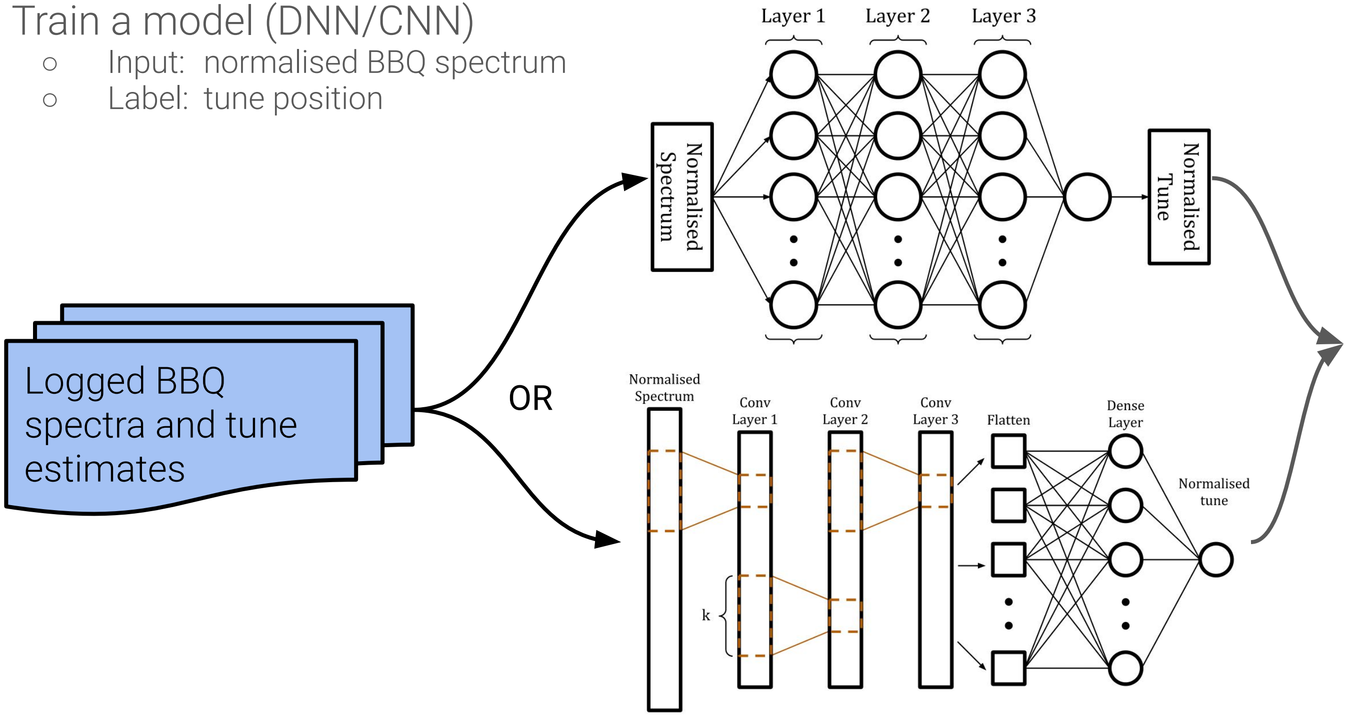




[1]L. Grech *et al.*, "An Alternative Processing Algorithm for the Tune Measurement System in the LHC," in *Proceedings of the 9th International Beam Instrumentation Conference (IBIC 2020)*, Virtual, 2020.

Naive approach

- Obtain dataset made up of logged BBQ spectra and BBQ tune estimates
- Train a model (DNN/CNN)
    - Input: normalised BBQ spectrum
    - Label: tune position



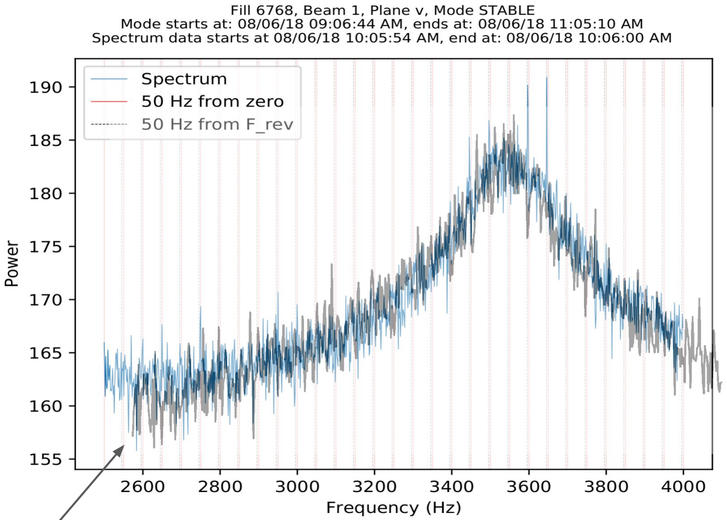
Correct tune predictions, right?

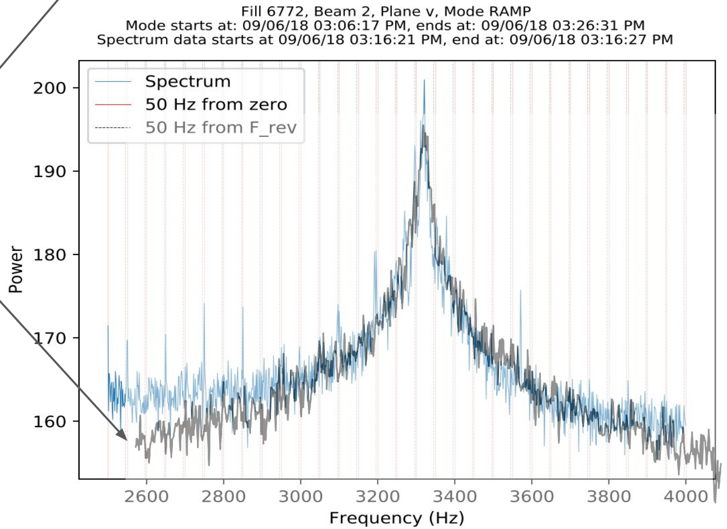
Naive approach I is too naive

- Logged BBQ spectra do not have correct labels
  - Logged tune estimates come from BQ algorithm!
  - We can simulate spectra as done in [1]
- 2^{nd} order system simulation
- Some examples of real spectra ...
- ... and simulated spectra on top

$$G(\omega) = \frac{\omega_{res}^2}{\sqrt{(2\omega\omega_{res}\zeta)^2 + (\omega_{res}^2 - \omega^2)^2}} + \mathcal{N}(0, \sigma^2)$$

$$\omega_{res}^{true} = \omega_{res}\sqrt{1 - 2\zeta^2}$$

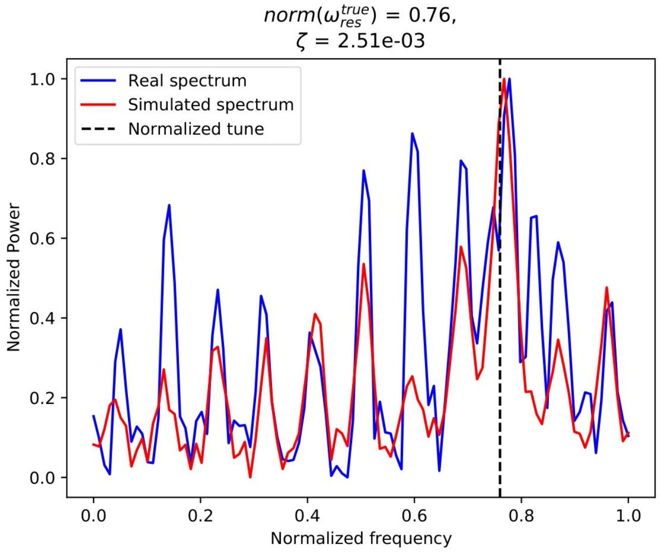




[1] L. Grech *et al.*, "An Alternative Processing Algorithm for the Tune Measurement System in the LHC," in *Proceedings of the 9th International Beam Instrumentation Conference (IBIC 2020)*, Virtual, 2020.

Simple approach I

- Create a simulated dataset:
  - Monte Carlo simulation - create different spectra
  - Inject 50 Hz harmonics with random amplitudes
  - Apply digital filter to widen harmonics
- Train DNNs and 1D CNNs with different architectures



	Layer 1	**Layer 2**	**Layer 3**	**#[1]**
ML#0	150	50	10	23,221
ML#1	300	100	20	62,441
ML#2	500	250	50	188,351

[1] Number of trainable parameters.

DNNs

	Layer 1			**Layer 2**			**Layer 3**			**Dense**	**#[4]**
	f[1]	**k[2]**	**s[3]**	**f**	**k**	**s**	**f**	**k**	**s**		
ML#3	32	3	3	16	3	3	8	3	3	20	2,753
ML#4	32	3	1	16	3	1	8	3	1	20	18,113
ML#5	64	3	3	32	3	1	16	3	1	20	18,905
ML#6	128	3	3	64	3	3	16	3	3	20	29,561
ML#7	64	3	1	32	3	1	16	3	1	20	40,025

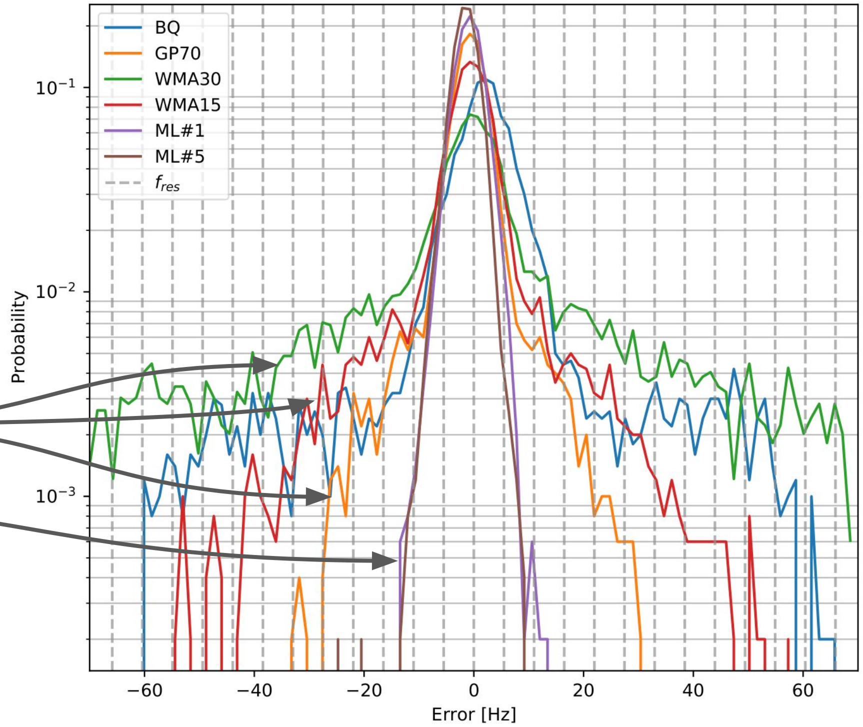
[1] Number of filters.
[2] Kernel size of convolution.
[3] Stride length, shift size of kernel.
[4] Number of trainable parameters.

1D CNNs

Simple approach II

- Evaluation
  - Compare predicted tunes to ground truth tune from simulation
  - Error = Prediction - Groundtruth
  - Do it over 5000 simulated spectra

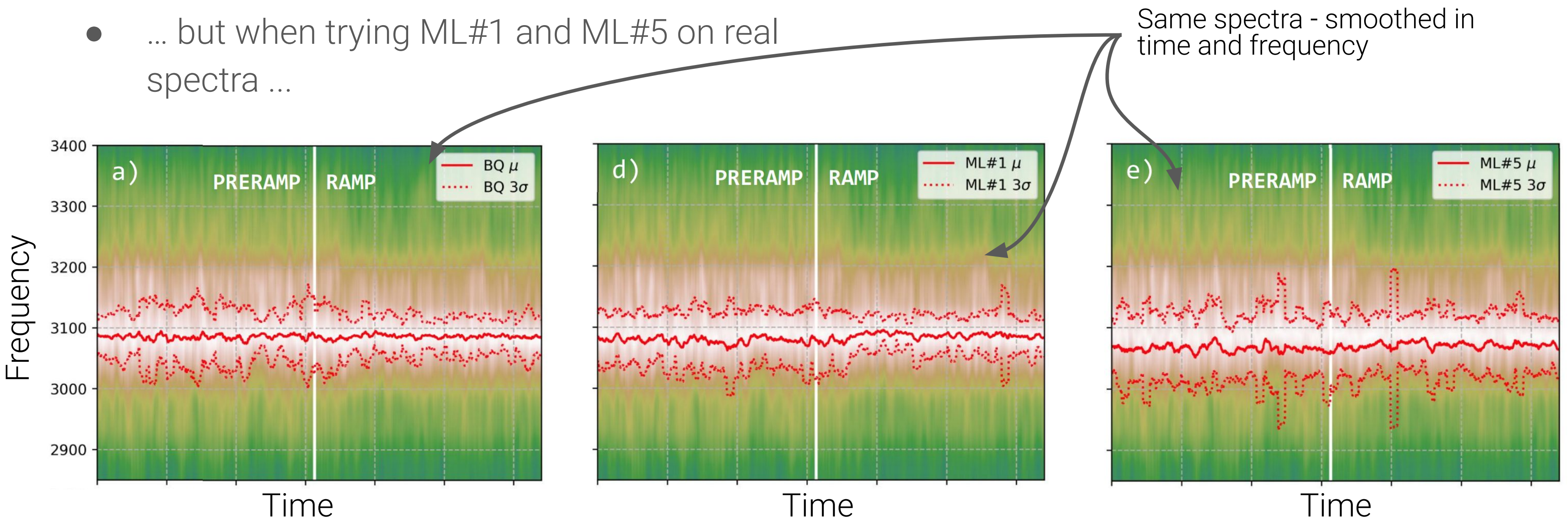
- Error probability distribution of errors from different models and algorithms
  - BQ - original tune estimation algorithm
  - GP and WMA - alternative algorithms[1]
  - ML #1 - best DNN model
  - ML #5 - best CNN model



[1] L. Grech *et al.*, "An Alternative Processing Algorithm for the Tune Measurement System in the LHC," in *Proceedings of the 9th International Beam Instrumentation Conference (IBIC 2020)*, Virtual, 2020.

Simple approach III

- Evaluation on simulated spectra show promising results ...
- ... but when trying ML#1 and ML#5 on real spectra ...



- ... the results are similar to BQ

Simple Approach IV

- BQ tune estimates were turning off the Tune Feedback (QFB)
    - Too unstable
- QFB had a stability metric:

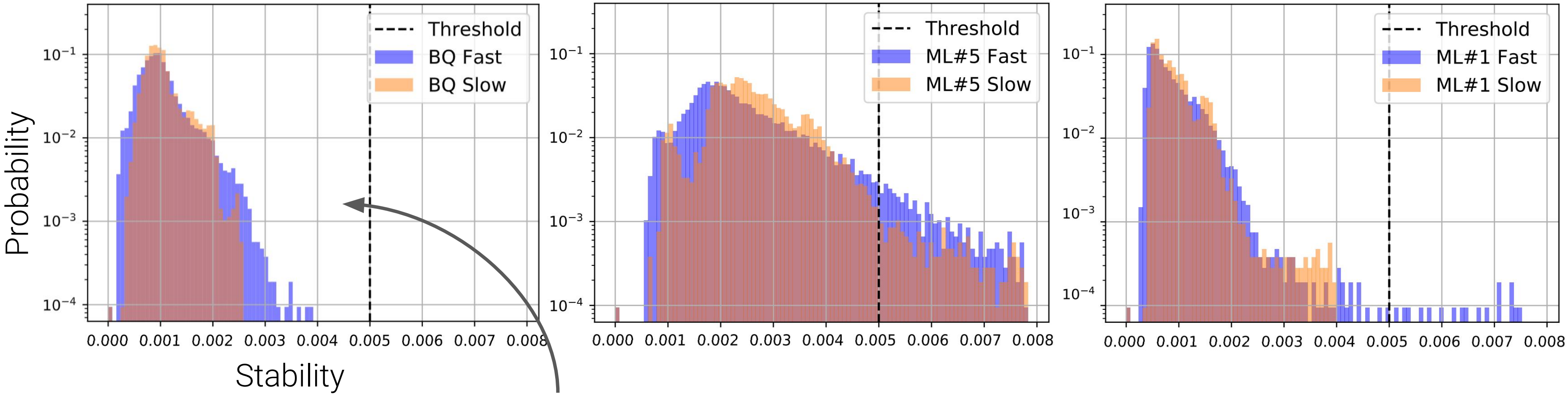
$$\begin{aligned} \Delta q_t &= q_t - q_{t-1} \\ S_t &= S_{t-1} * (1-\alpha) + \Delta q_t * \alpha \end{aligned}$$

1 Fast - 1 s

1 Slow- 10 s

Spectra from one fill were used to obtain these stability plots

(Different fill from the one used for SimGAN)

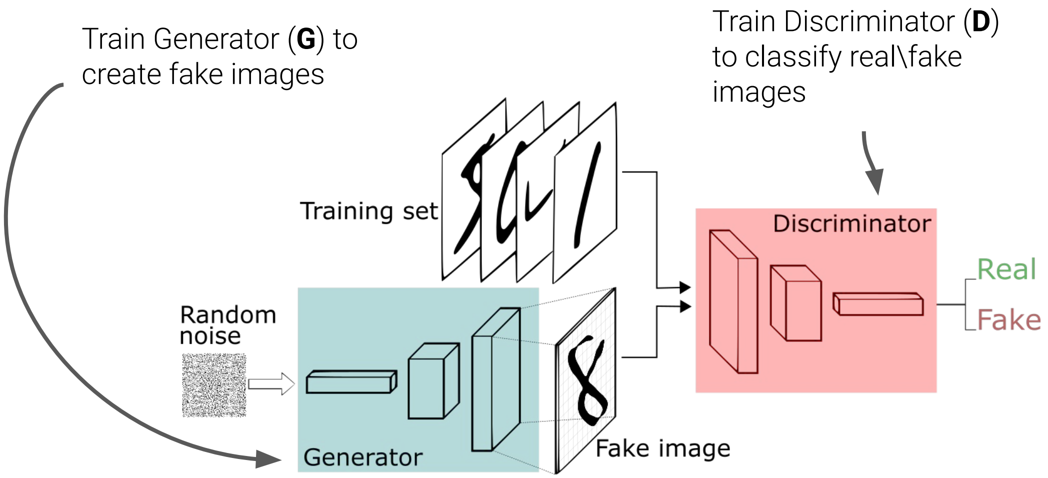


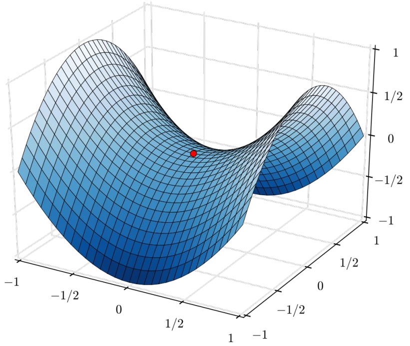
If both stabilities are beyond this point, QFB switches off

Does not look good for Simple Approach...

Improving the dataset I

- It was known that simulated spectra do not look perfectly real
- This affected training
  - Models overfit to simulations
- We can make the training dataset better by using a type of Generative Adversarial Network (GAN)





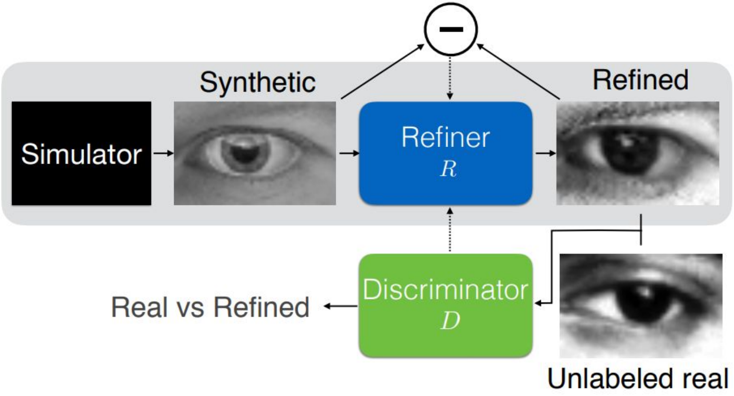
Minimax problem

G maximises loss of **D**

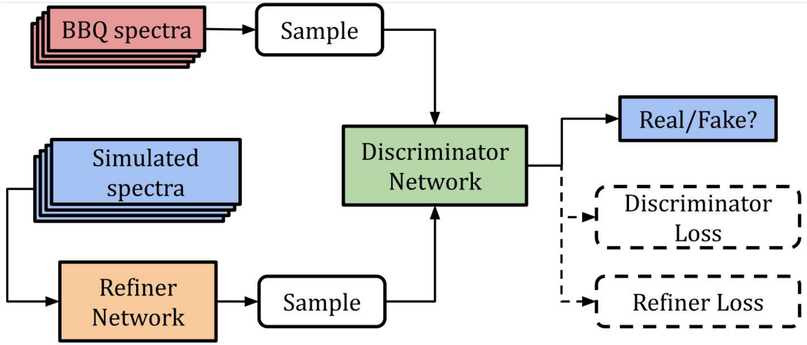
D minimises class. loss

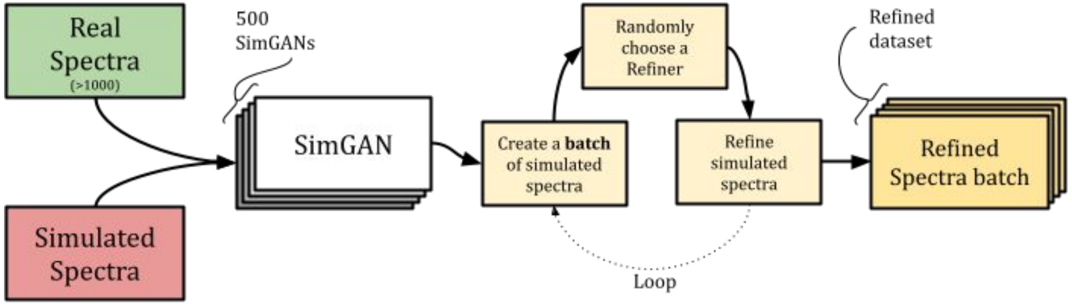
Improving the dataset II

- Variant of GAN called **SimGAN**[1]
- Applying it to simulated and real BBQ spectra



- Slight problem
  - 1 Refiner is not enough! It can produce similar looking artefacts
- Solution
  - Train 500 SimGANs with slightly different parameters

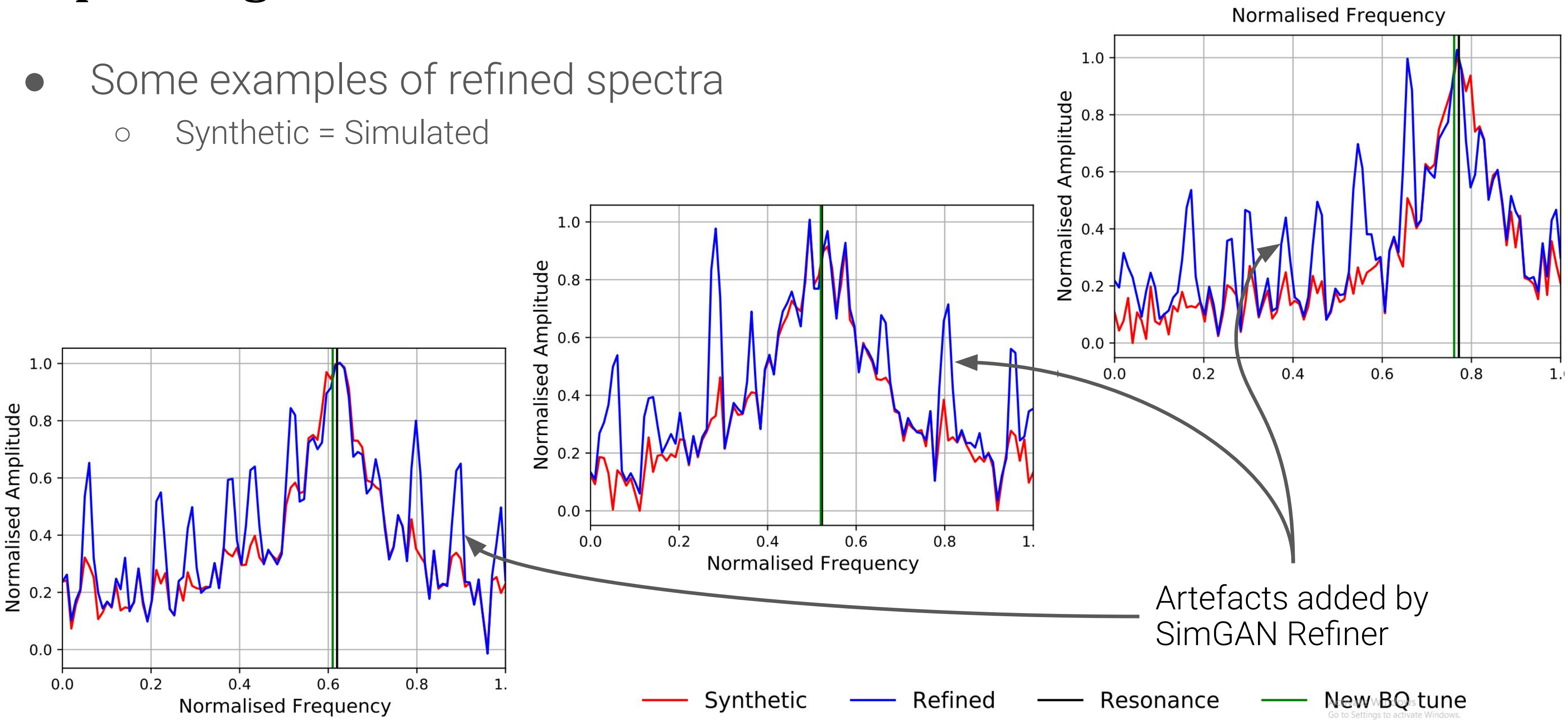




[1]A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in Proceedings of the IEEE conference on computer vision and pattern recognition, Hawai'i Convention Center, Honolulu, Hawaii, United States, 2017, pp. 2107–2116.

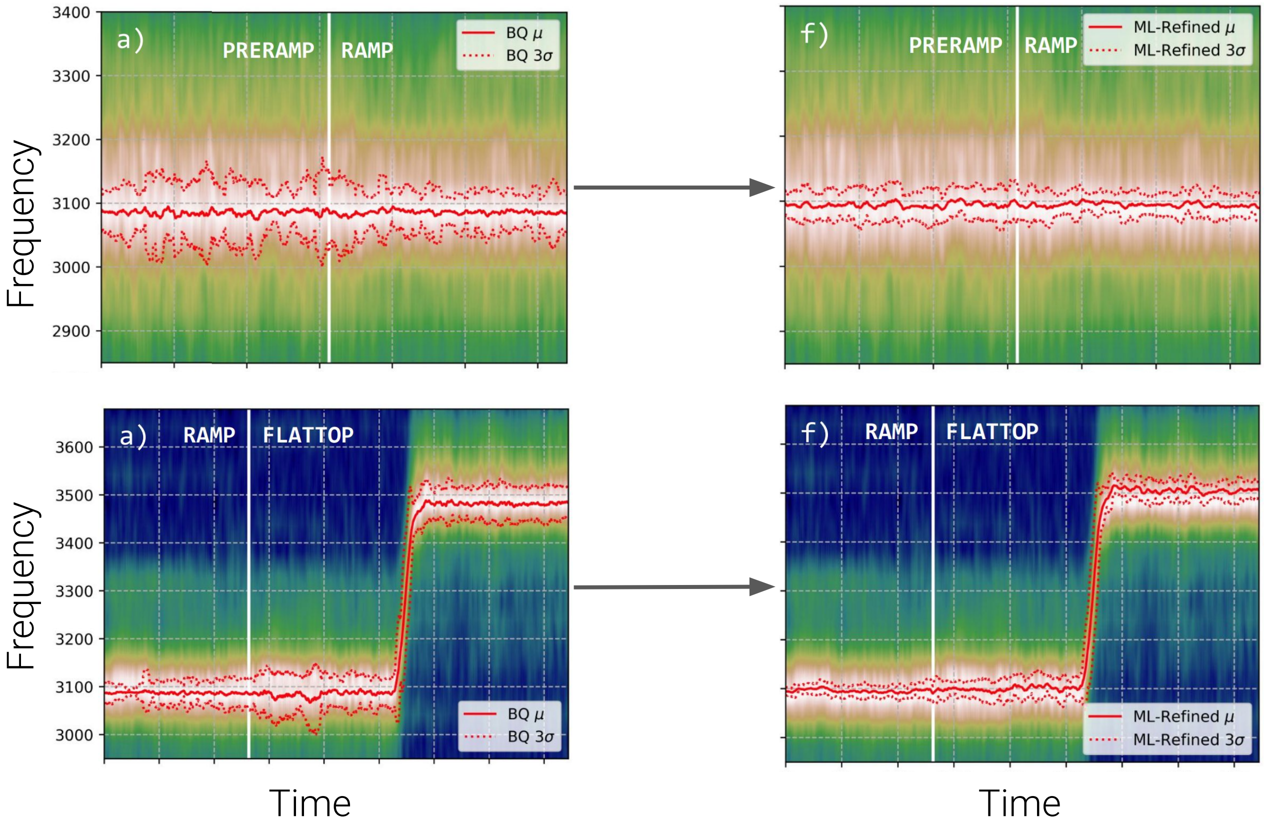
Improving the dataset III

- Some examples of refined spectra
  - Synthetic = Simulated



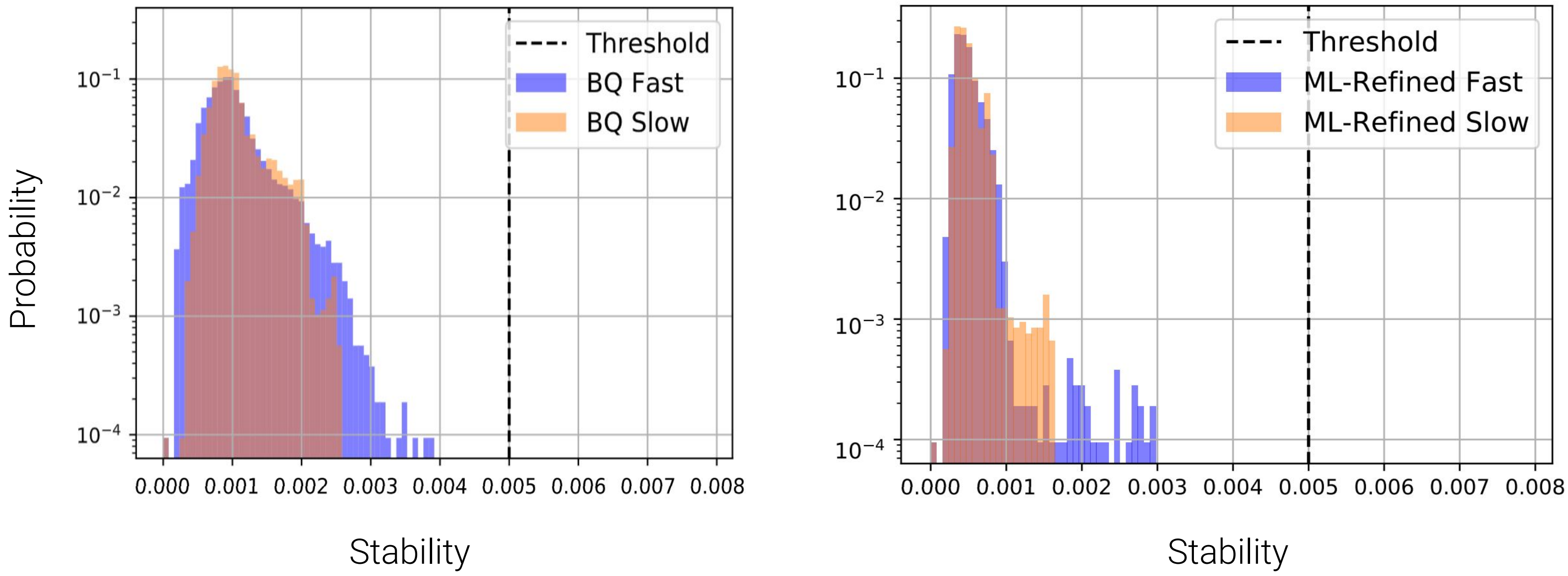
Refined Approach I

- Train the best model architecture attempted in Simple Approach (ML#1) with refined dataset - **ML-Refined**



Refined Approach II

- What about the QFB stability metric?



- ML-Refined gave the most stable estimates from all tune estimation systems attempted

Conclusion

- Train a NN to estimate the tune from a BBQ spectrum
- Naive approach
  - Use logged data as is
  - Too simple
- Simple approach
  - Use simulated spectra
  - Overfit to simulation
- Refined approach
  - Refine simulated spectra with SimGANs
  - Produced good results

Publication

- L. Grech, G. Valentino, and D. Alves, “A Machine Learning Approach for the Tune Estimation in the LHC,” *Information*, vol. 12, no. 5, p. 197, Apr. 2021.