Radio signal modulation classification

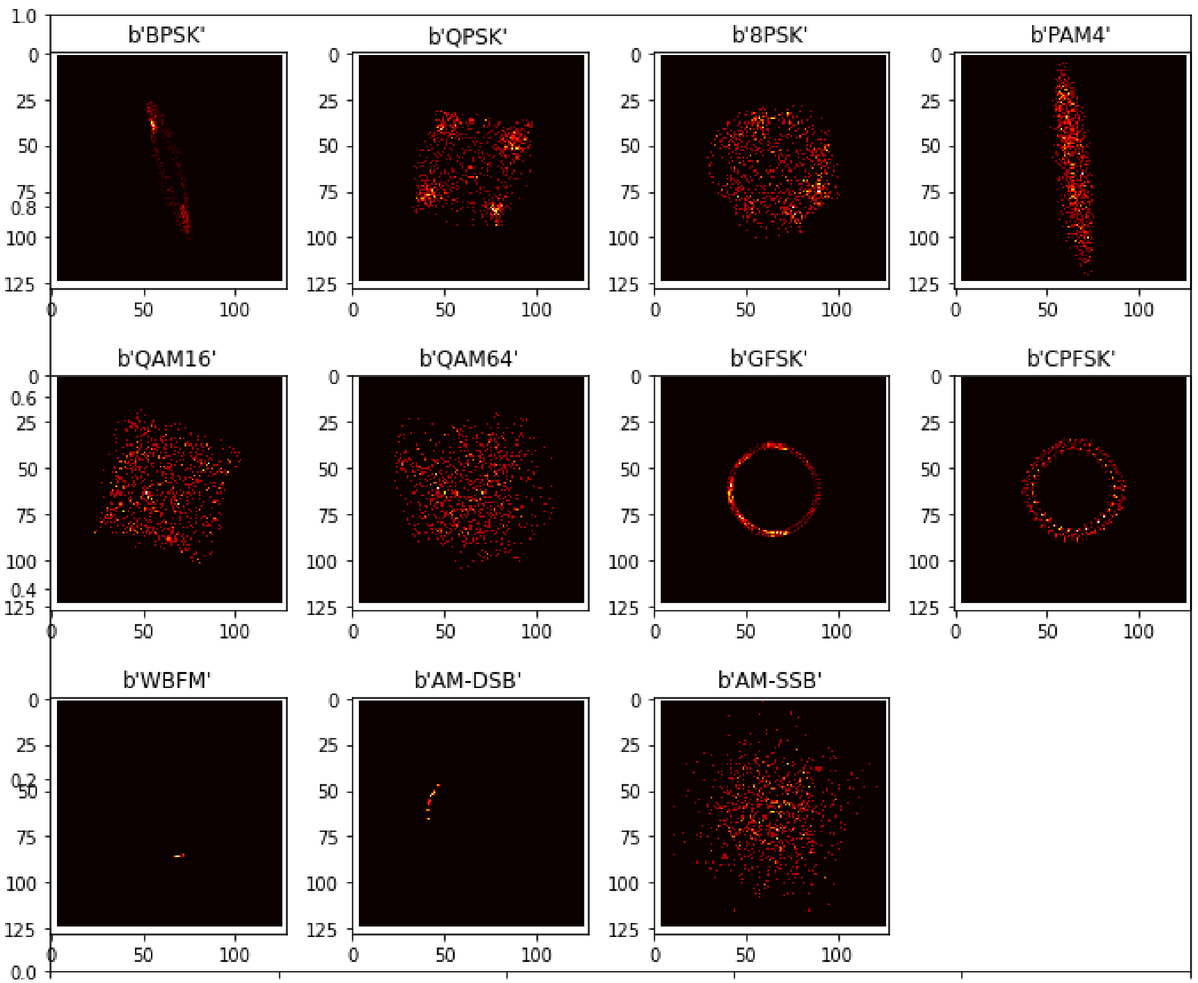
Updates 28 Jul

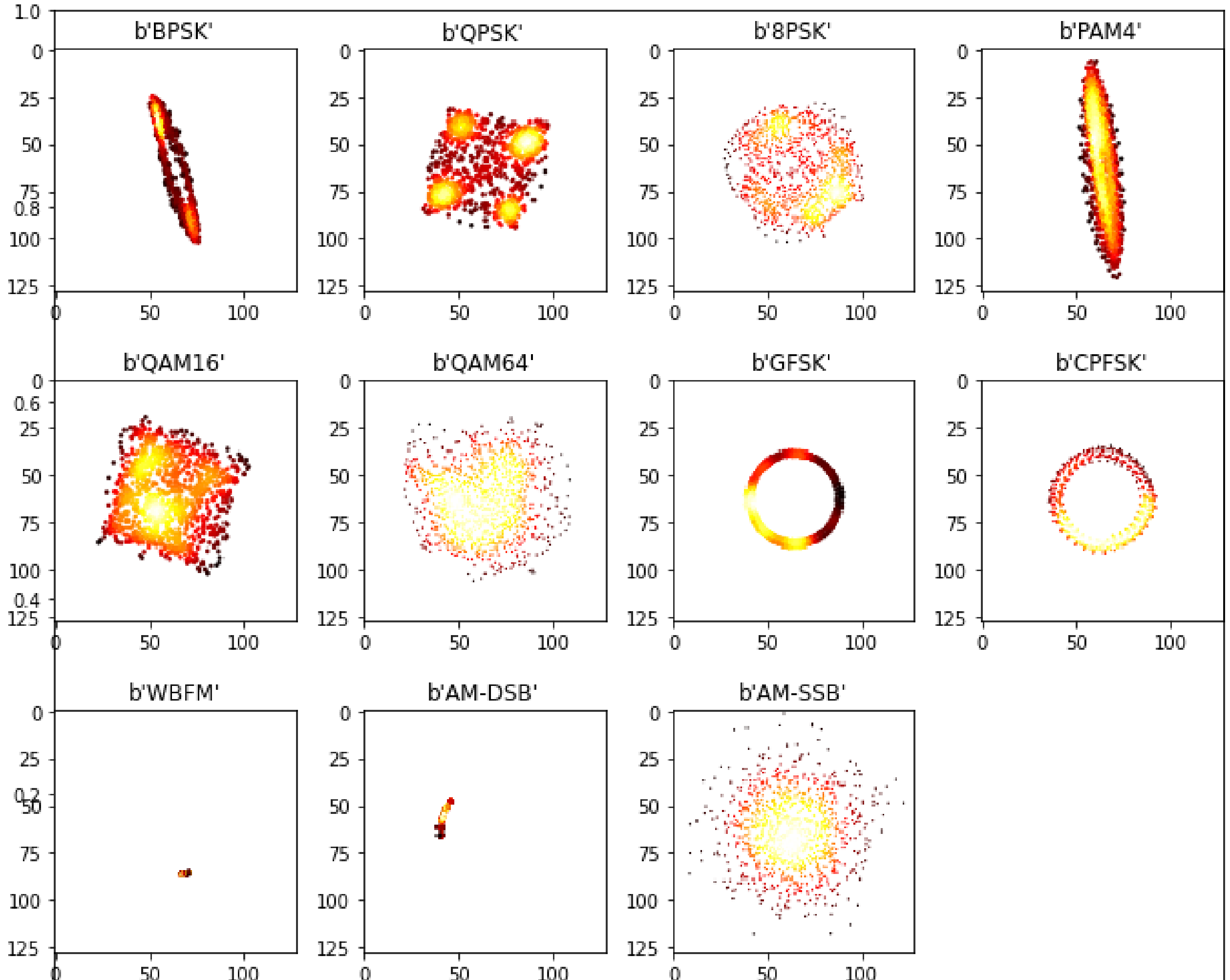
Summary

1. Constellation classification and visualization
2. LSTM, and comparison with methods tried
3. 2018 DeepSig dataset and Matlab dataset

1. Representation of constellation diagram

- (1) Histogram (counts number of points in each bin) instead of (2) directly plotting points and applying gaussian kernel density estimation (kde)

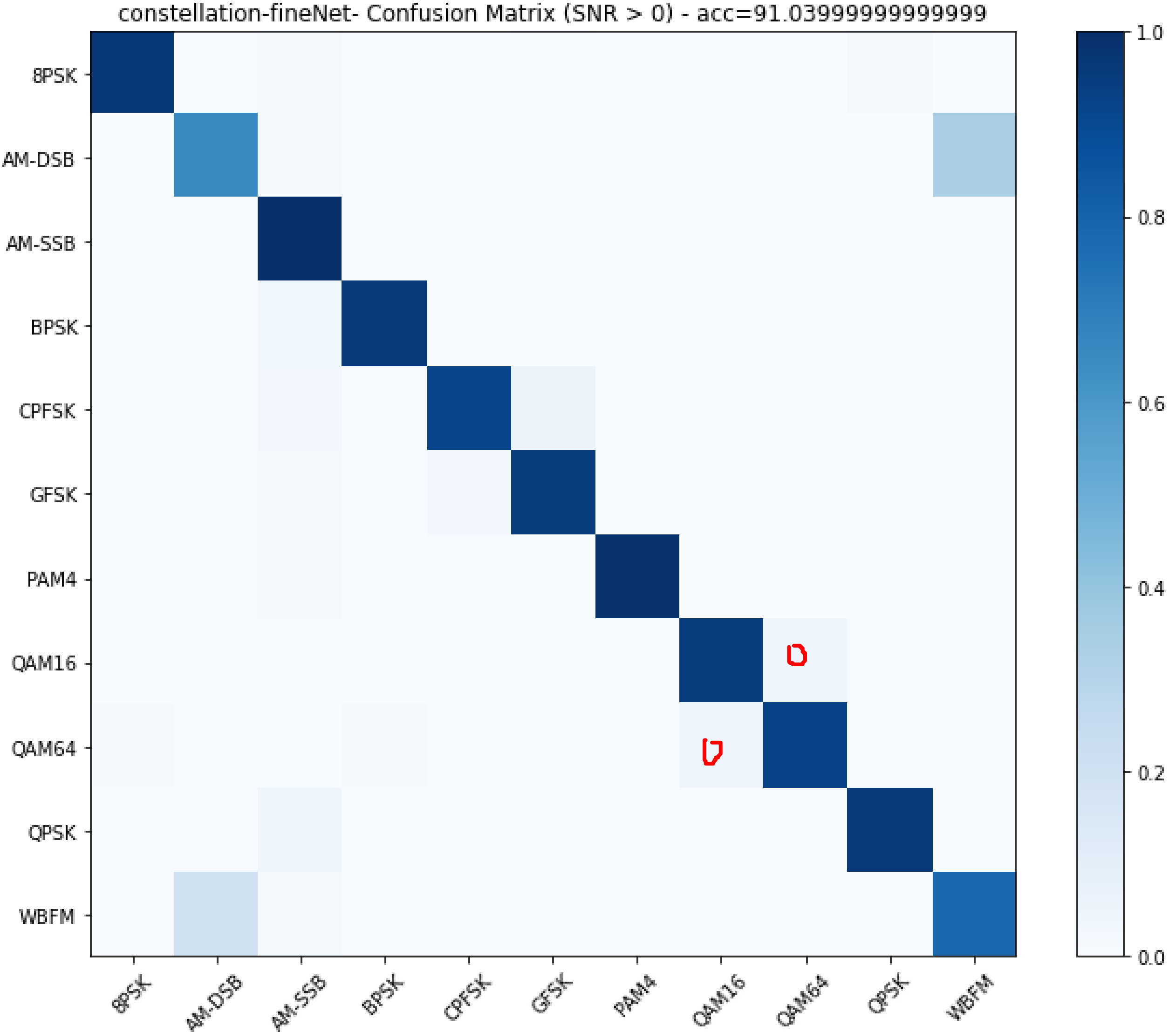




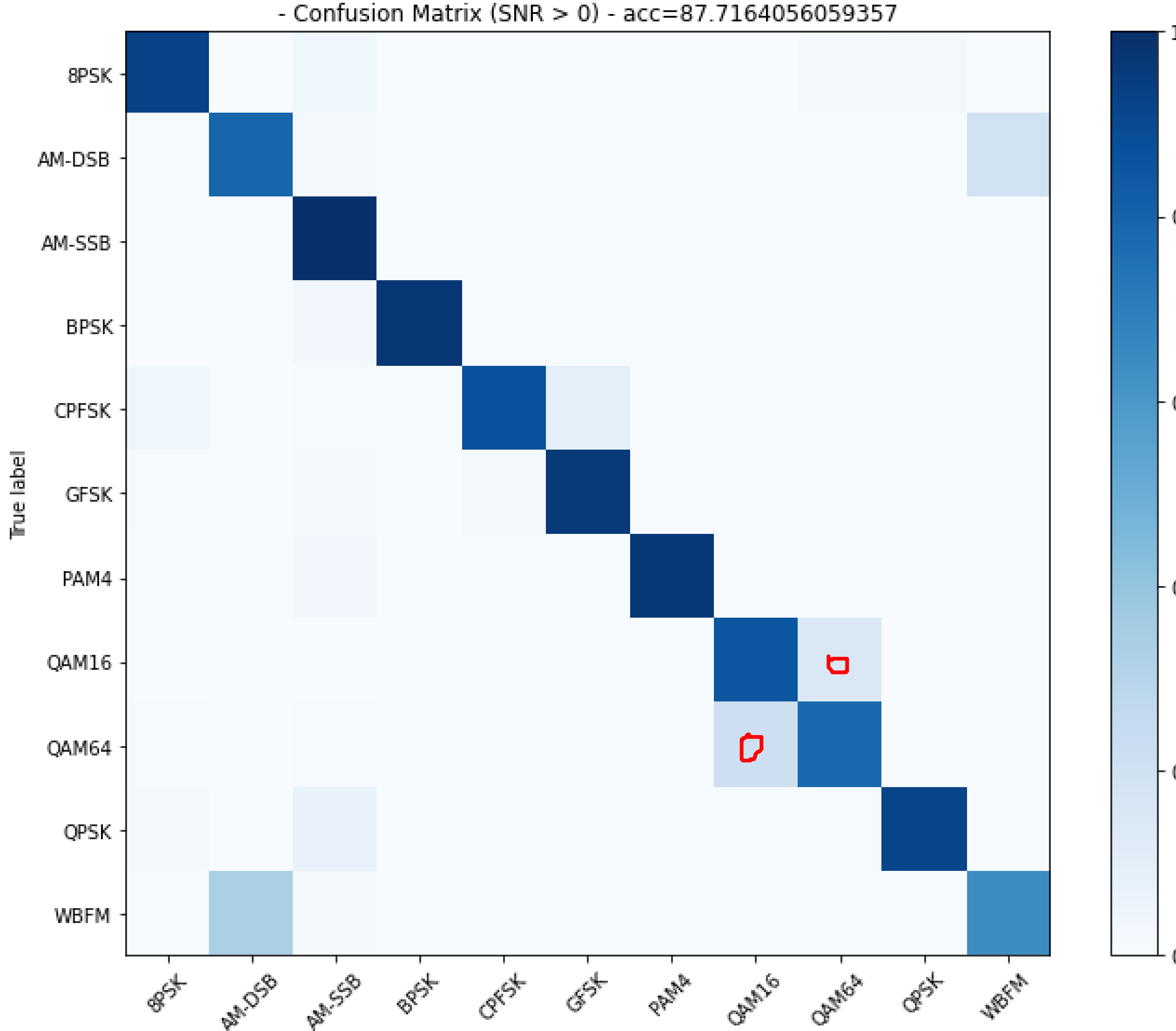
1. Representation of constellation

- Histogram captures finer detail, more effective in differentiating the two QAMs

Histogram



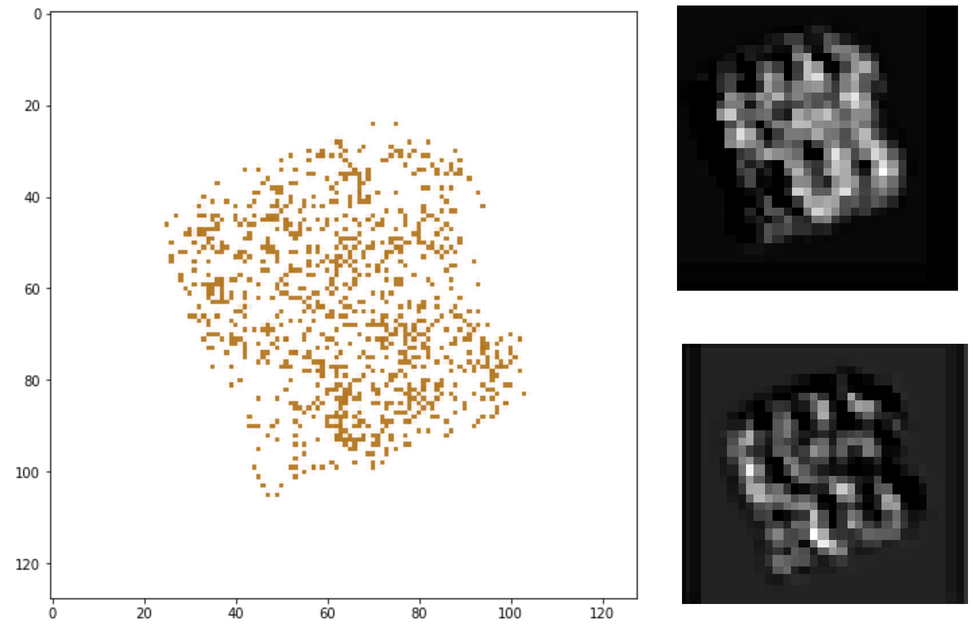
Gaussian KDE



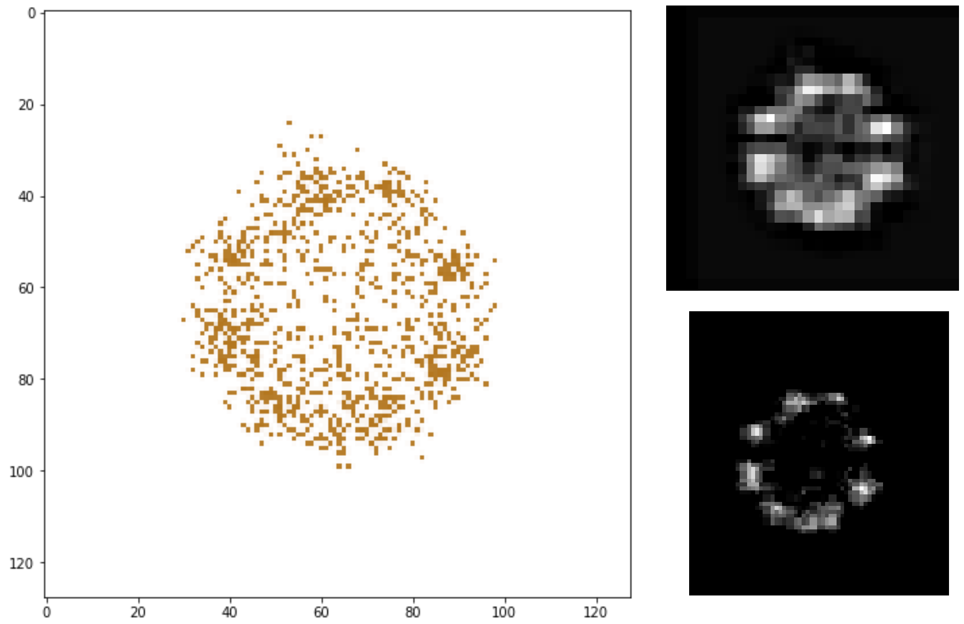
1. Visualisation of intermediate activation maps

- Even for constellations not coloured by density, model can still pick out density quite clearly

QAM16, 12dB



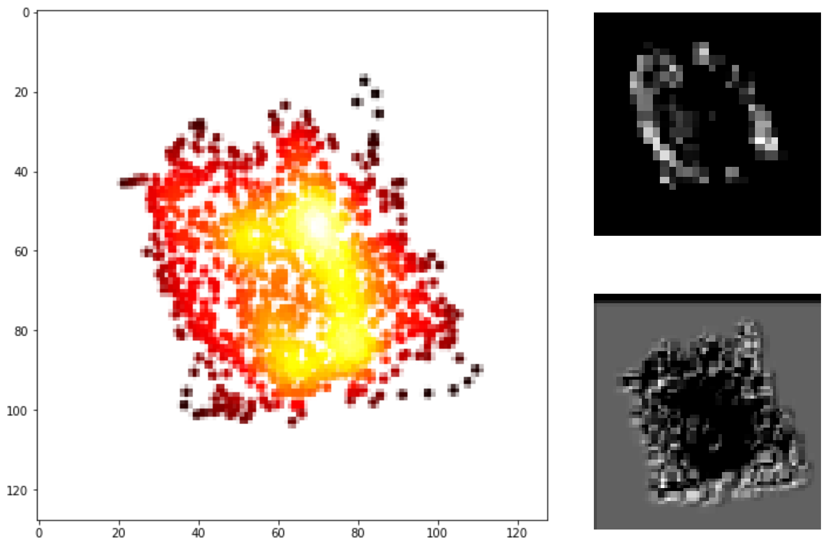
8PSK, 8dB



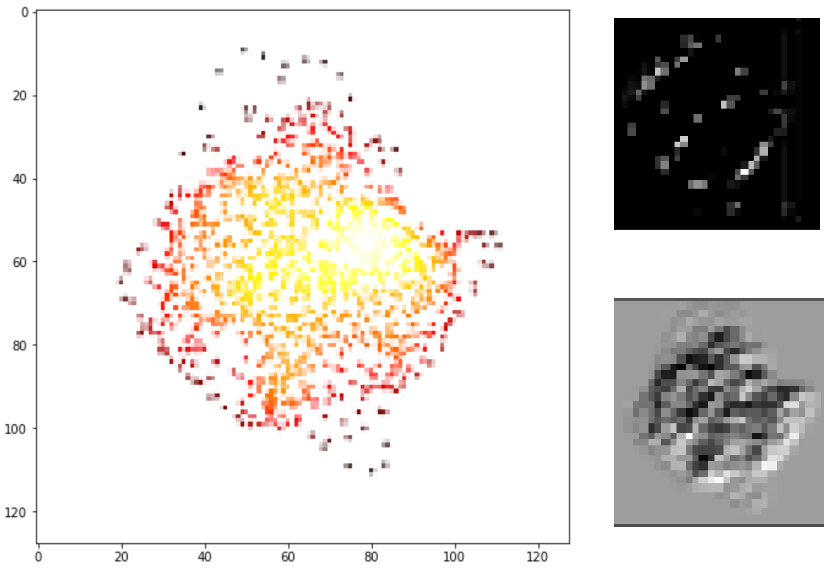
1. Visualisation of intermediate activation maps (coloured gaussian kde)

- Gaussian smoothing actually makes classification more difficult

QAM16, 6dB



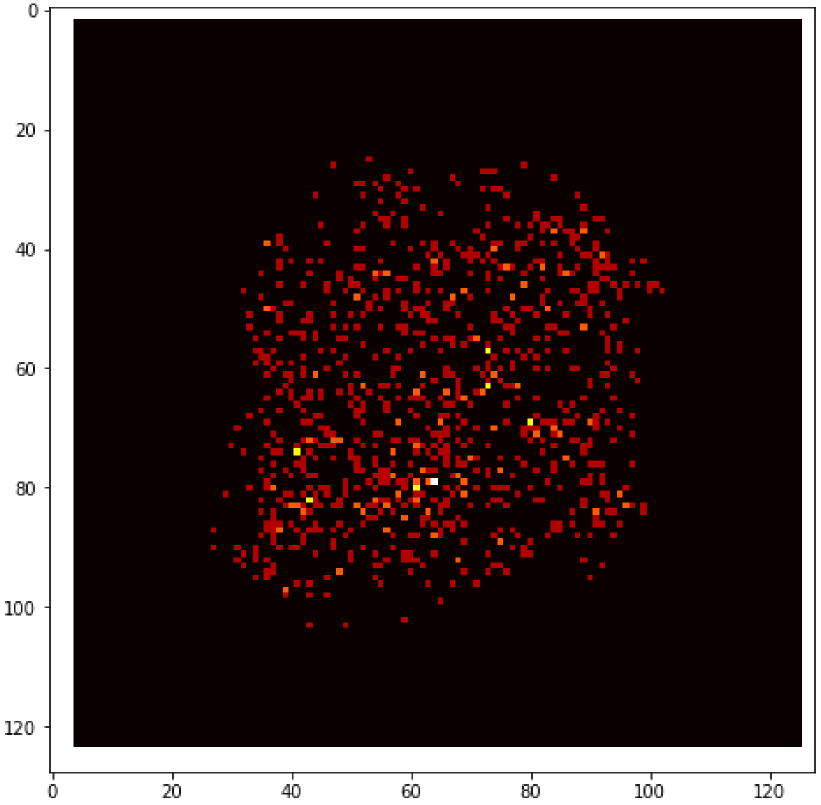
QAM64, 8dB

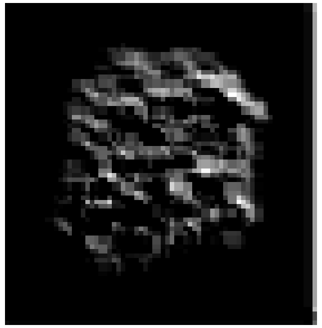


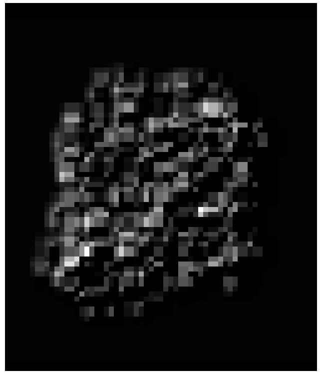
1. Visualisation of intermediate activation maps (histogrammed)

- Histogram effective at distinguishing different QAM

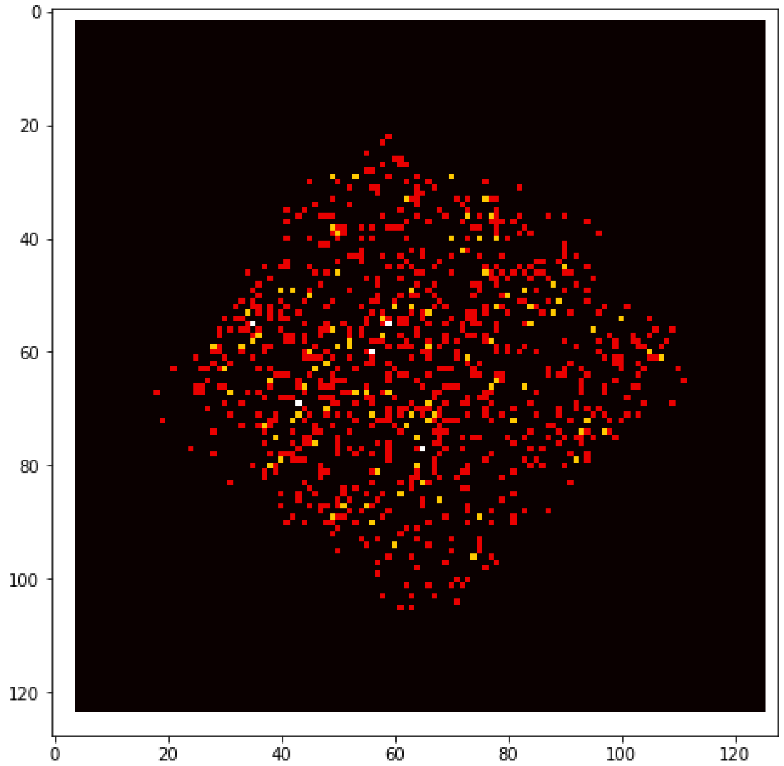
QAM64, 4dB

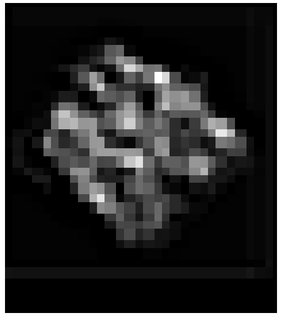


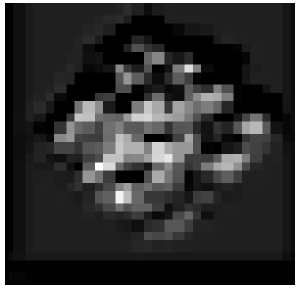




QAM16, 8dB

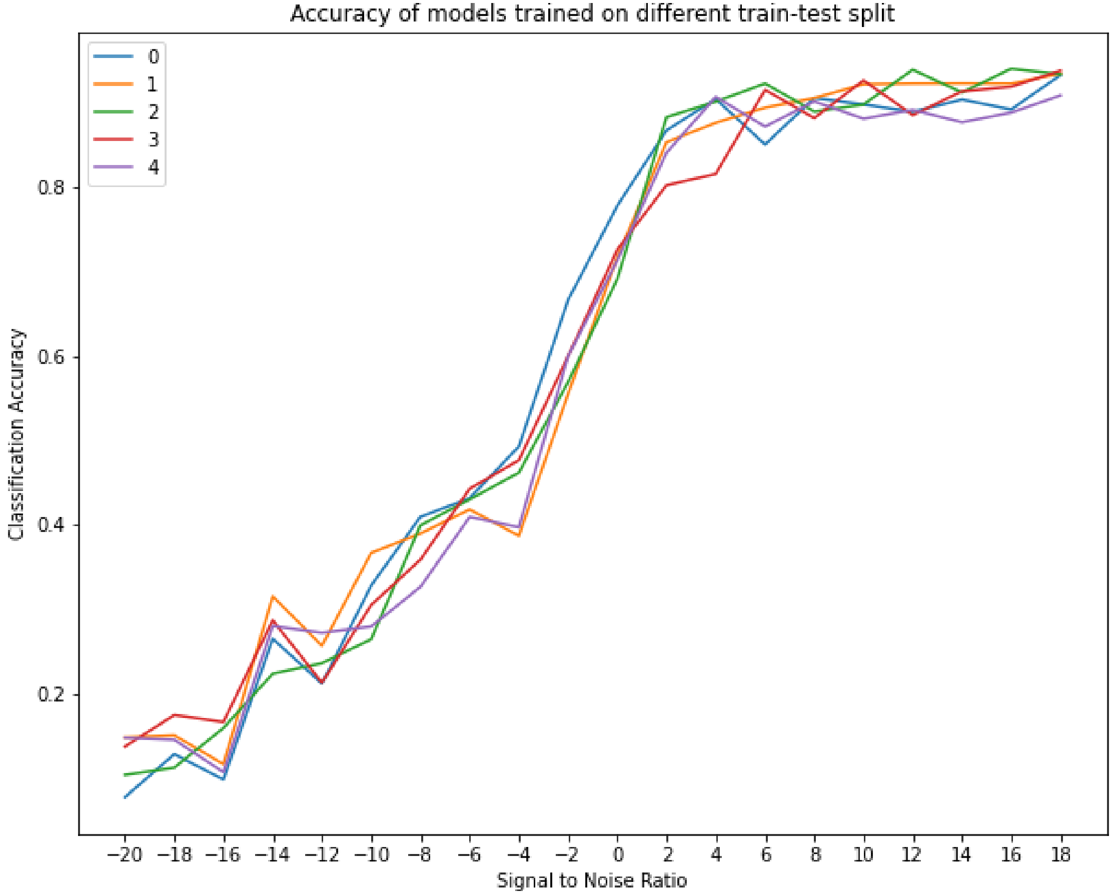


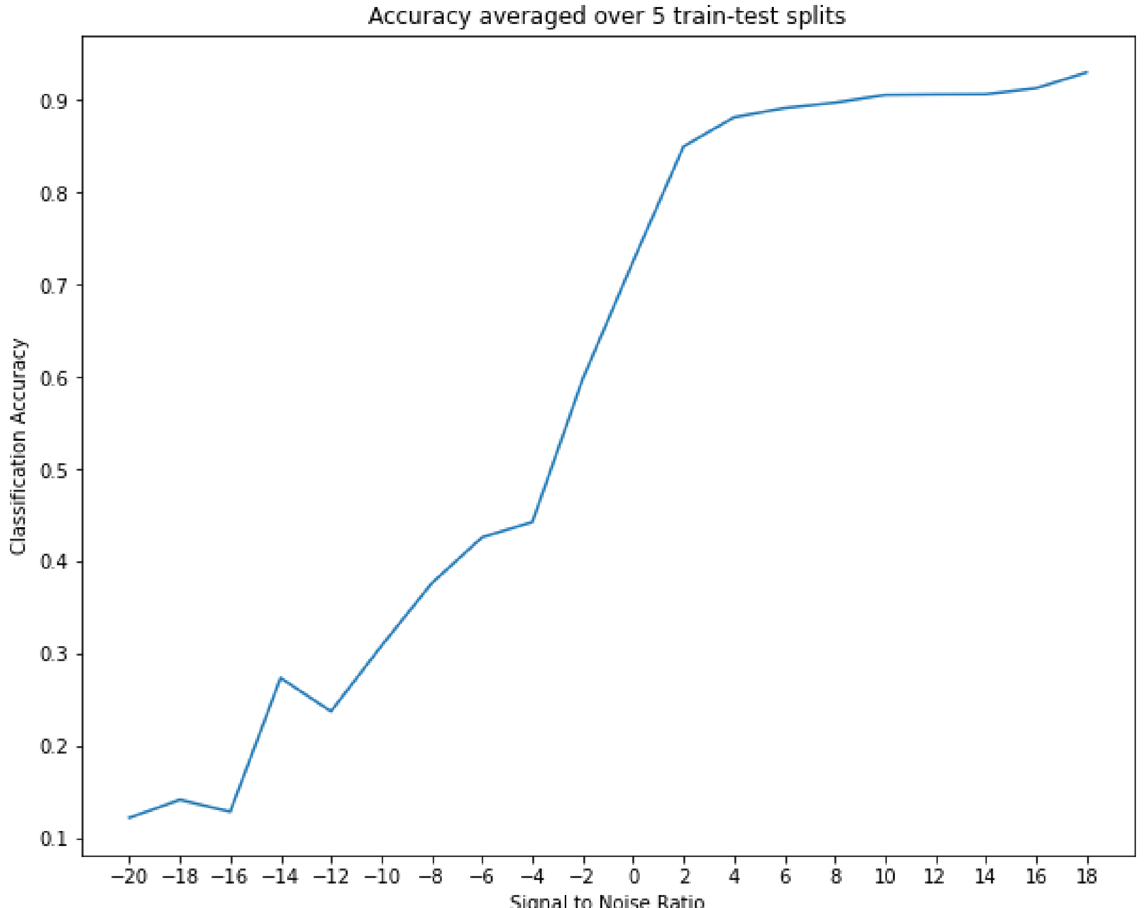




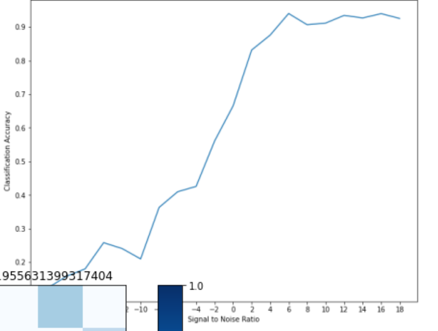
1. Constellation classification, cross validation

- **Divided dataset into 5 folds, each fold takes turn being test set**
- Even after averaging, we see kinks from -16 to -12, -6 to -2dB. (not sure why)
- -4dB onwards we get rapidly increasing accuracy, 2dB onwards start to saturate → maybe should just train with SNRs > -4dB?

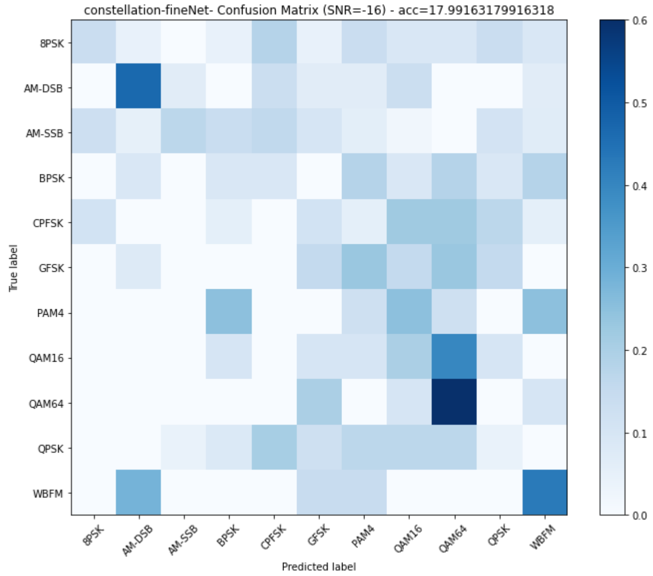




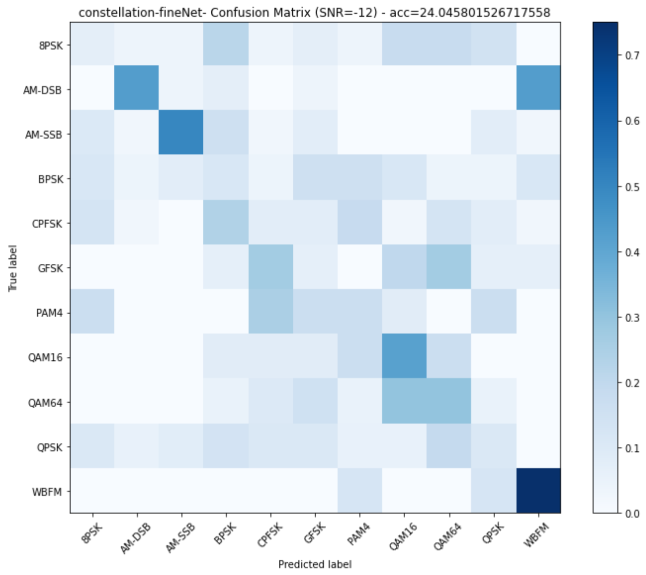
1. Investigate kinks at -16 to -12, -6 to -2dB



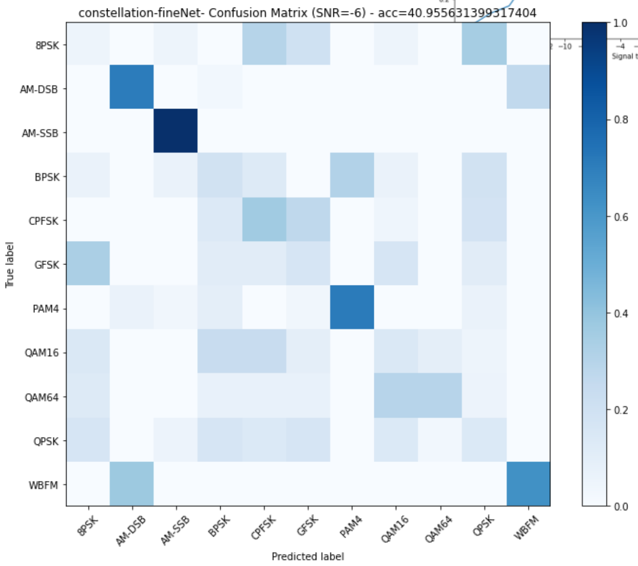
-16dB, 18%



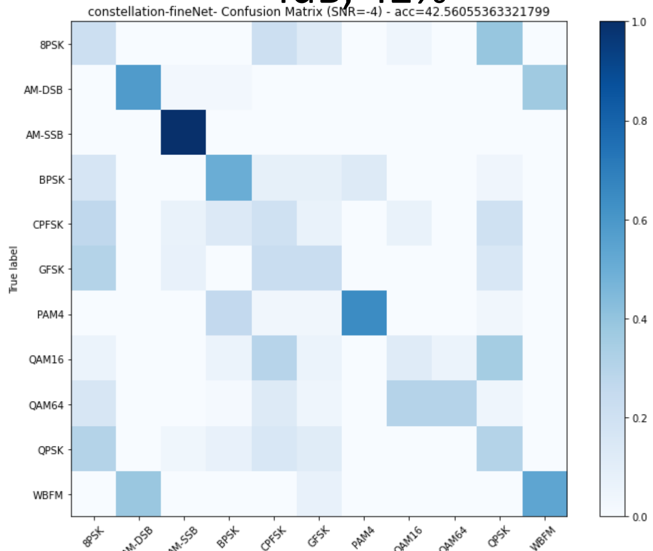
-12dB, 24%



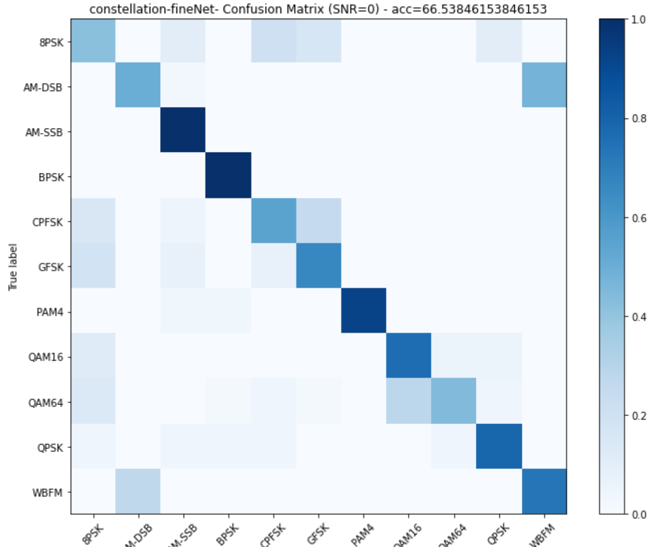
-6dB, 41%



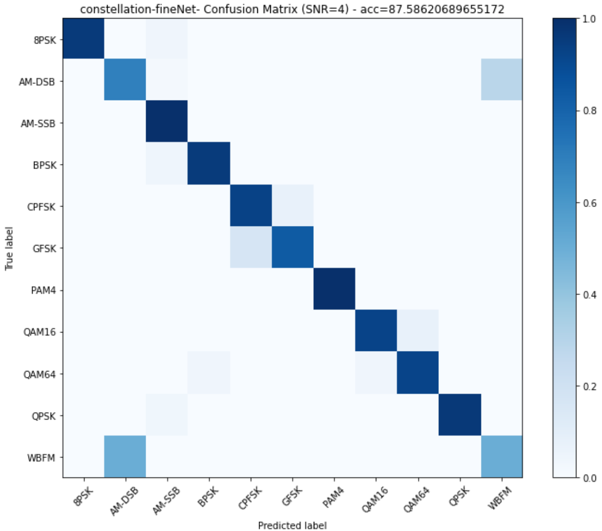
-4dB, 42%



0dB, 66%

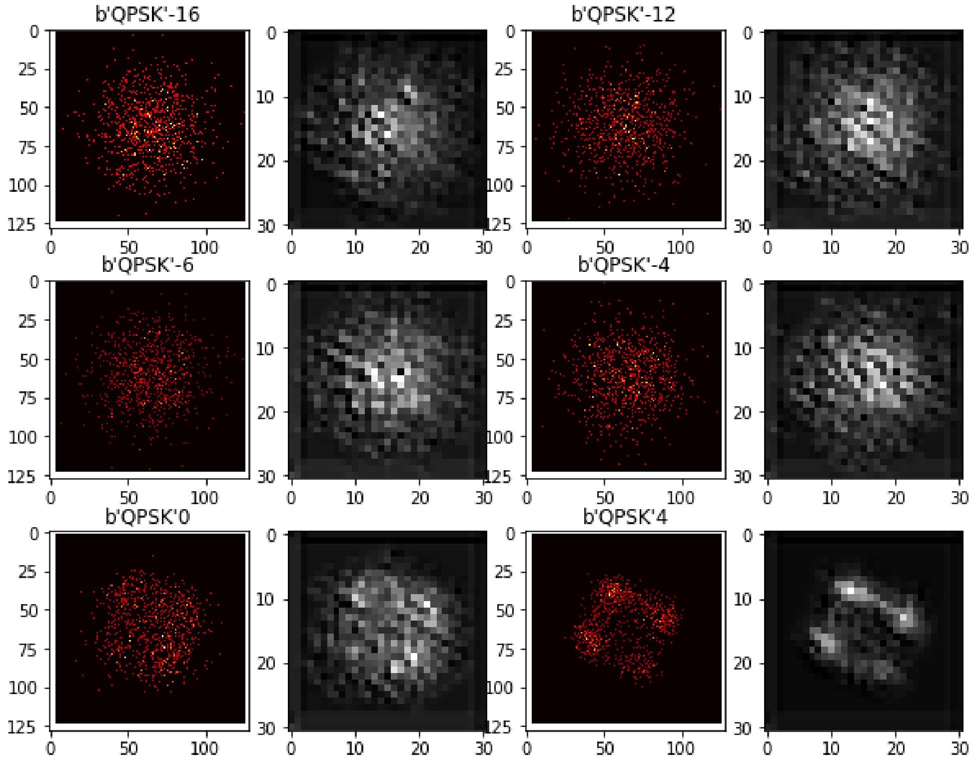


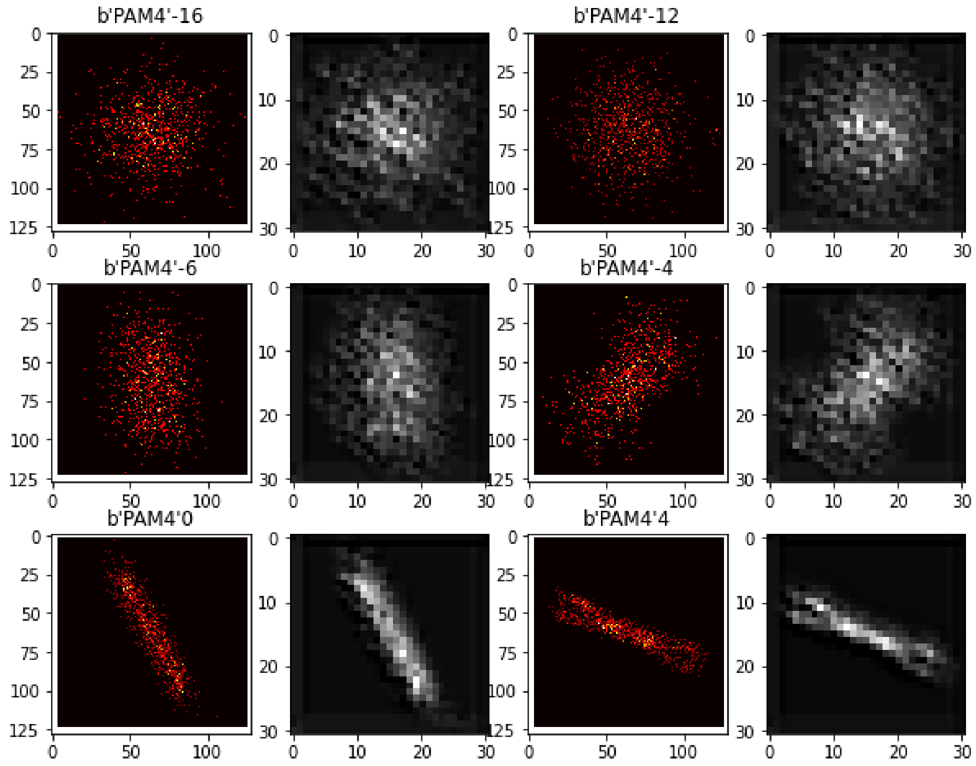
4dB, 87%



1. Investigate kinks at low SNR

- More easily recognisable constellations

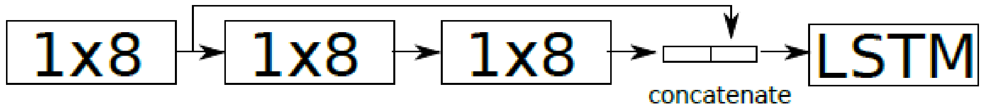




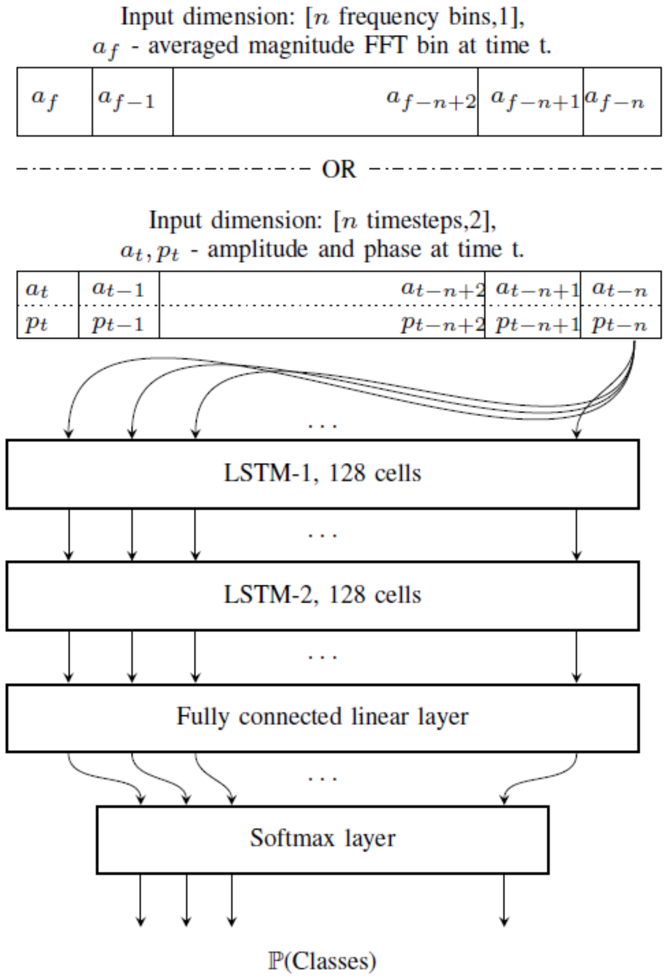
2. LSTM on time-series data, justification

- Deep Architectures for Modulation Recognition (DeepSig)
  - Communications receivers have a matched **filter**, **synchronizer**, and **sampler**.
  - The neural network architecture analog to this is a **convolutional layer** with **pooling** followed by an **LSTM**.
  - (?) Often the filter up front decimates to a small number of samples per symbol for the synchronizer which performs phase shifts to find the optimal sampling point. The sampler then slices to bits. The neural network architecture analog to this is a convolutional layer with pooling followed by an LSTM.
- Deep Learning Models for Wireless Signal Classification with Distributed Low-Cost Spectrum Sensors
  - LSTM allows efficient classification on variable sample rates and sequence lengths, unlike CNN models
  - Found that simple LSTM models can achieve good accuracy, if input data is formatted as amplitude and phase instead of IQ samples (not observed yet)

2. LSTM model architectures



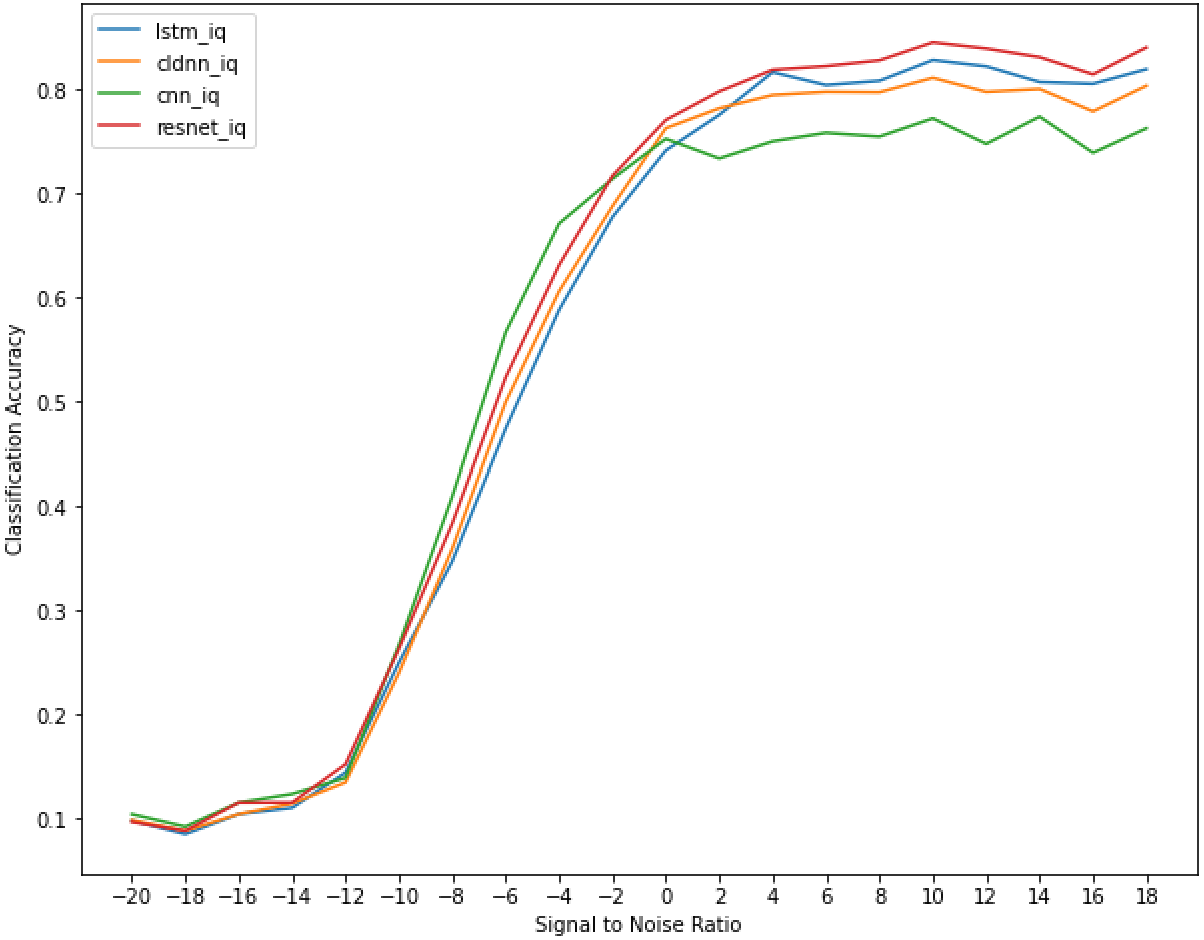
CLDNN from Deep Architectures for Modulation Recognition



Simple LSTM from Deep Learning Models fo
Wireless Signal Classification with Distribute
Low-Cost Spectrum Sensors

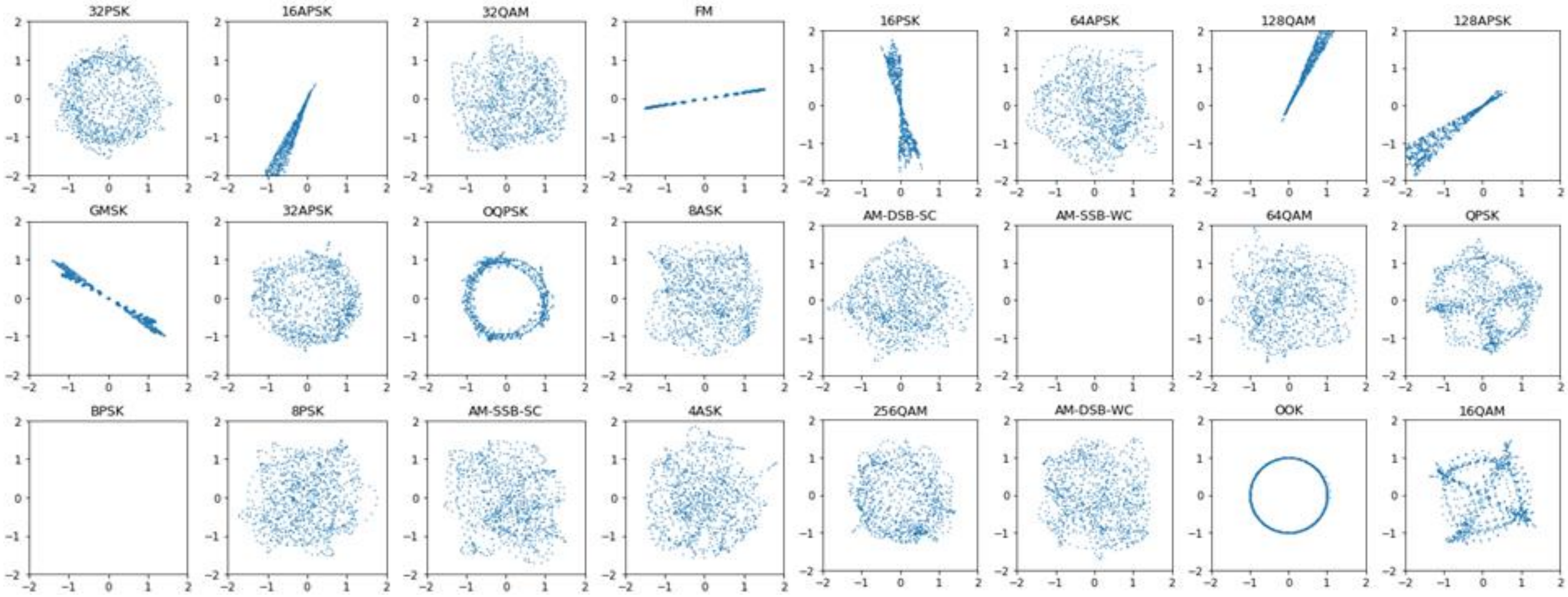
2. LSTM results

- Simple LSTM_IQ ~ LSTM_polar > CLDNN_IQ > CLDNN_polar
- Curve smooth, unlike in constellation.
- Haven't explored LSTM in detail, maybe ResNet + LSTM will improve also
- Also haven't figured out why different from paper results



3. Other datasets – 2018 radioML (DeepSig)

- 24 classes, labelled wrongly

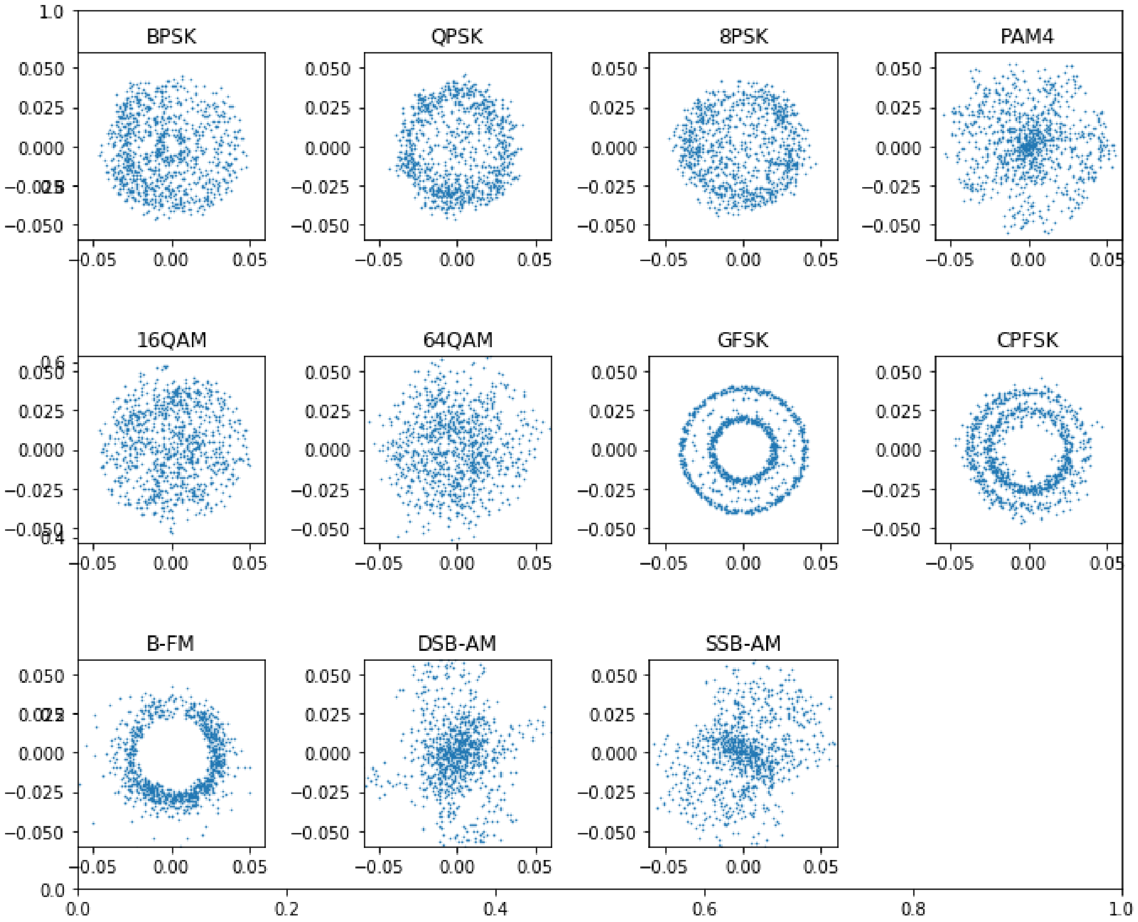


3. Other datasets - Matlab

- https://www.mathworks.com/help/deeplearning/ug/modulation-classification-with-deep-learning.html#ModulationClassificationWithDeepLearningExample-2
- Communications toolbox can simulate:
  - AWGN
  - Rician multipath fading
  - Clock offset, resulting in center frequency offset and sampling time drift

3. Matlab dataset

- AWGN 30dB → but a lot more noise than radioML
- Rician multipath
  - Delay profile [0 1.8 3.4] with gains [0 -2 -10] dB
- Max clock offset 5ppm



Next steps

- ~~Haven't really tested hyperparameters properly yet for constellation and I/Q classification, should tune and compare with papers~~
- ~~Explore LSTM more~~
- ~~See whether constellation can distinguish higher order modulation types~~
- ~~See how classification performs with different types of channel impairments~~

Discussed for next steps

1. Data collection
   Try to generate data using MATLAB with the following channel effect:
   a. free space path loss
   b. rayleigh
   c. Rician (urban)
   d. okumura-hata
   e. Egli
   (And what is the forest/jungle one?
2. Different models
   Ok, LSTM is the main type of RNN so it's ok.
   If you have time in a few weeks' time, maybe can try GAN? See if can regenerate deteriorated signals.
3. Multiple signals
   a. Superimposed at the same time
   b. Added sequentially
   Check whether model can detect

Note

- Understand before implement.
  - Model parameters, filter size, concatenate
  - When implementing from papers, understand why these methods work (e.g. constellation)
- Industry can classify -4dB alr (but hard to tell -4dB is how corrupted)