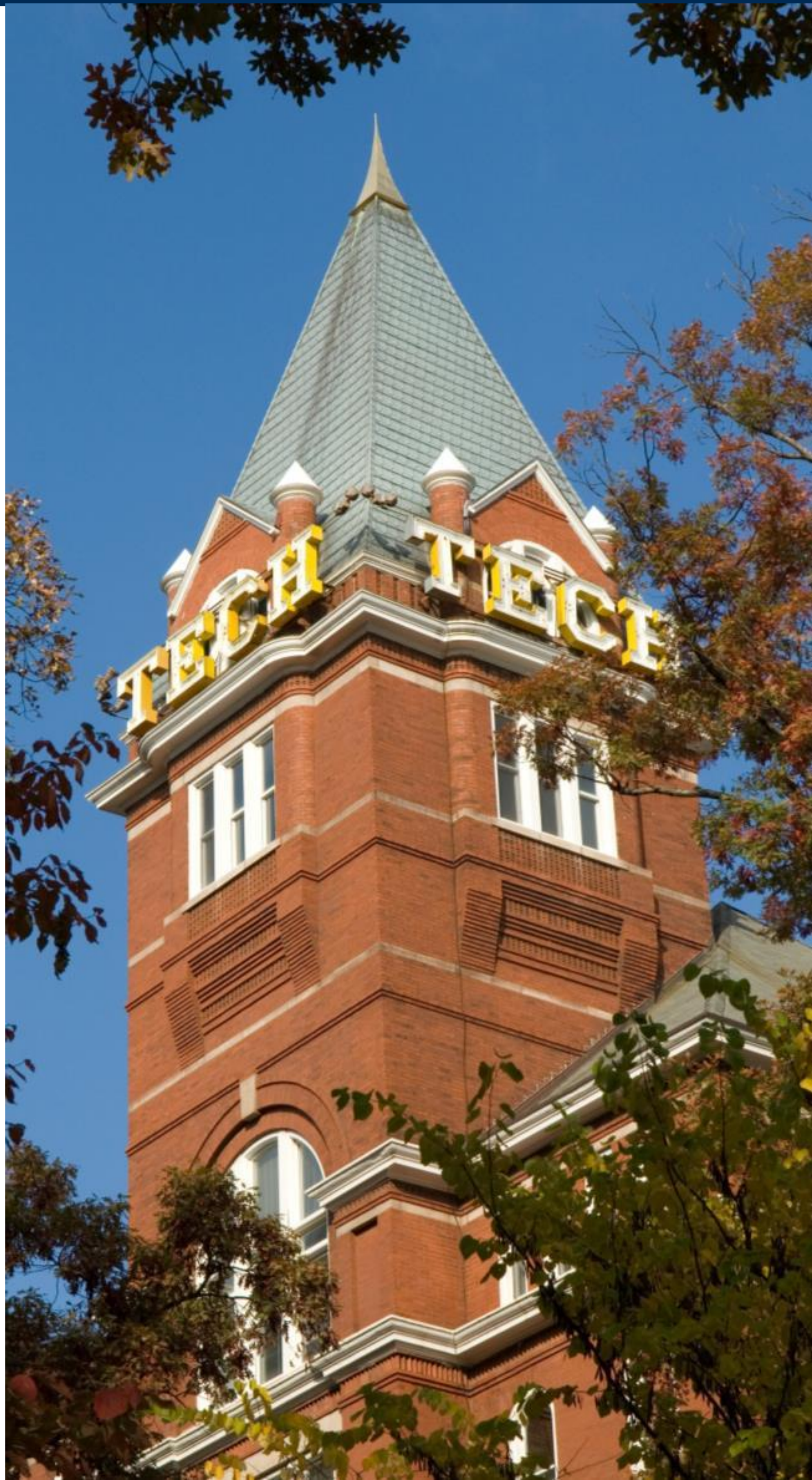
Georgia Tech Research Institute

Modulation Pattern detection Using Complex Convolutions in Deep Learning

25th International Conference on Pattern Recognition (ICPR 2020)

Jakob Krzyston[1,3], Rajib Bhattacharjea[2], Andrew Stark[3]

[1]Georgia Institute of Technology, [2]DeepSig.io, [3]Georgia Tech Research Institute



Introduction

- Transceivers used for telecommunications transmit and receive sequences of data that are conventionally thought of as representing complex numbers
  - Components are known as I and Q channels ($a + jb \rightarrow I + jQ$)

- Predetermined modulation patterns provide unique structure that can be decoded by the receiver to reproduce the intended message

- There are sources of noise and distortion in wireless channels, leading to errors in decoding the data, and making simple modulation classification particularly challenging

I/Q Modulation Overview

- I/Q signals are represented in Euler Form
  - $A(t)e^{j\phi(t)} = A(t)\cos(\phi(t)) + jA(t)\sin(\phi(t))$
    - $A(t)$: real-valued time-dependent magnitude
    - j: imaginary unit, $\sqrt{-1}$
    - $\phi(t)$: real-valued time-dependent angle of rotation
  - I Channel → $A(t)\cos(\phi(t))$
  - Q Channel → $A(t)\sin(\phi(t))$

- I/Q channel modulations can occur independently or jointly

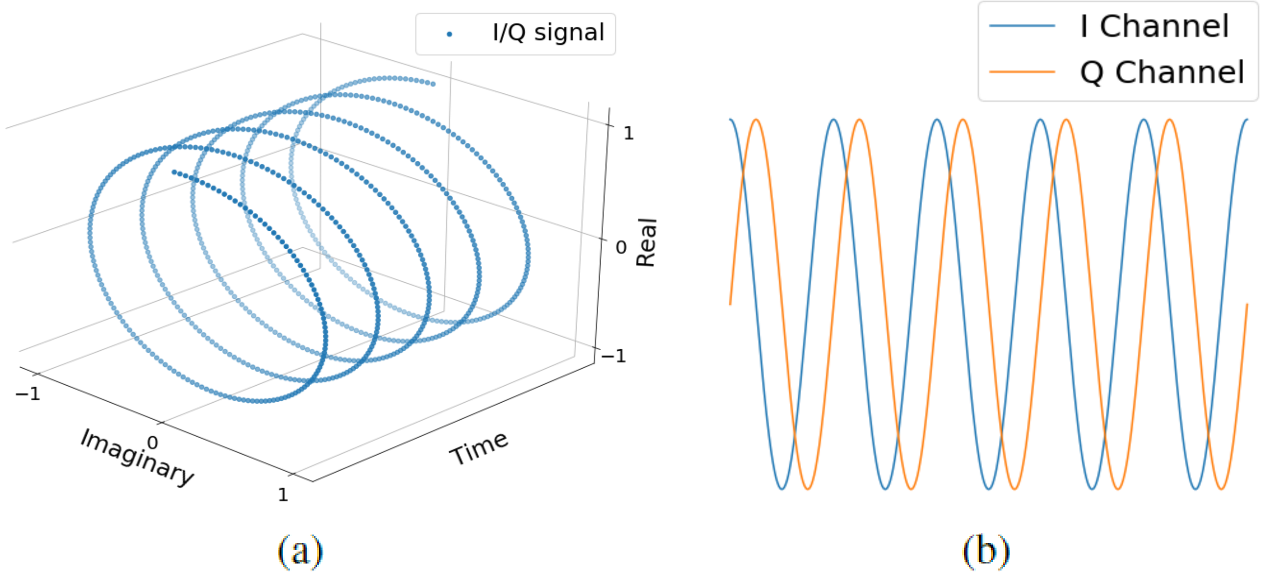


Fig. 1. (a) Example I/Q signal, $e^{j\phi(t)}$, shown in three dimensional space. (b) A two dimensional view of the example I/Q signal shown in 1a. The blue plot shows the I channel over time, $\cos(\phi(t))$, and the orange plot shows the Q channel over time, $\sin(\phi(t))$. In this report, this visualization will be used to display I and Q channels of a given I/Q sample.

Deep Learning for Modulation Classification

- Seminal work: O'Shea et al., 2016
- Since then, researchers have rushed to apply the latest architectures to this space
  - LSTM
  - CLDNN
  - ResNet
- Krzyston et al., 2020, demonstrated simple method to compute complex-valued convolutions using real-valued CNN's
- This work focused on simple CNNs to investigate the benefits of using complex convolutions in various train/test conditions (more on this later)

Complex Numbers in Deep Learning

- Arjovsky et al., 2016 suggest the use of complex numbers in deep learning for training stability
- Trabelsi et al., 2017 devised a method to extract complex-valued features from real-valued inputs
- Recently, Chakraborty et al., 2019 detailed a method to compute complex-valued convolutions via weighted Fréchet mean on a Lie Group
    - Necessitated special convolutional layer and activation function

Computing Complex Convolutions (Krzyston et al., 2020)

- Input: $Z_n = I_n + jQ_n, \quad I_n, Q_n \in \mathbb{R}$

$$\begin{array}{|c|c|} \hline I_1 & Q_1 \\ \hline I_2 & Q_2 \\ \hline I_3 & Q_3 \\ \hline \vdots & \vdots \\ \hline I_N & Q_N \\ \hline \end{array}$$

- Filter: $h_m = h'_m + jh''_m, \; h'_m, h''_m \in \mathbb{R}$

- Convolution via Deep Learning yields X_{DL}

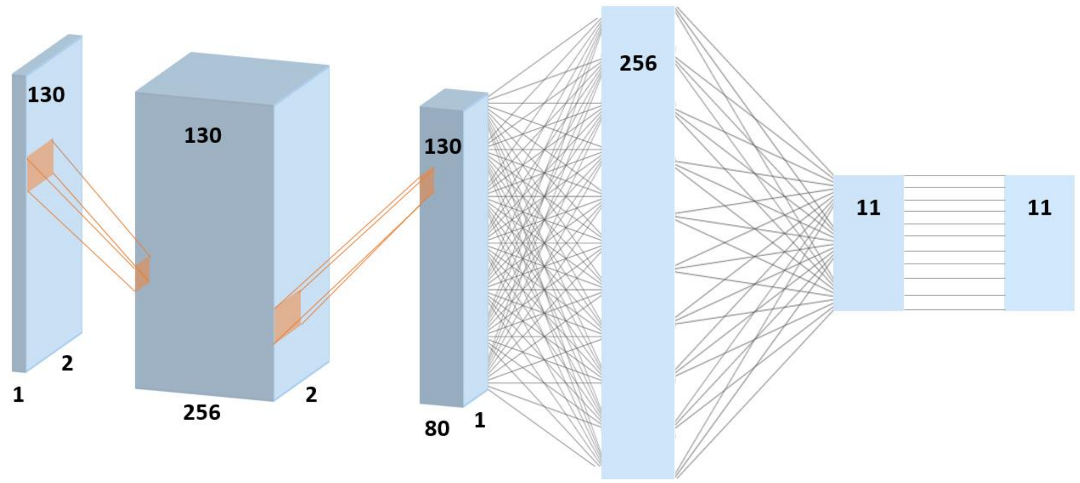
$$X_{\mathrm{DL}} = \begin{array}{|c|c|} \hline I_1 & Q_1 \\ \hline I_2 & Q_2 \\ \hline I_3 & Q_3 \\ \hline \vdots & \vdots \\ \hline I_N & Q_N \\ \hline \end{array} * \begin{array}{|c|c|} \hline h'_1 & h''_1 \\ \hline h'_2 & h''_2 \\ \hline h'_3 & h''_3 \\ \hline \vdots & \vdots \\ \hline h'_M & h''_M \\ \hline \end{array} \Longrightarrow X_{\mathrm{DL}} = \begin{array}{|c|c|c|} \hline \uparrow & \uparrow & \uparrow \\ I * h' & I * h'' + Q * h' & Q * h'' \\ \downarrow & \downarrow & \downarrow \\ \hline \end{array}$$

- Linear combination corrects this $X = X_{\mathrm{DL}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix}$

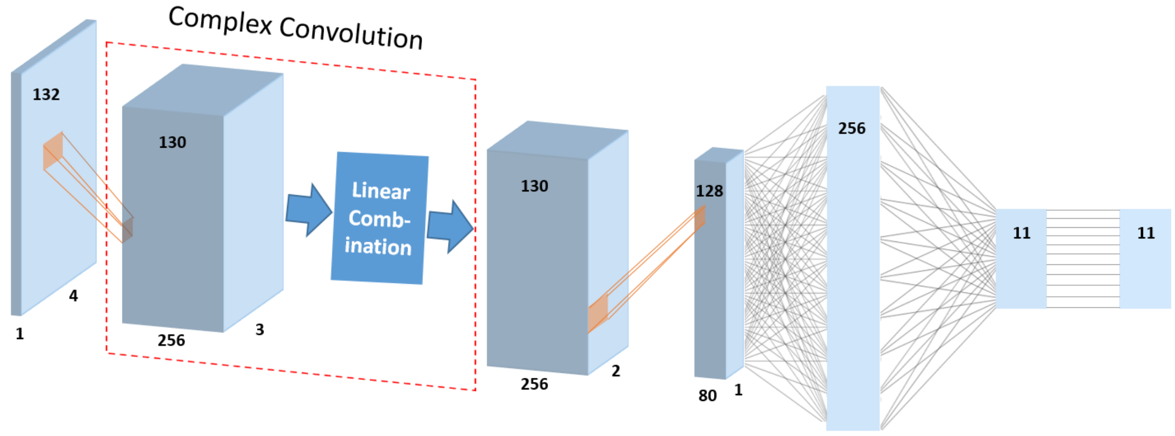
$$\Longrightarrow X = \begin{array}{|c|c|} \hline \uparrow & \uparrow \\ I * h' - Q * h'' & I * h'' + Q * h' \\ \downarrow & \downarrow \\ \hline \end{array}$$

Deep Learning Architectures

- CNN2 (O'Shea et al., 2016)
  - Two convolutional layers
  - All convolutional filters are 1D
  - 256 node dense layer



- Krzyston 2020 (Krzyston et al., 2020)
  - One complex convolutional layer
    - Convolutional filter is 2D
  - One traditional convolutional layer
  - 256 node dense layer



- CNN2-257 (Krzyston et al., 2020)
  - CNN2 with 257 node dense layer
  - More parameters than Krzyston 2020

I/Q Modulation Classification Task & Experimental Design

- Dataset
  - RadioML 2016.10A open source dataset
  - 11 modulation classes
  - SNR range (in dB): [-20,18], steps of 2 dB
  - Numerous types of noise
  - 1000 samples per modulation pattern per dB SNR

- Train/Test Paradigms

Train dB SNR	Test dB SNR
[-20,18]	[-20,18]
[-20,-2]	[0,18]
[0,18]	[-20,-2]

  - Training split 50/50
  - Each experiment repeated five times

Results: Classification

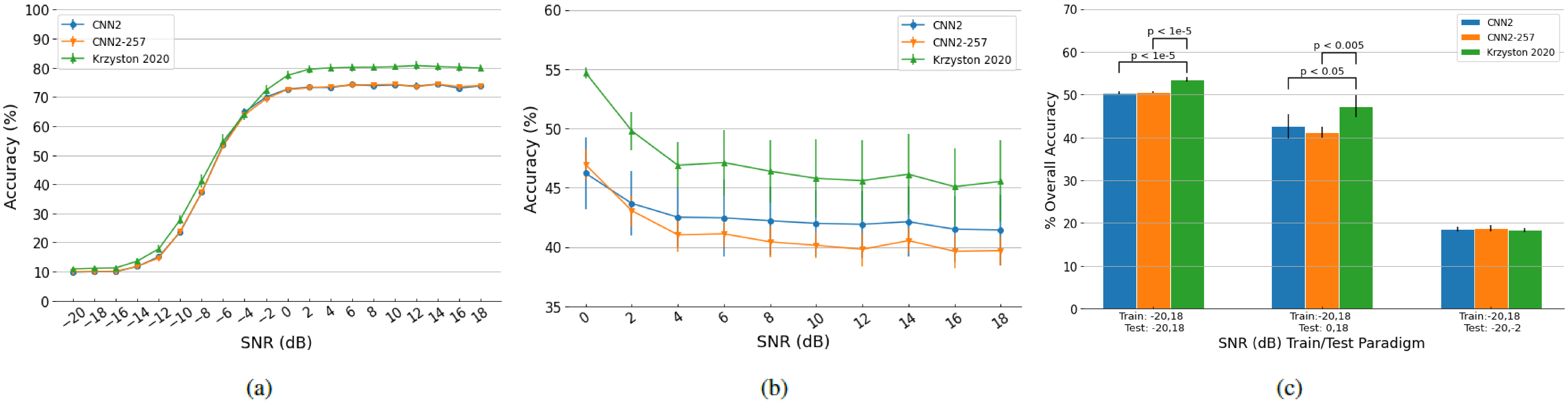


Fig. 4. Classification accuracy plots as a function of SNR, with the addition of standard deviation bars over the five trials, of (a) the experiment performed in [7], and (b) Experiment 1 the networks were trained on [-20, -2] db SNR data and tested on [0, 20] db SNR data. Figure 4(c) displays the average accuracy, over all modulations and SNRs, and standard deviation for each experiment, along with p-values comparing the performances amongst the architectures, if statistically significant. The unpaired student t-test was used to compute the p-values.

Results: 'One-Hot' Activations

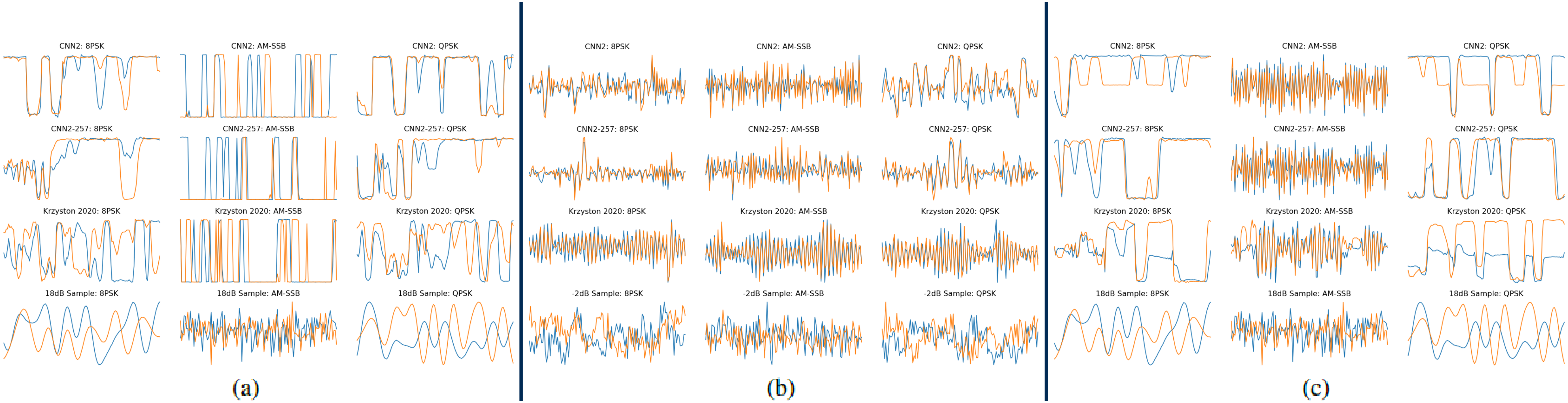


Fig. 8. Computed maximized input images for each architecture in (a) experiment performed in [7], (b) Experiment 1, and (c) Experiment 2. These images result in one-hot classification by the respective architecture, for the respective modulation pattern. Additionally, a sample from each modulation at the highest SNR included in the training set of that train/test paradigm is included for visual comparison.

Conclusion

- We demonstrate the ability to compute complex convolutions in CNNs outperforms traditional CNNs with statistical significance in two of three train/test paradigms

- Complex convolutions are able to capture more useful content in a complex signal than traditional deep learning convolutions

- Email: jakobk@gatech.edu