Denoising Autoencoder for raw, wireplane, waveforms in DUNE’s LArTPC detector

June 20, 2023

Van Tha Bik Lian, Kate Scholberg, Mike Wang, Benjamin Hawks, Janina Hackenmueller, Tejin Cai, Jovan Mitrevski, Tingjun Yang, Thomas Junk, Maira Khan, Jennifer Ngadiuba

Accelerated AI Algorithms for Data-Driven Discovery

Duke UNIVERSITY

Overview

- Background/Motivation
- DUNE LArTPC detector
- Challenge for low energy signals
- Deep learning approach
  - 1DCNN ROI finder (possible replacement for DUNE primitive triggering algorithm)
  - **Autoencoder** (CURRENT PROJECT)

Core Collapse Supernova

Image from: Fast Machine Learning for Science Worskshop 2022, 10/3-10/6 SMU (Ben Hawks)

Core Collapse Supernova

So far…

Managed to observe due to amature astronomers looking at the right spot!

We want more!

How?

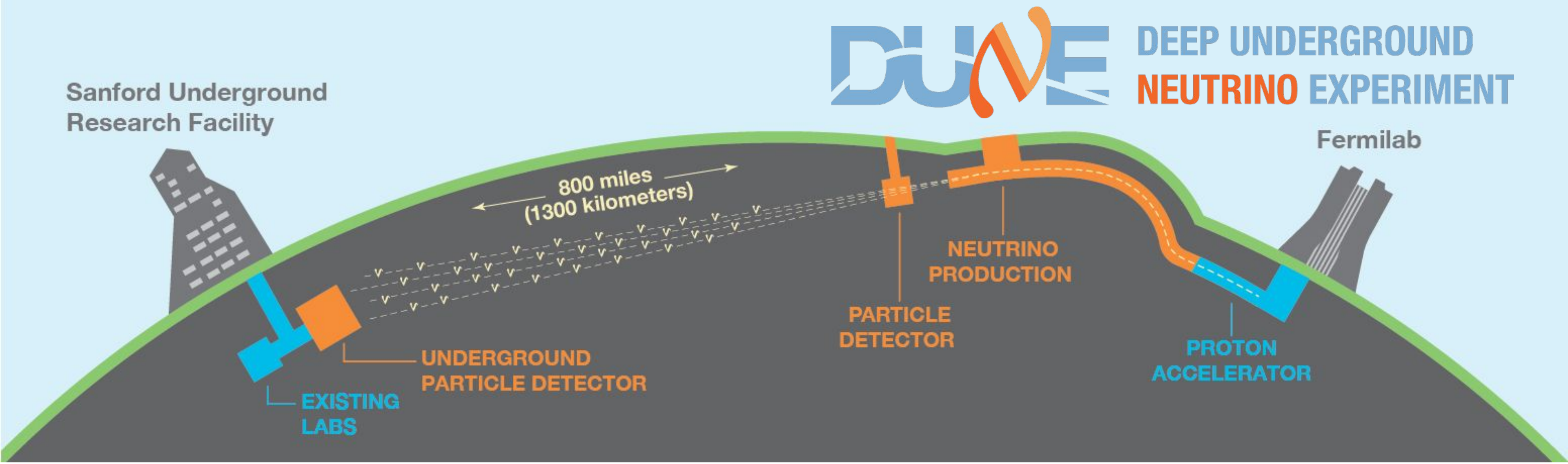
Neutrinos!

- Bursts of neutrinos are produced and sent out into the universe as a result of a core-collapse supernova event
- They may pass through our atmosphere
    - Their detection will improve our knowledge of core collapse rates [1]
    - If we detect them fast enough, we can **coordinate different instruments, like the LEGACY SURVEY OF SPACE AND TIME @ RUBIN OBSERVATORY (LSST) to better observe and understand these events in real time**

Neutrinos are know to be highly elusive

- Need MASSIVE detectors

Background:



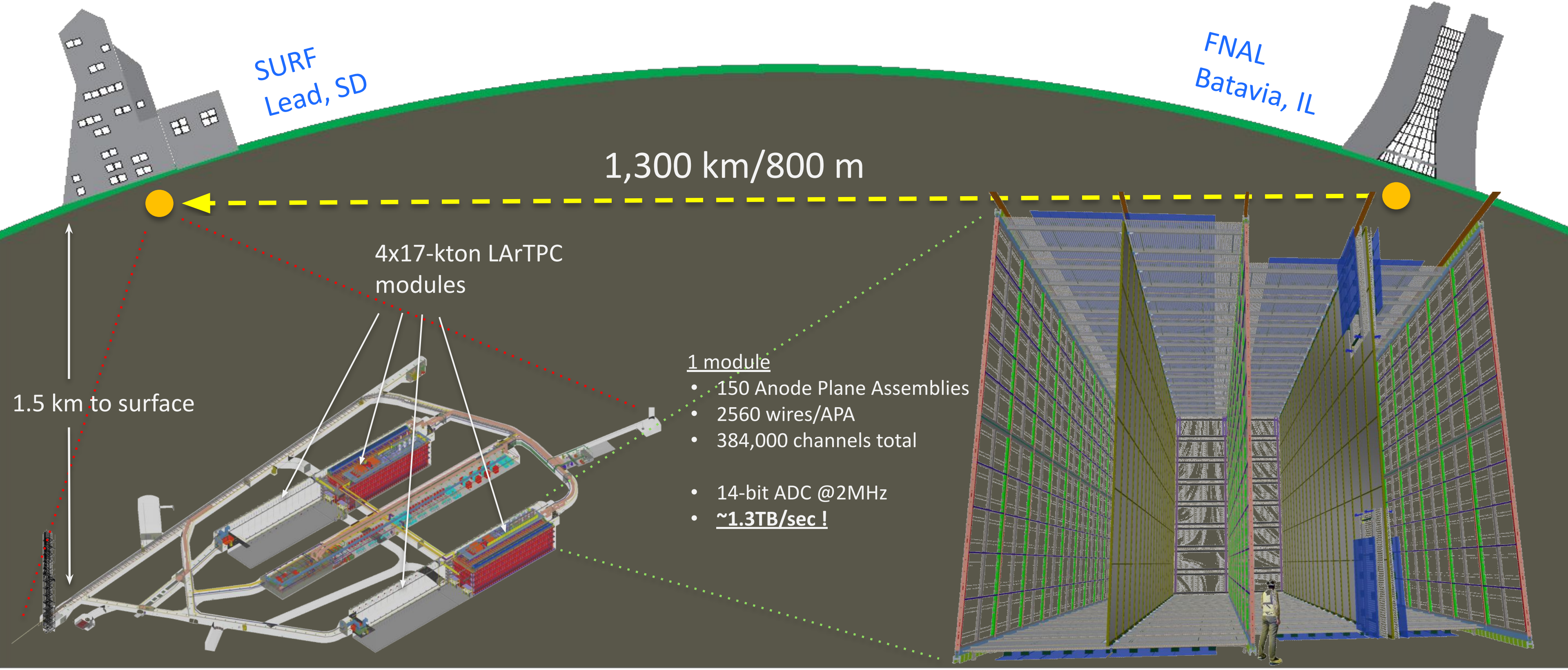
DUNE: long baseline physics program

- Determining neutrino mass hierarchy
- Observing CP violation
- Precise measurement of neutrino oscillation parameters

Beyond the long-baseline program

- detection of neutrinos from core-collapse supernovae, searches for nucleon decay, studies of solar neutrinos, and atmospheric neutrino oscillation studies to supplement the long-baseline measurements
- energy range in the 1MeV (solar) to 10 MeV(core-collapse)

Background: Liquid Argon Time Projection Chambers (LArTPC)



Background: Liquid Argon Time Projection Chambers (LArTPC)

Always on Detectors

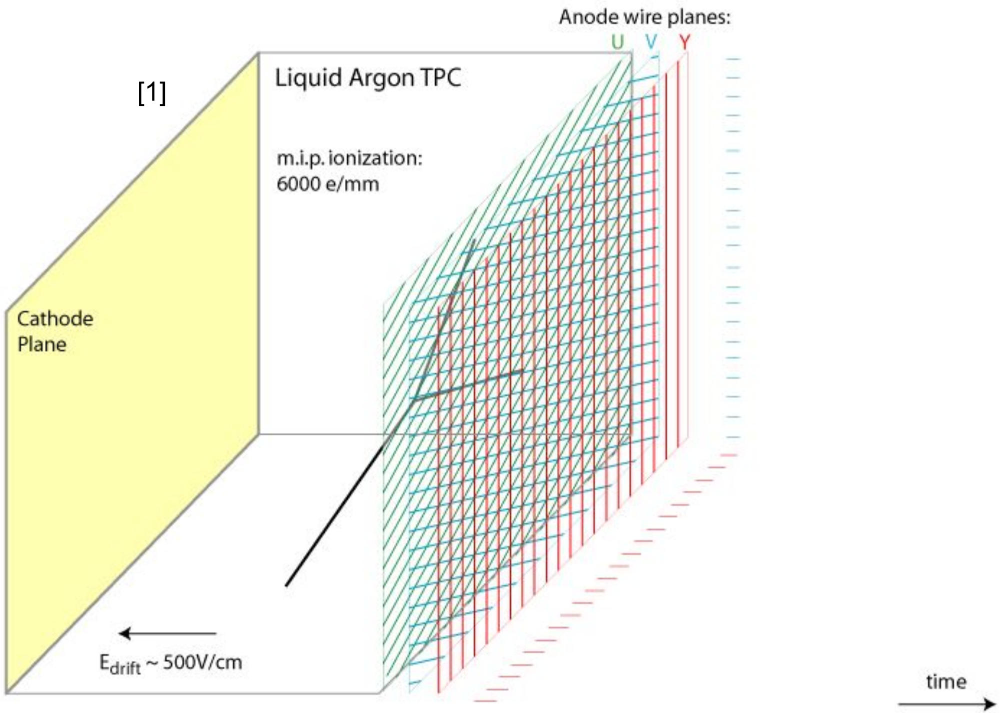
- Signals induced by ionization charges
- Wire planes at the end of drift path

Electronic readout

- Multiple wire planes with different angular orientations (2 spatial coordinates)
- Combined with a third from drift time, we can do detailed reconstruction

Technology of choice for massive next generation neutrino experiments like DUNE

[1]



https://arxiv.org/pdf/2106.09911.pdf

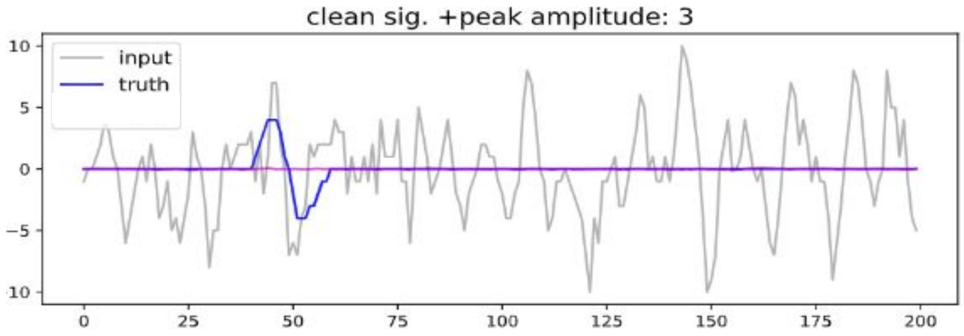
Beyond the long-baseline program

- Detection of neutrinos from core-collapse supernovae
- have energy as low as 10 MeV

Induced signals

- They are close to the noise threshold
- Conventional approach applies a minimum ADC threshold cuts which discriminates signal waveforms from noise

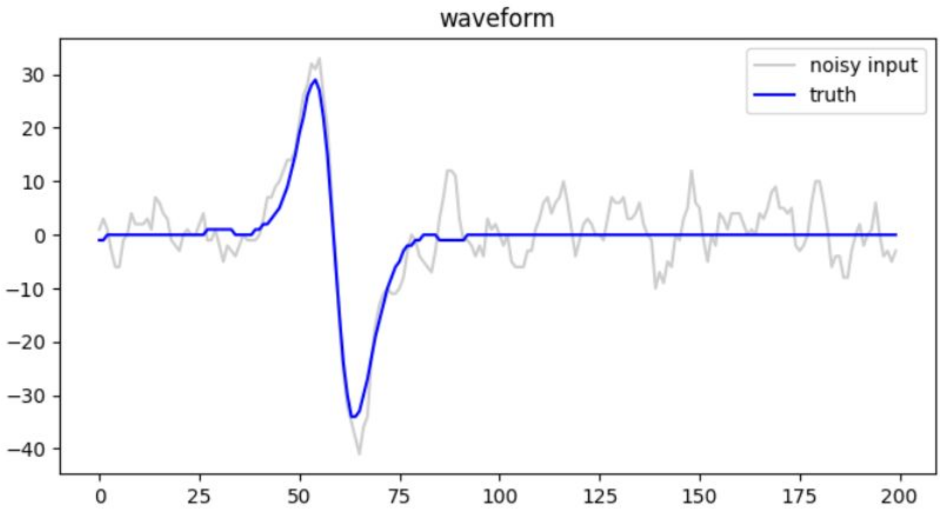
→ Results in poor low-energy efficiency

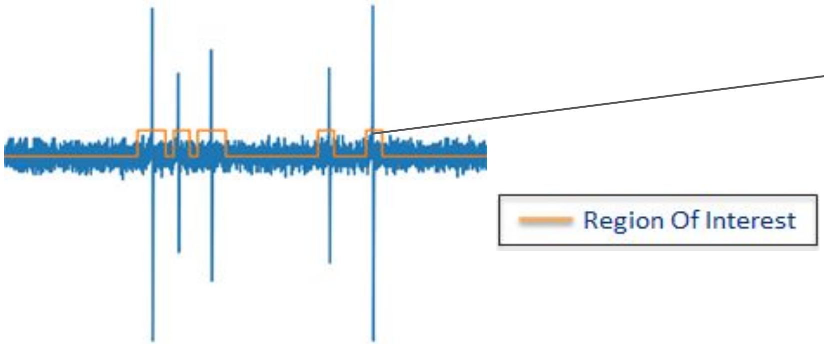


Task: Optimize low-energy efficiently using DL approach

Develop DL technique to apply to the raw waveforms from individual LArTPC wires

- Detect presence of a signal and identify a region of interest (ROI)
- First attempt to apply DL methods directly to raw waveforms associated with single LArTPC wires
- Potential to be applied in the low-level filtering and triggering in online data acquisition (DAQ) systems





ROI pred: 1.0

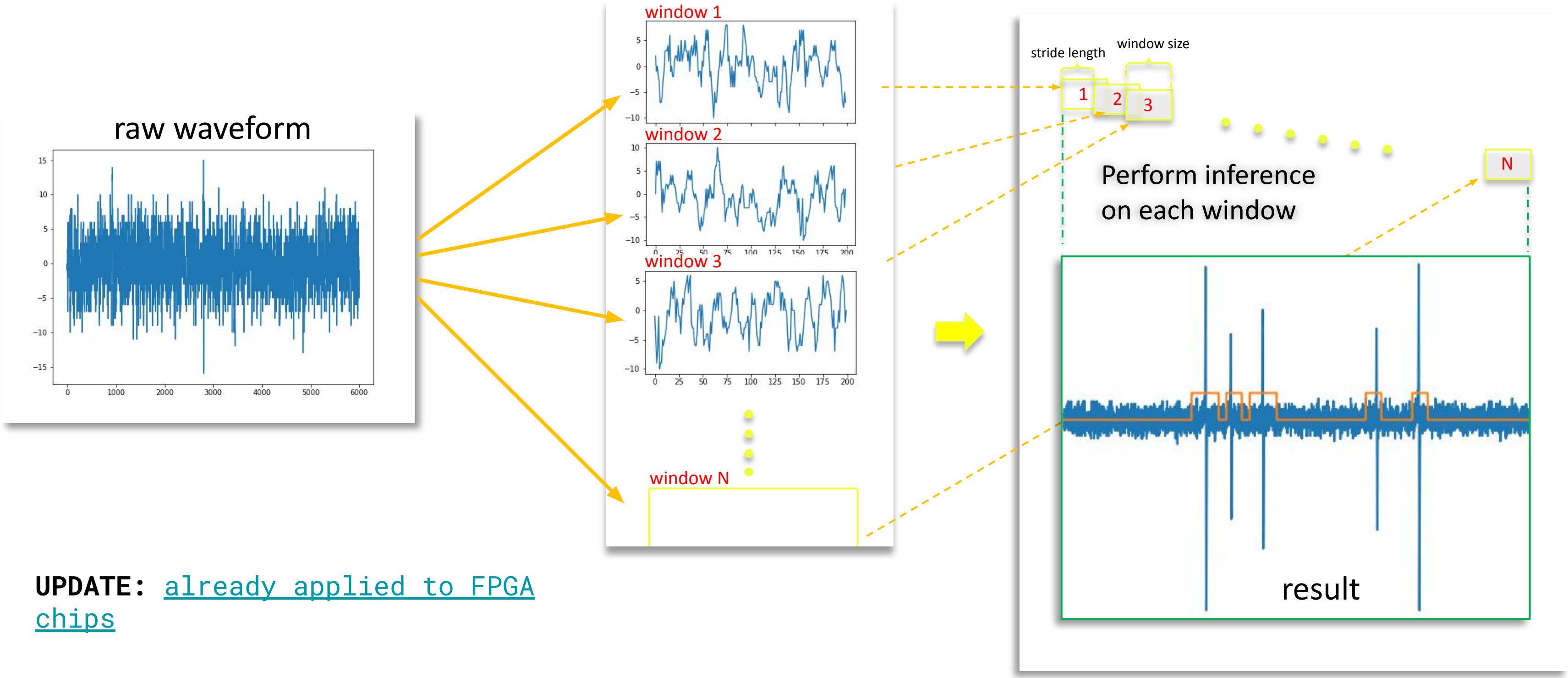
ROI truth: 1.0

ROI pred >= 0.94 consider signal

Reference to paper: [Extracting low energy signals from raw LArTPC waveforms using deep learning techniques — A proof of concept]

Fast Machine Learning for Science Worskshop 2022, 10/3-10/6 SMU (Ben Hawks)

ALGORITHM - ROI FINDER 1DCNN MODEL



UPDATE: already applied to FPGA chips

TWEPP 2022 Topical Workshop on Electronics for Particle Physics, 9/19-9/23 Norway (Jovan Mitrevski)

Fast Machine Learning for Science Worskshop 2022, 10/3-10/6 SMU (Ben Hawks)

Reference to paper: Extracting low energy signals from raw LArTPC waveforms using deep learning techniques — A proof of concept

Dataset (Monte Carlo)

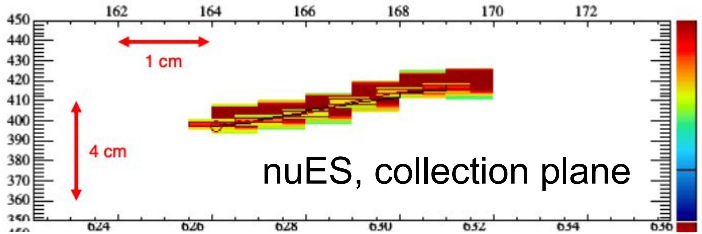
Type of 'signals'

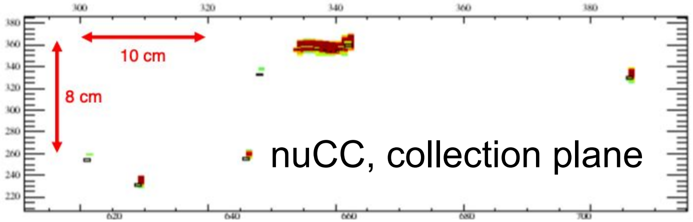
- nuES/nuCC
  - Two types of neutrino interactions
  - ***nuES:*** Neutrino Elastic Scattering off electrons
  - ***nuCC:*** Neutrino Charged Current Interaction
- Ar39
  - Radiological background
    - **NOTE:** They are not what we are interested but we are concerned with signal wire waveforms so these are useful
    - Ar39 samples were produced to have a more balanced number of samples across ADC counts

Noise:

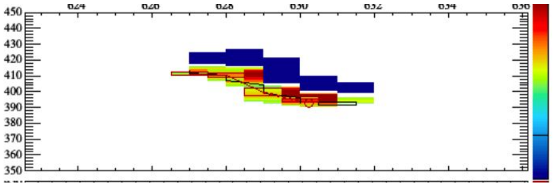
- Electronic noise

2D projections of wire vs time

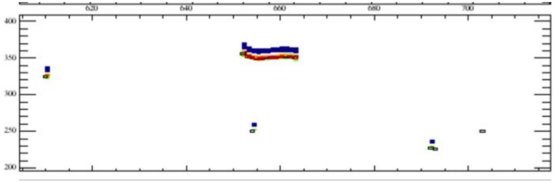




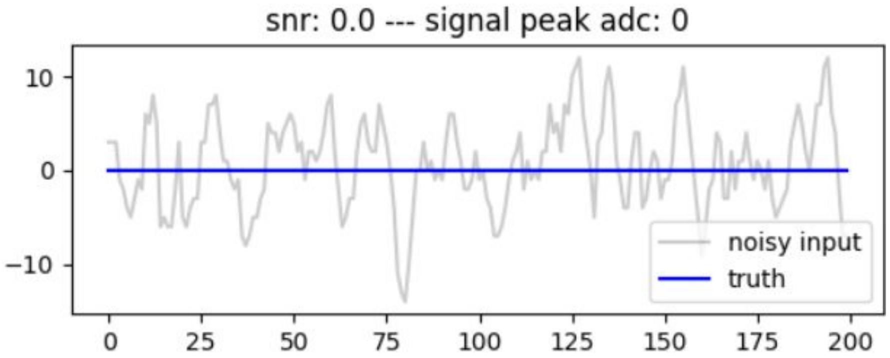
nuES, induction plane

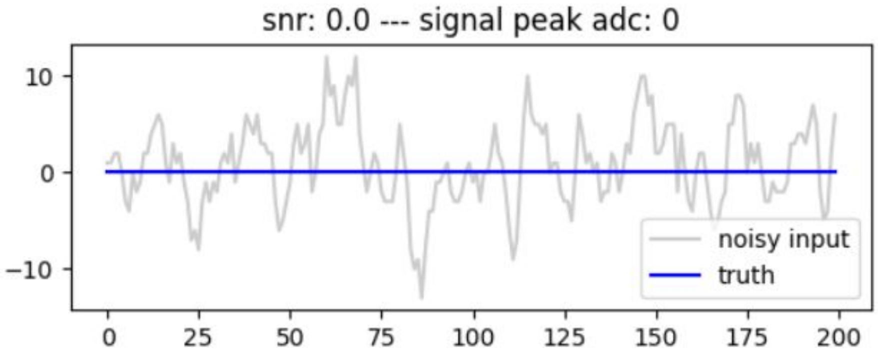


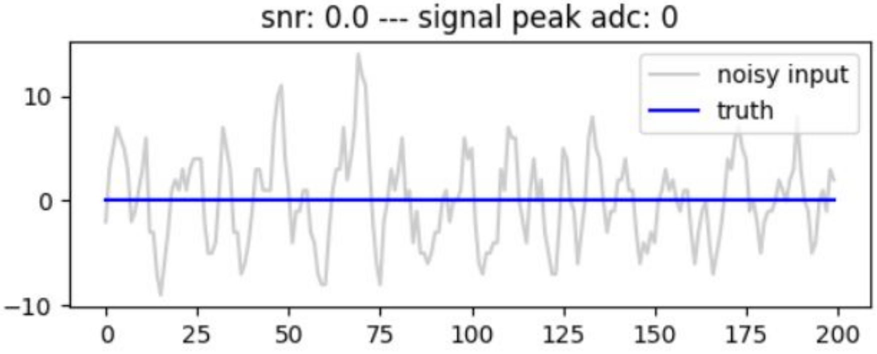
nuCC, induction plane

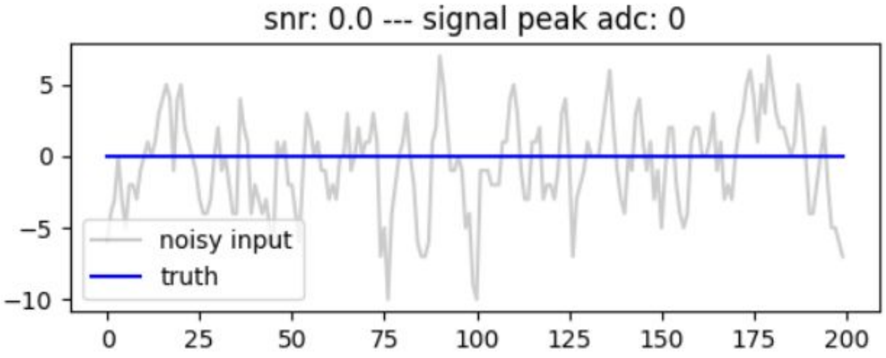


How they look



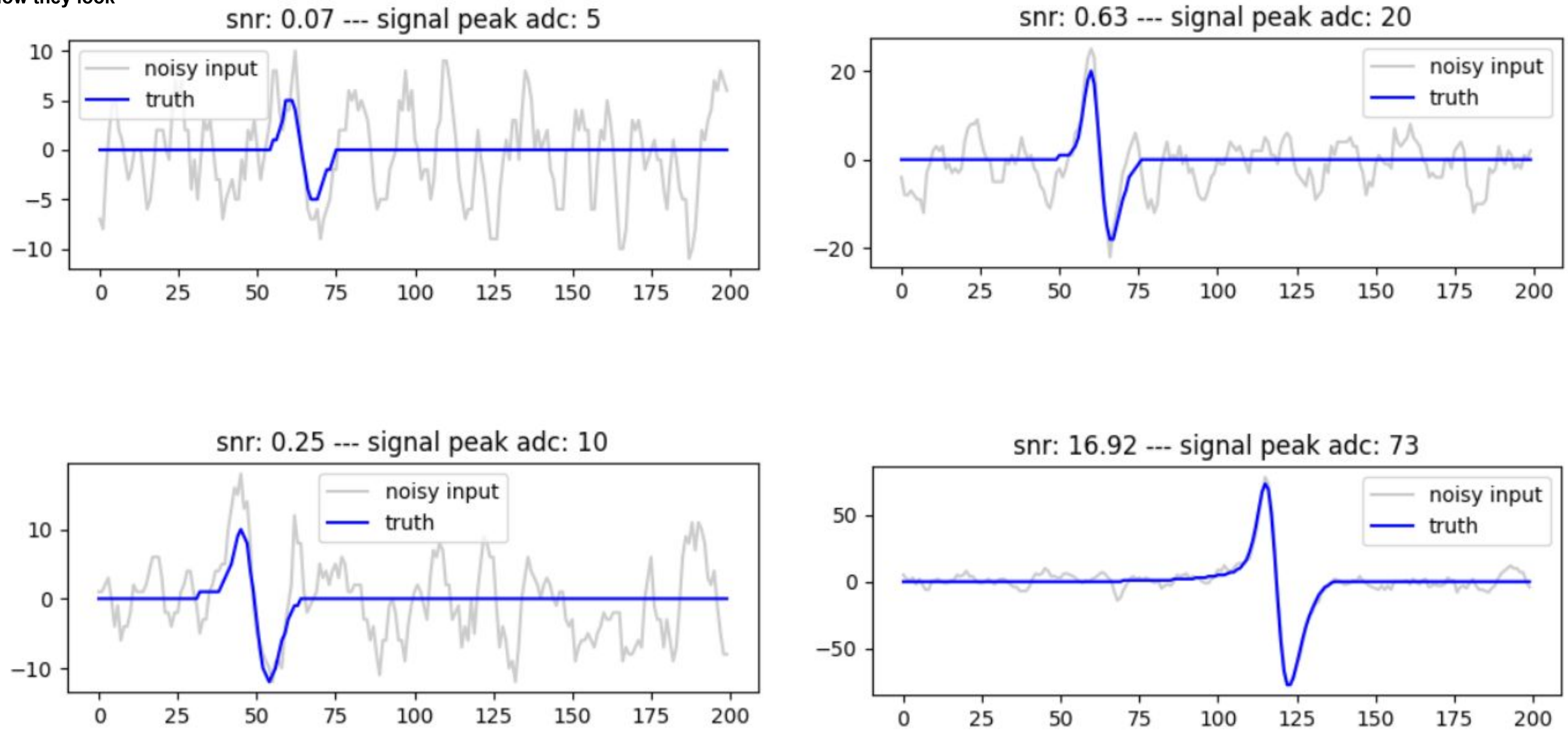






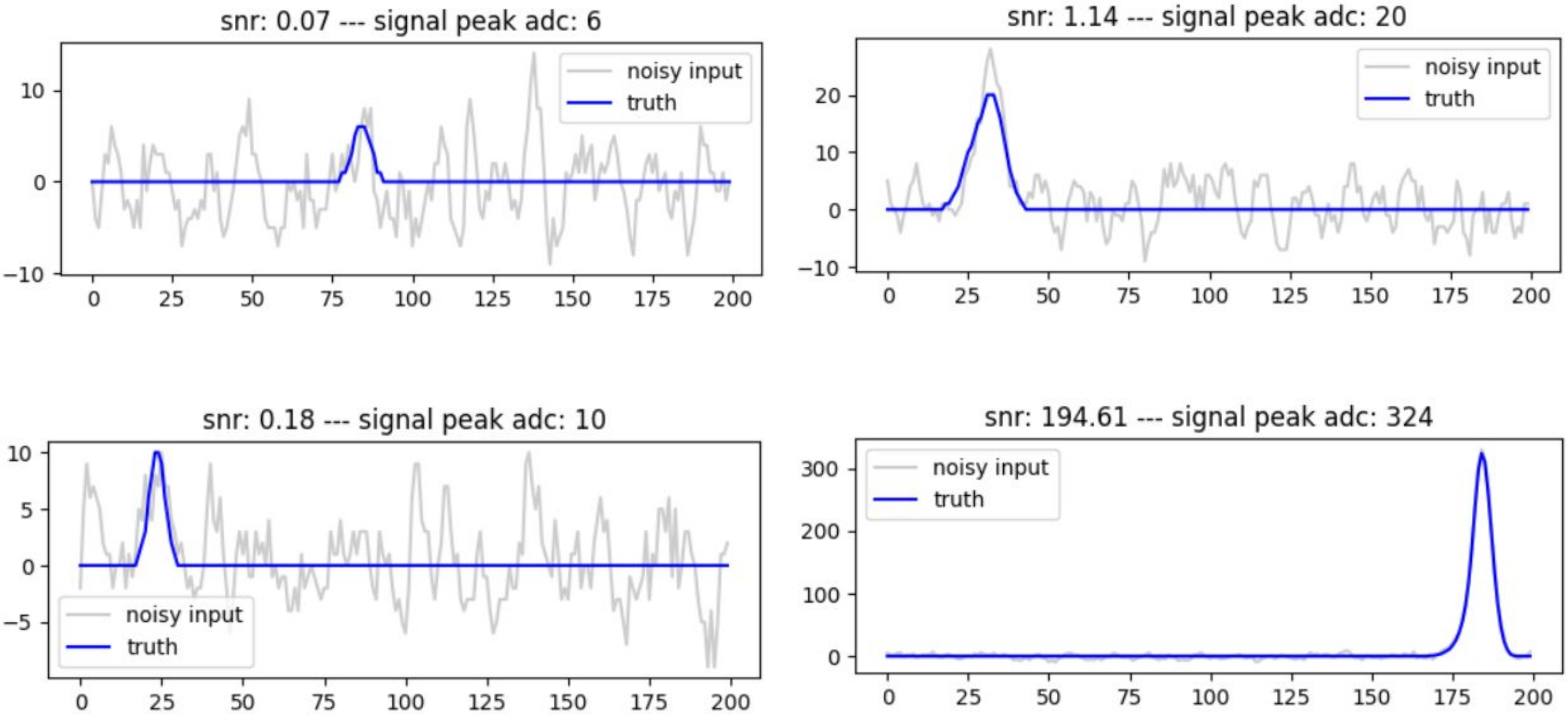
Induction plane (U) - NOISE

How they look



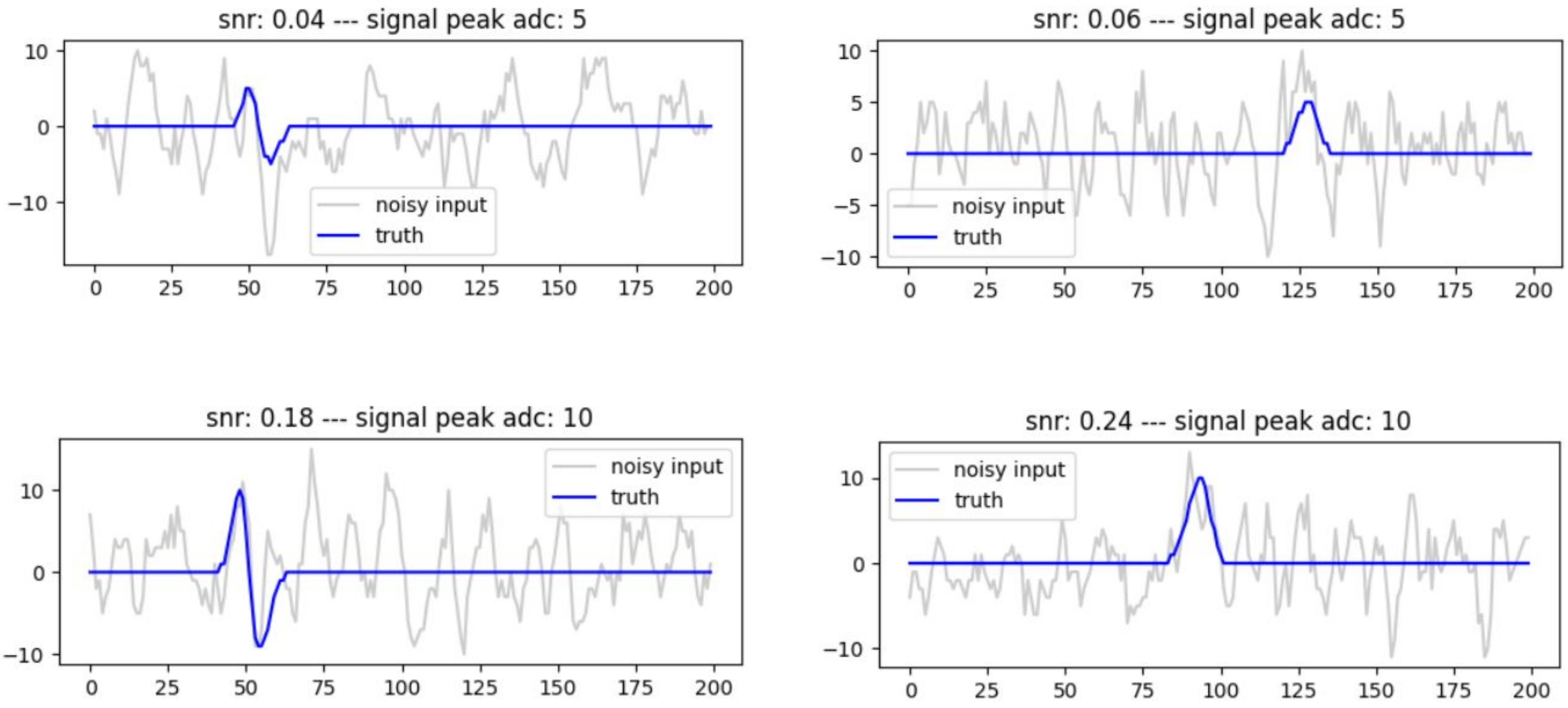
Induction plane (U) - NU_ES/NU_CC

How they look



Collection plane (Z) - NU_ES/NU_CC

How they look



Plane U **Plane Z**

Ar39 Samples

Last thing about dataset

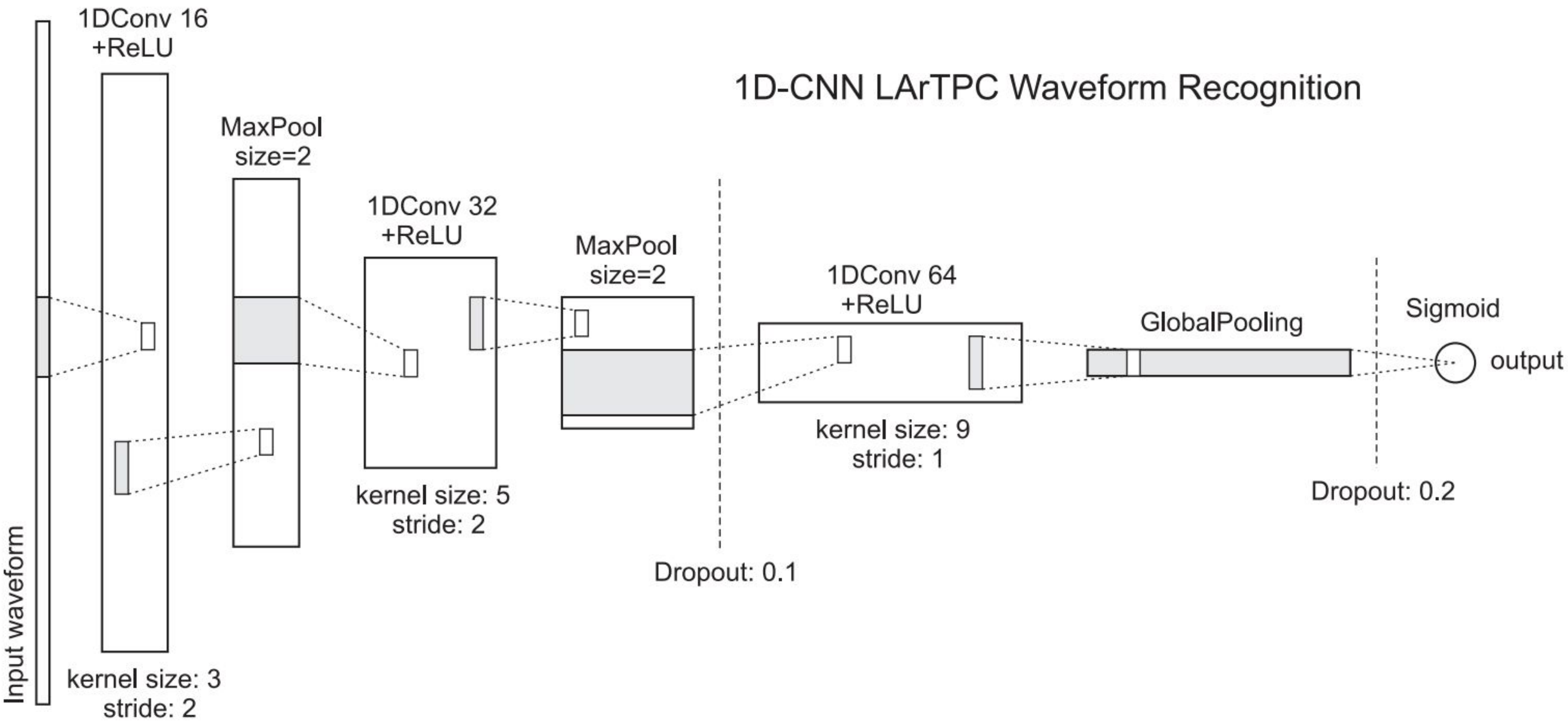
ADC range we use for clean signals

- NU_ES/NU_CC signals have ADCs ranging from 5 to ~2000
- Ar39 signals have ADCs ranging from 5 to ~33

Noise:

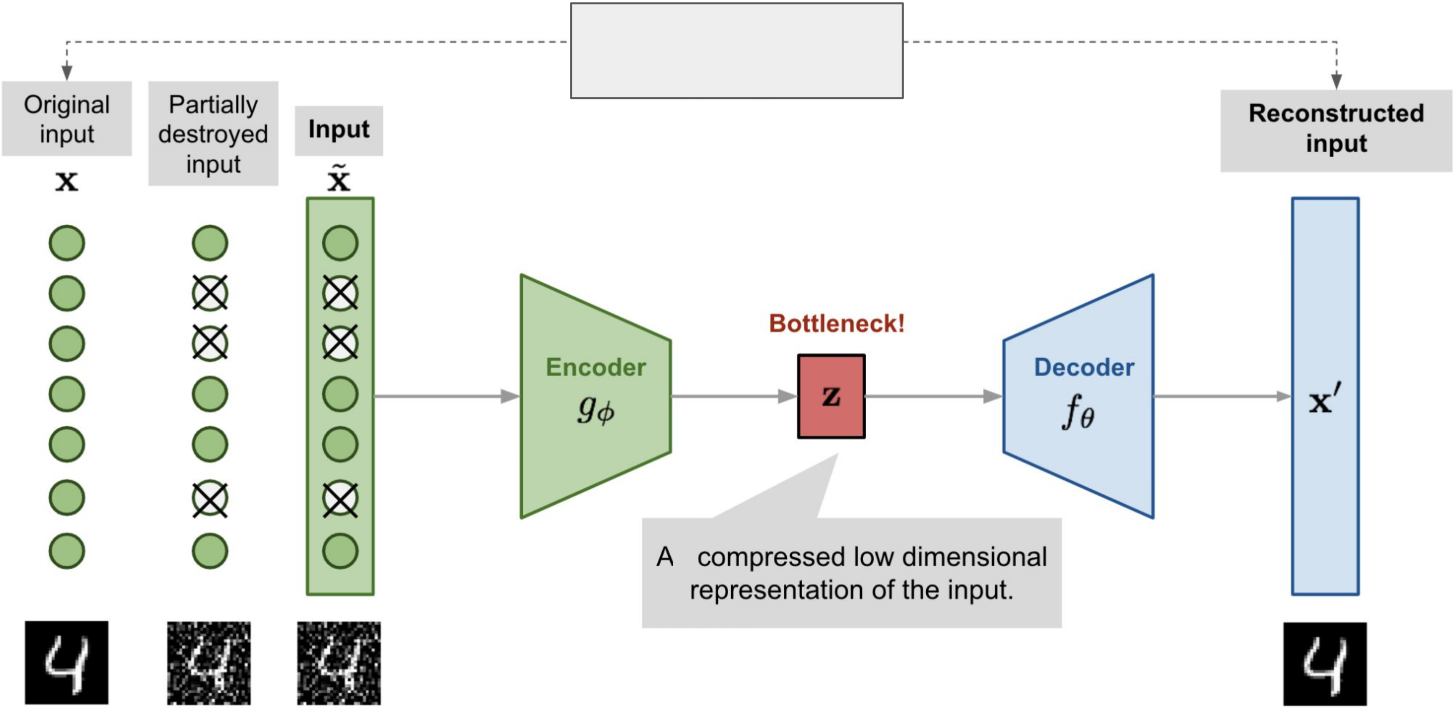
- Induction planes (U, V)
  - Average adc level: 12.43
  - RMS: 12.56
- Collection plane (Z)
  - Average adc level: 10.88
  - RMS: 10.99

1D-CNN LArTPC Waveform Recognition



[More on in-storage computing + FPGA implementation of 1-DCNN here]

Autoencoder based on trained 1DCNN weights

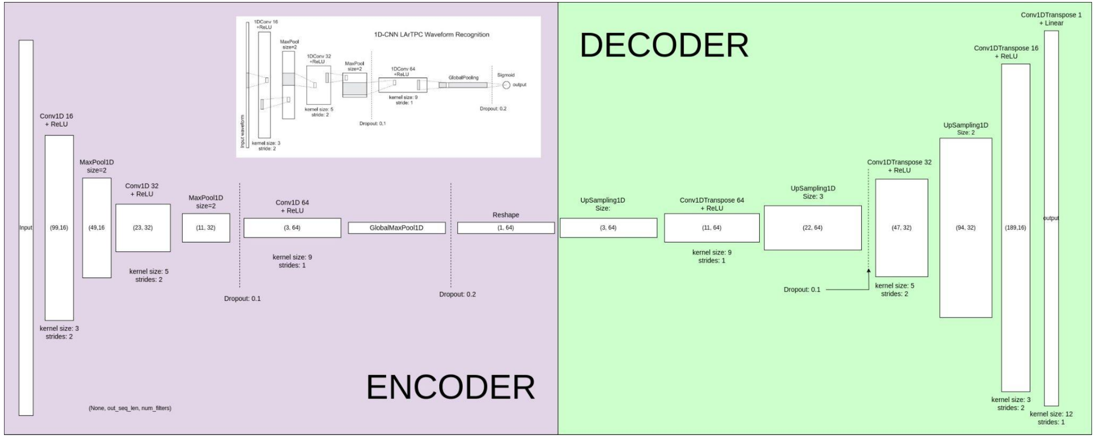


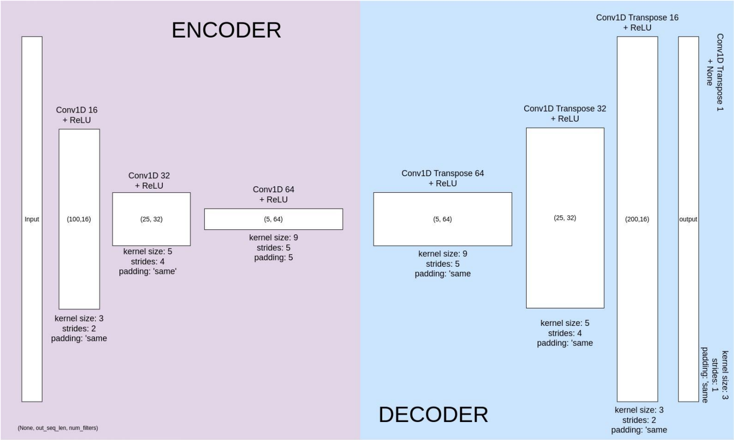
ROI from 1D CNN → input

Ideal: noise/background not reconstructed, only signal

2 Iterations of the AutoEncoder

- AE based on 1DCNN, use CNN weights (MODEL 1)
- AE based on 1DCNN without pooling layers (MODEL 2)





Model 1 (with pooling layers)

Optimizer: adam
Learning rate: 0.001
Loss function: MSE
Batch size: 2048
Epochs: 1000 with early stopping

70,097 parameters

- Trainable: 48,945
- Non-trainable: 21,152

Induction plane (U):

NU_CC/NU_ES + Electronic Noise

Training set:
~40k:40k noise:signal

Validation set:
~10k:10k: noise:signal

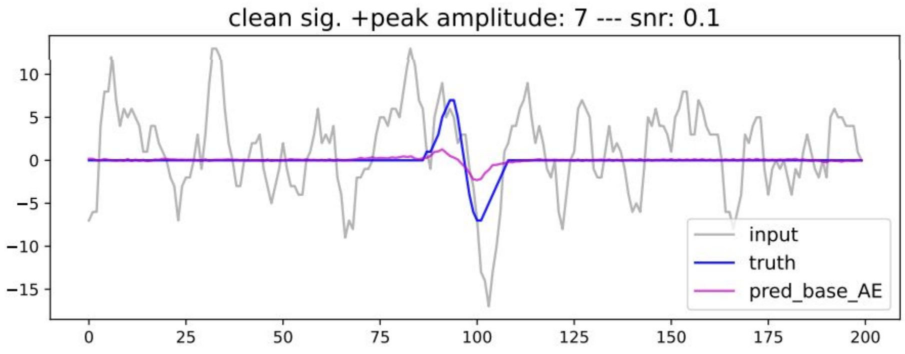
Testing set:
~50k:50k noise:signal

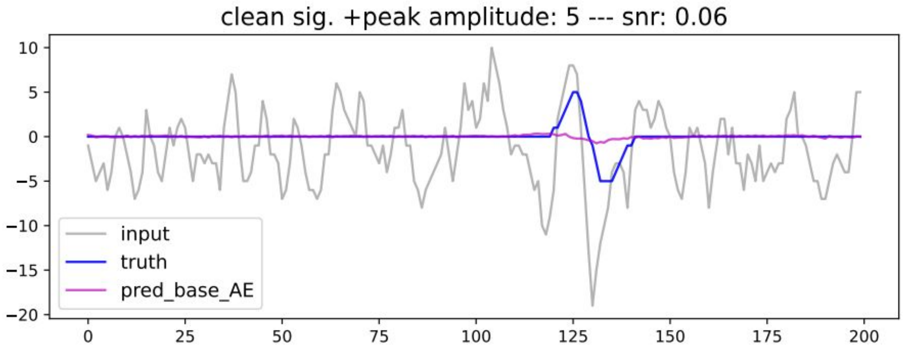
True Positive Rates(TPr) and Noise Rejection (bkg_rej)

Model 1 (pooling layers, cnn weights)

Model 1	ADC_5_7	ADC_8_10	ADC_11_13	ADC_14_16	ADC_17_19	ADC_20_22	ADC_gt_22	bk_rej (%)
TPr	2.7	15.3	43.0	73.8	92.0	98.3	99.9	99.9

Struggling at low ADC count samples





Noise Rejection (bkj_rej) is determined by feeding the network 200k pure electronic noise samples.

A rejection is when the peak amplitude of the model's predicted wave is less than 5

True Positive rate is calculated by feeding the model samples from the testing set.

A TP is when the model reconstruct a signal AND the peak amplitude of the model's predicted wave is >= 5

Model 2 (no pooling layers)

Optimizer: adam
Learning rate: 0.001
Loss function: MSE
Batch size: 12800
Epochs: 1000 with early stopping

69,953 parameters

- Trainable: 69,953
- Non-trainable: 0

Induction plane (U):

Training Set: Varies, will mention in each slide

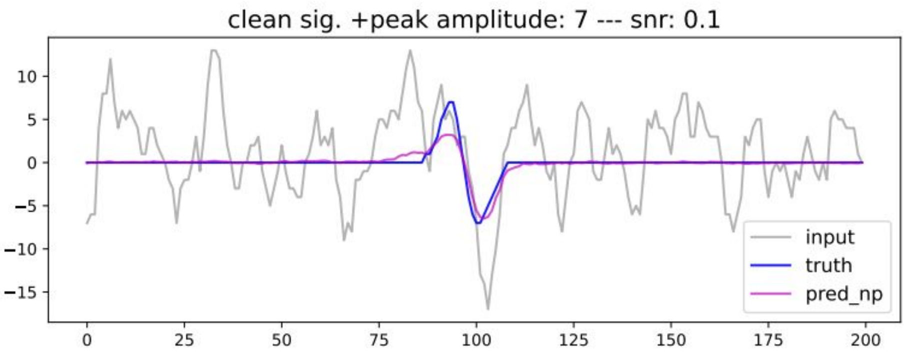
Testing set:

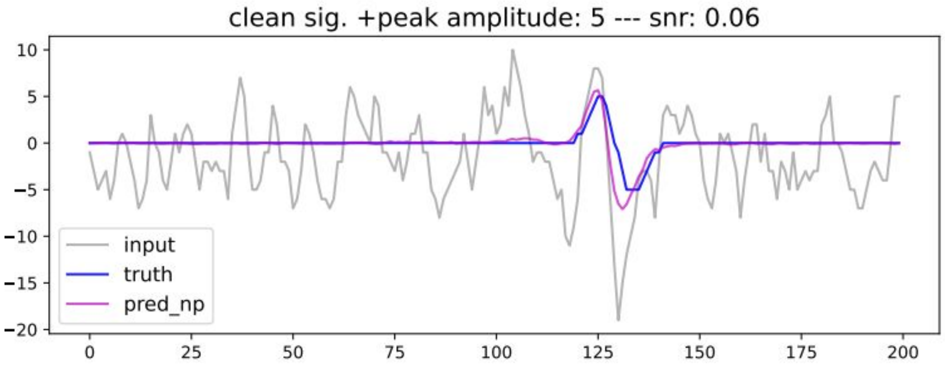
~50k:50k noise:signal (NU_CC/NU_ES samples)

True Positive Rates(TPr) and Noise Rejection (bkg_rej)

Model 2 (no pooling layers)

Model 2	ADC_5_7	ADC_8_10	ADC_11_13	ADC_14_16	ADC_17_19	ADC_20_22	ADC_gt_22	bk_rej (%)
(1) TPr (%)	11.5	42.8	77.2	96.1	99.6	100	100	99.6

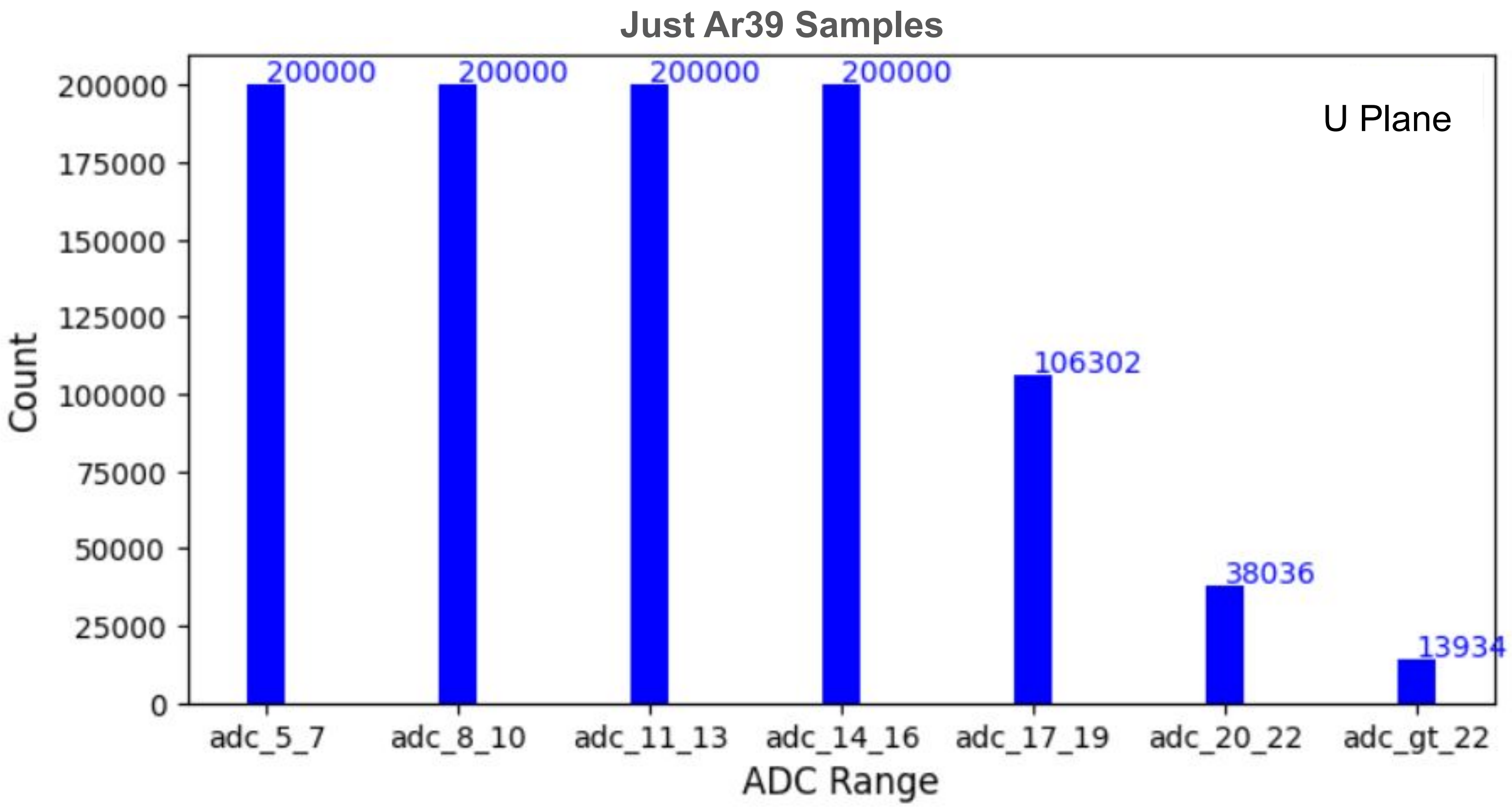




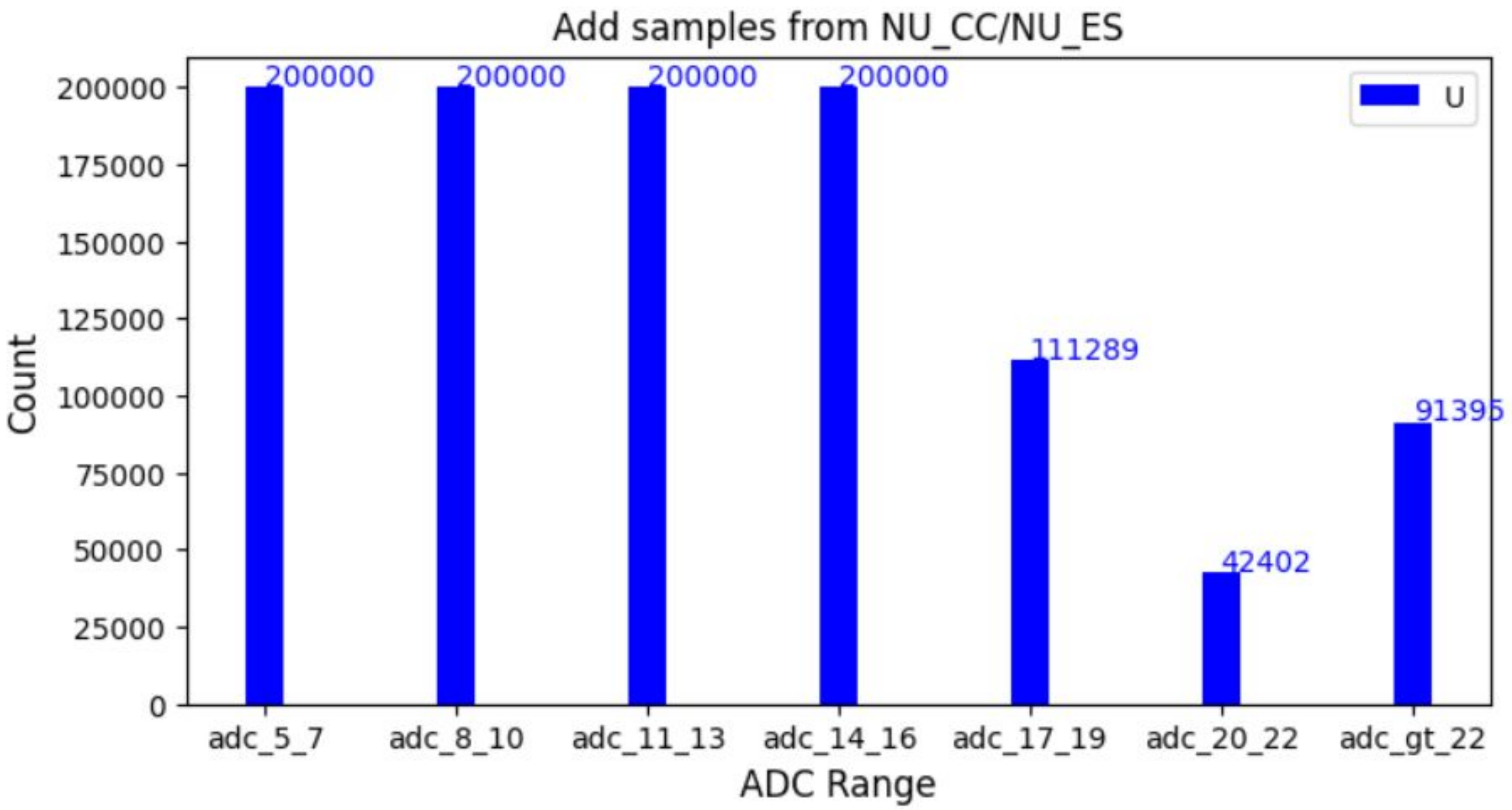
Improvement at lower ADC count range

Trained with nu_es/nu_cc samples - test on nu_es/nu_cc samples

Train with Ar39



Train with Ar39 + NU_CC/NU_ES



In the following slide… results for model 2

(1) Trained with nu_es/nu_cc samples - test on nu_es/nu_cc samples

(2) Trained with ar39 samples + nu_cc/nu_es at ADC > 16 (full) - test on nu_es/nu_cc samples

(3) Trained with ar39 samples (adc 5_10) - test on nu_es/nu_cc samples

(4) Trained with ar39 samples (adc 5_13) - test on nu_es/nu_cc samples

True Positive Rates(TPr) and Noise Rejection (bkg_rej)

Model 2 (no pooling layers)

(1) Trained with nu_es/nu_cc samples - test on nu_es/nu_cc samples
(2) Trained with ar39 samples + nu_cc/nu_es at ADC > 16 (full) - test on nu_es/nu_cc samples
* (3) Trained with ar39 samples (adc 5_10) - test on nu_es/nu_cc samples ***(most promising)***
(4) Trained with ar39 samples (adc 5_13) - test on nu_es/nu_cc samples

Model 2	ADC_5_7	ADC_8_10	ADC_11_13	ADC_14_16	ADC_17_19	ADC_20_22	ADC_gt_22	bk_rej (%)
(1) TPr (%)	11.5	42.8	77.2	96.1	99.6	100	100	99.6
(2) TPr (%)	21.4	56.2	84.2	97.1	99.5	99.9	100	96.95
* (3) TPr (%)	18.3	51.8	81.0	96.0	99.0	99.8	100	97.89
(4) TPr (%)	22.0	56.9	84.1	97.2	99.4	99.9	100	96.83

Compared to:

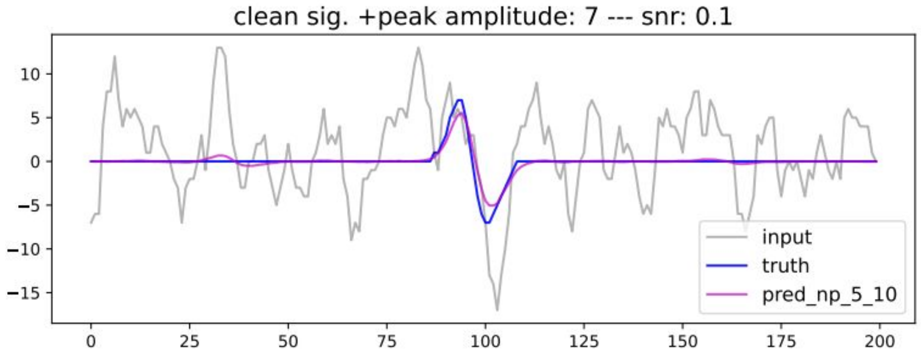
Model 1	2.7	15.3	43.0	73.8	92.0	98.3	99.9	99.9

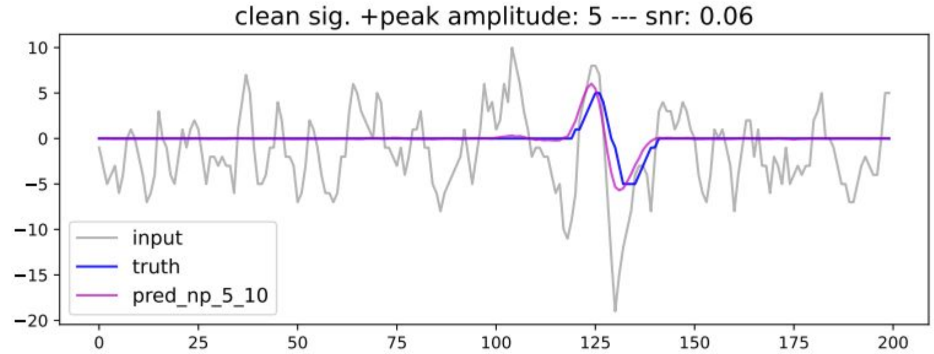
Note again: *TPr* and *bk_rej* here are based on a threshold cut of the model's predictions

Predicted wave is considered noise if: `max(pred_wave) < 5`

a3d3 Duke University

Model 2 (no pooling layers)





(3) Trained with ar39 samples (adc 5_10) - test on nu_es/nu_cc samples

Reconstruction improved!

Summary

- Simple changes in model architecture can vastly alter the latent space and performance of the decoder
- Important to have balanced dataset across ADC ranges
    - Ar39 samples improved model dramatically

Next:

- Continue to optimize model 2
- **Explore:**
    - Curriculum Learning
    - RNN based Denoising Autoencoder

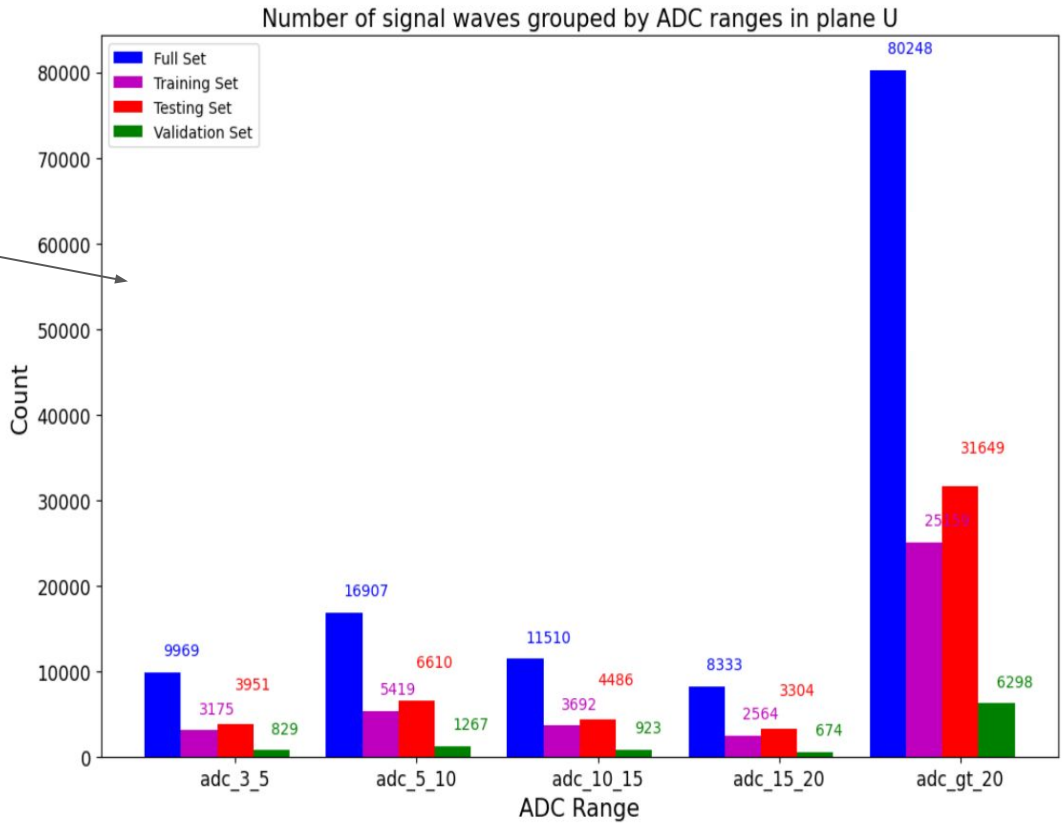
BACKUP

Dataset

1. dune_train_v2
   a. **Noise:** electronic noise
   b. **Signals:** nu_cc/nu_es

2. Ar39 events in LAr
   a. **noise:** 1.4 million samples
   b. 10 million samples per wireplane

Ar39 samples were produced to have a more balanced number of samples across ADC counts

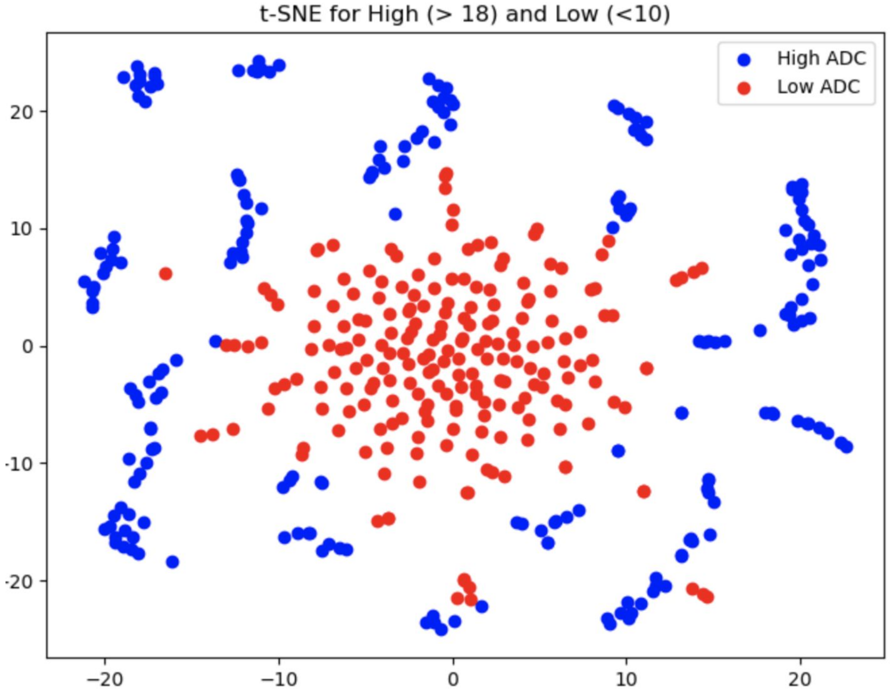


Latent Space Analysis: With Max Pool

MODEL 1

Plot shows how two groups (low adc & high adc) are seen at the latent space.

This information is what’s passed to the decoder.



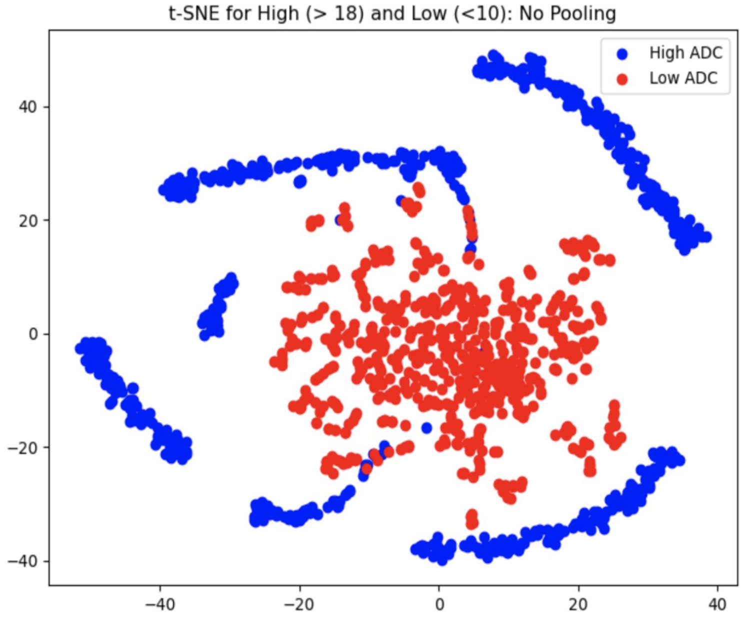
Clusterings of blue dots tell us the model is able to pick up features and group them together

We can expect the model to struggle for samples with adc < 10

tSNE plot for NUES/NUCC windows in dune_train_v2

Latent Space Analysis: No Pooling

MODEL 2



Slight change in the model's architecture has improved the model at the latent space

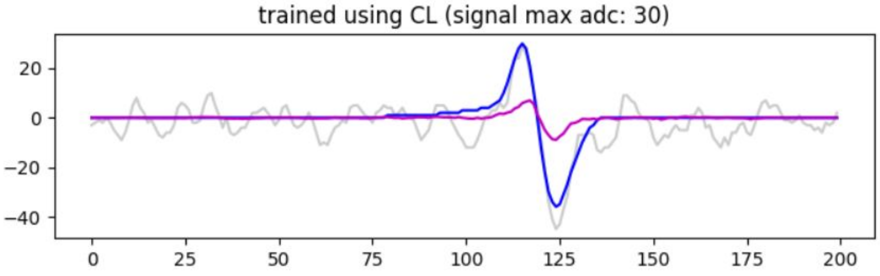
tSNE plot for NUES/NUCC windows in dune_train_v2

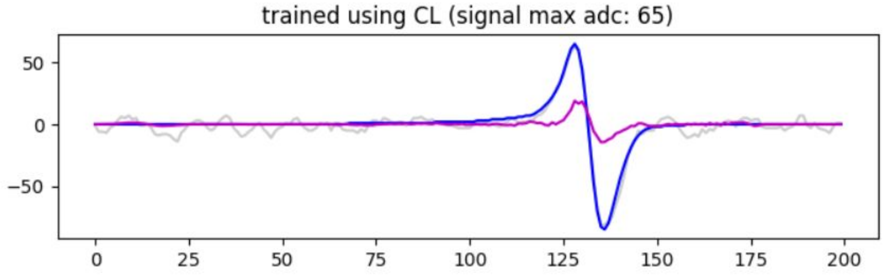
Exploring: Curriculum Learning

Start training with easy samples, and increase difficulty

So far… so ***not*** good

- Model is having a hard time capturing amplitudes (even for high ADC count samples)



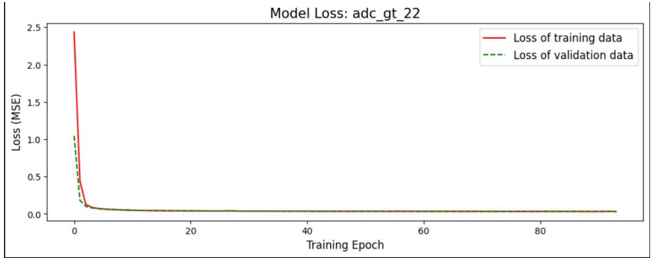


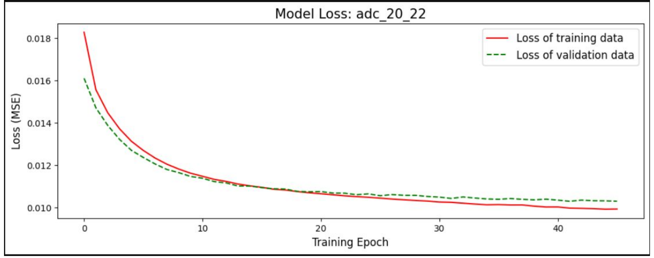
['adc_5_7', 'adc_8_10', 'adc_11_13', 'adc_14_16', 'adc_17_19', 'adc_20_22', 'adc_gt_22']

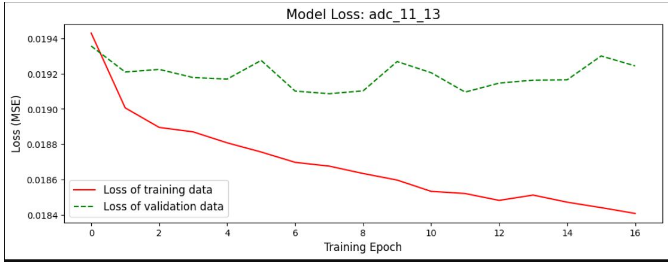
- For every adc range above, get 200k samples (max 1.4 million samples given 1.4 million noise samples)
- For the two induction planes U, V → had to grab samples from nu_cc + nu_es dataset
- Split EACH adc group into 50:50 training/testing, and mix in the same number of noise samples for each set
- Split training set into 80:20 training/validation
- MEAN + STD
    - I am thinking of taking the mean + std the training ACROSS the adc_groups

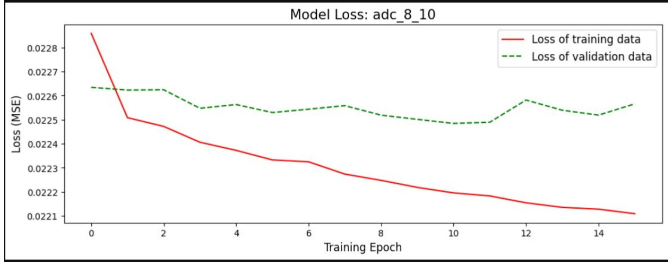
Saving the processed data for later use

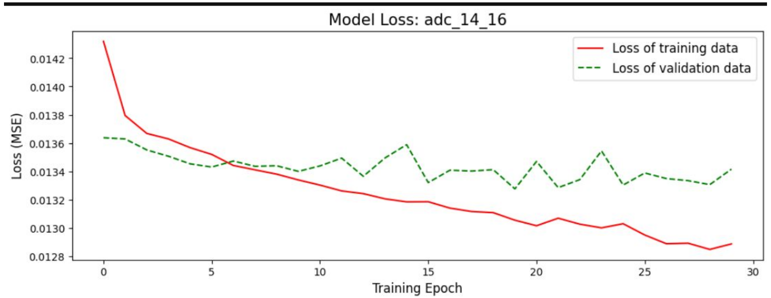
```
grp0_ = np.array([[x_train_5_7, y_train_5_7], [x_valid_5_7, y_valid_5_7], [x_test_5_7, y_test_5_7]], dtype=object)
grp1_ = np.array([[x_train_8_10, y_train_8_10], [x_valid_8_10, y_valid_8_10], [x_test_8_10, y_test_8_10]], dtype=object)
grp2_ = np.array([[x_train_11_13, y_train_11_13], [x_valid_11_13, y_valid_11_13], [x_test_11_13, y_test_11_13]], dtype=object)
grp3_ = np.array([[x_train_14_16, y_train_14_16], [x_valid_14_16, y_valid_14_16], [x_test_14_16, y_test_14_16]], dtype=object)
grp4_ = np.array([[x_train_17_19, y_train_17_19], [x_valid_17_19, y_valid_17_19], [x_test_17_19, y_test_17_19]], dtype=object)
grp5_ = np.array([[x_train_20_22, y_train_20_22], [x_valid_20_22, y_valid_20_22], [x_test_20_22, y_test_20_22]], dtype=object)
grp6_ = np.array([[x_train_gt_22, y_train_gt_22], [x_valid_gt_22, y_valid_gt_22], [x_test_gt_22, y_test_gt_22]], dtype=object)

np.savez_compressed('/home/vlian/Workspace/curriculum_learning_processed_data/samples',
                    adc_5_7=grp0_, adc_8_10=grp1_, adc_11_13=grp2_,
                    adc_14_16=grp3_, adc_17_19=grp4_, adc_20_22=grp5_, adc_gt_22=grp6_)
```

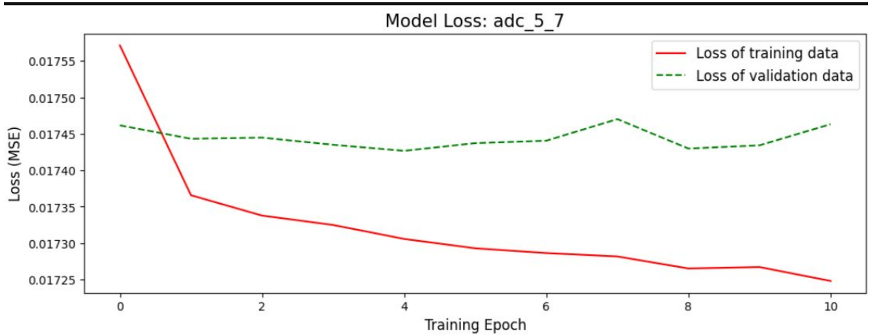










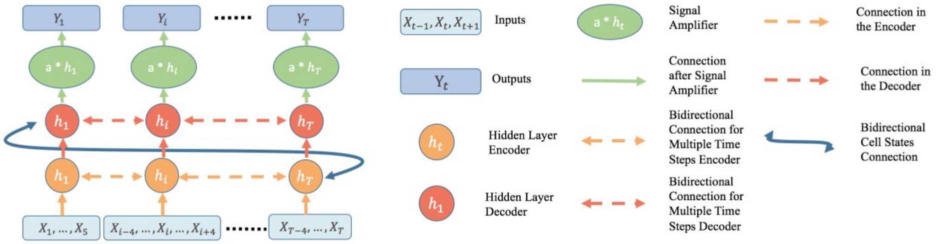


Exploring: Curriculum Learning

Exploring: EDRDAE for Gravitational Waves

- Denoising Autoencoder based on Bidirectional LSTMs

Reference to paper: [DENOISING GRAVITATIONAL WAVES WITH ENHANCED DEEP RECURRENT DENOISING AUTO-ENCODERS](#)

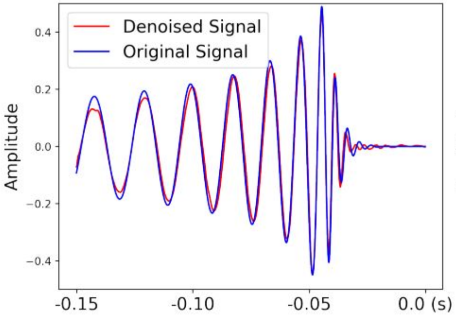


[1]

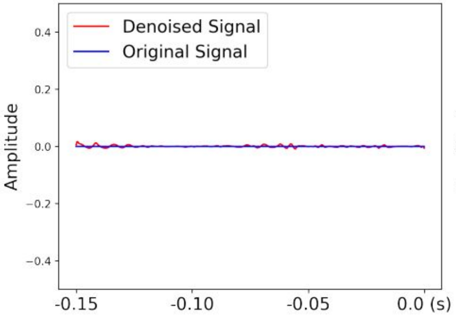
[We'd like to apply this to our dataset]

Have reproduced results! Working on implementing for our dataset

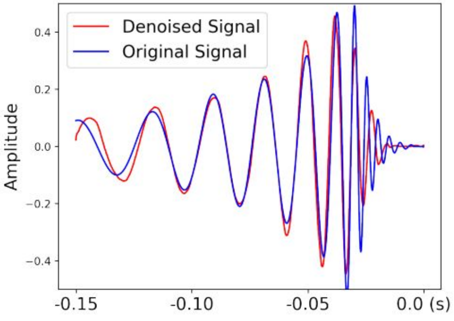
Paper from UIUC to denoise gravitational wave data at low SNR



(d) EDRDAE: quasi-circular

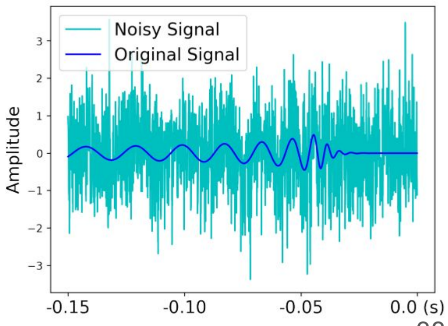


(e) EDRDAE: pure noise



(f) EDRDAE: eccentric signals

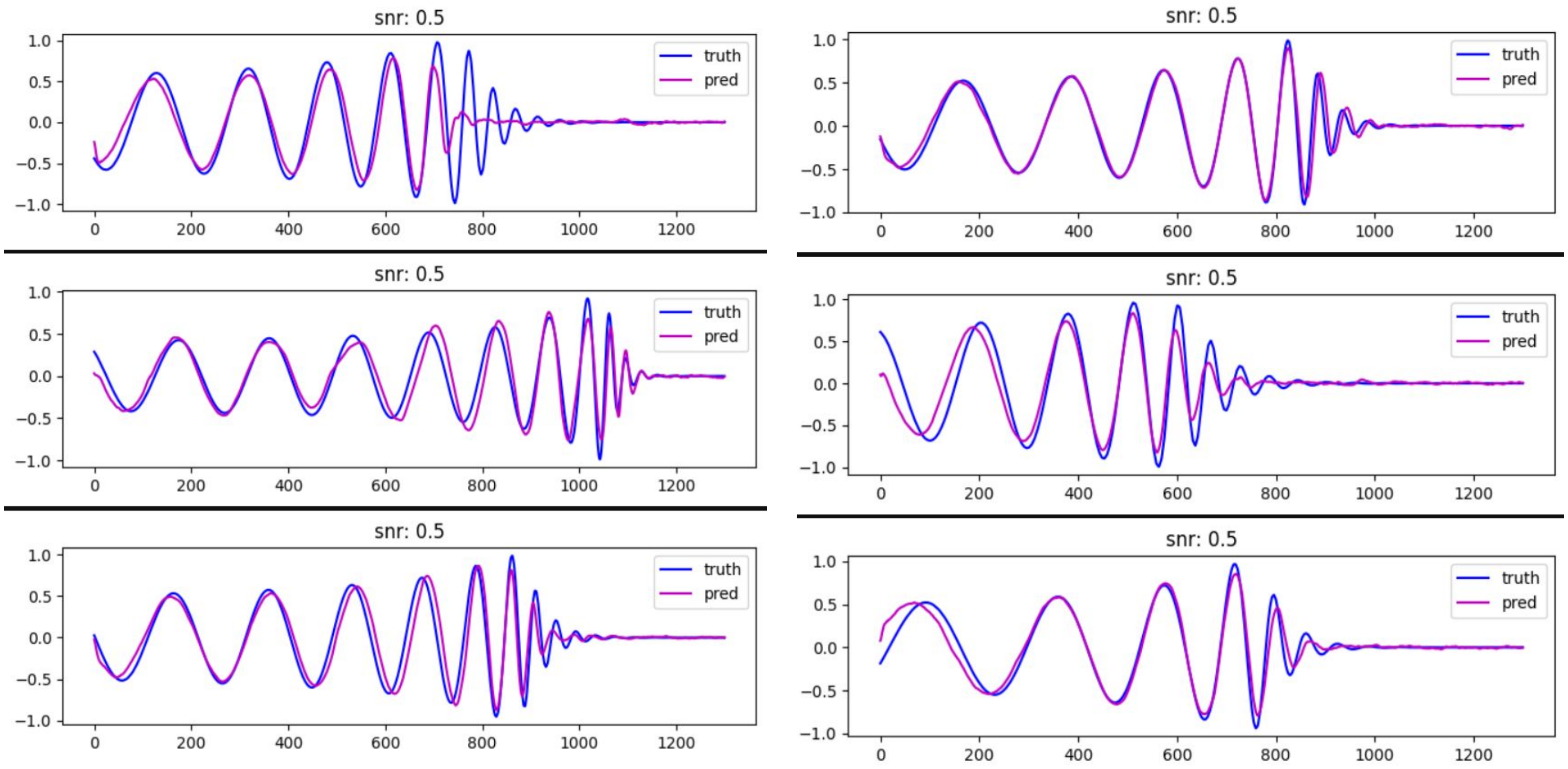
$$\mathrm{SNR}_{\mathrm{peak}} = \frac{\text{Maximum value of a given signal}}{\text{Standard deviation of noise}}$$



$\mathrm{SNR}_{\mathrm{peak}} = 0.5$

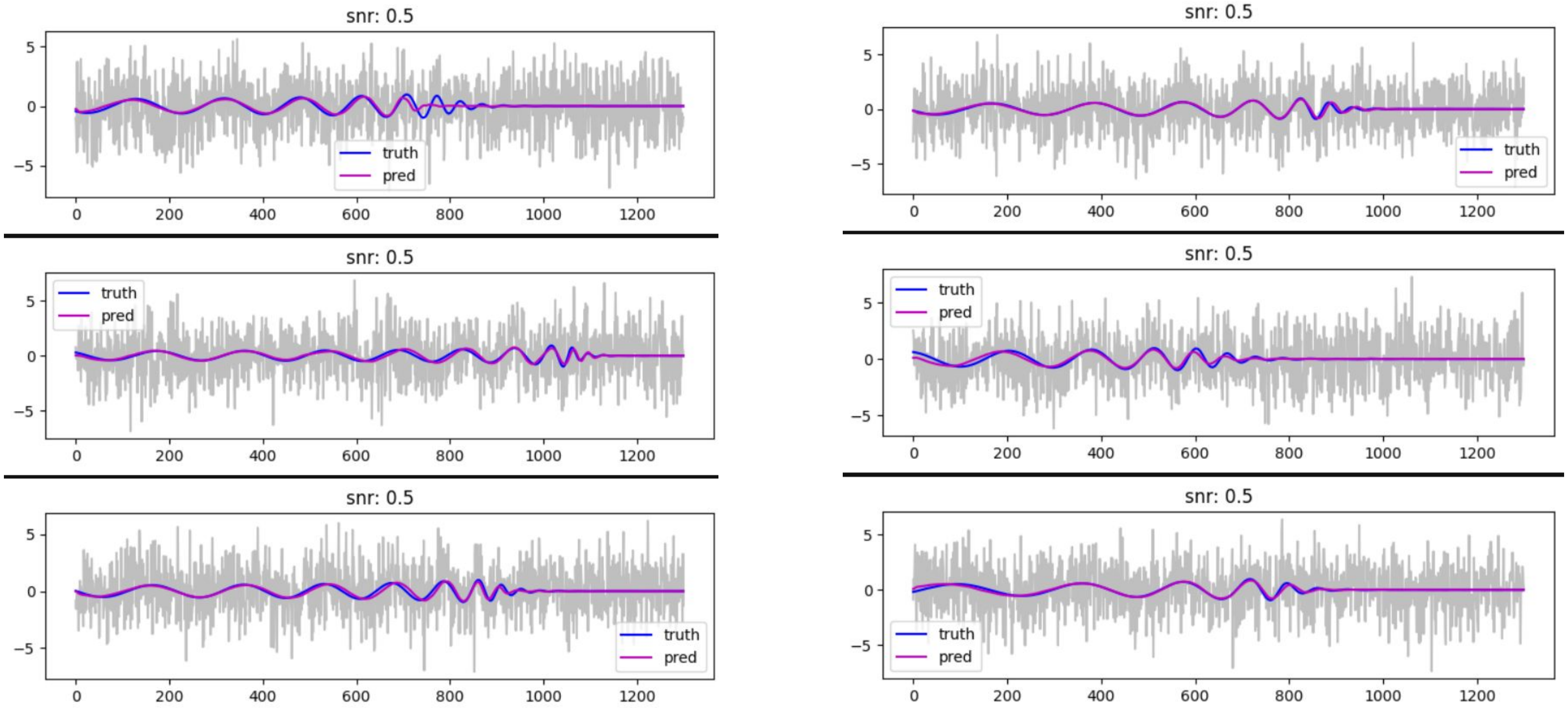
Reproducing results

Trained and tested on *GAUSSIAN NOISE*



Reproducing results

Trained and tested on *GAUSSIAN NOISE*



Model 2 (trained from scratch)

Optimizer: adam
Learning rate: 0.001
Loss function: MSE
Batch size: 2048
Epochs: 1000 with early stopping

70,097 parameters
- Trainable: 48,945
- Non-trainable: 21,152

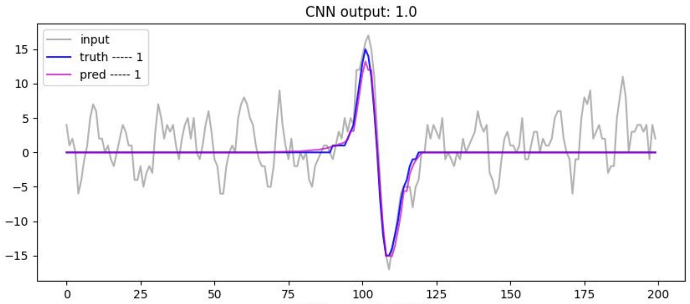
Induction plane (U):

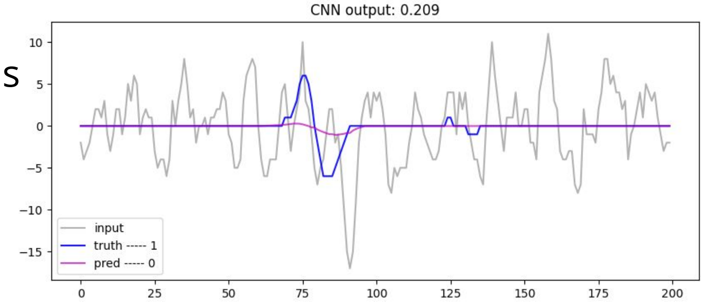
Training set:
~40k:40k noise:signal

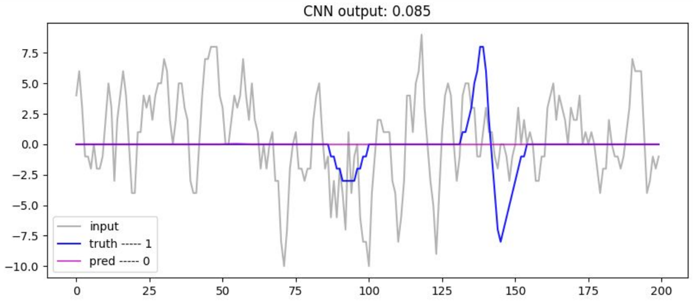
Validation set:
~10k:10k: noise:signal

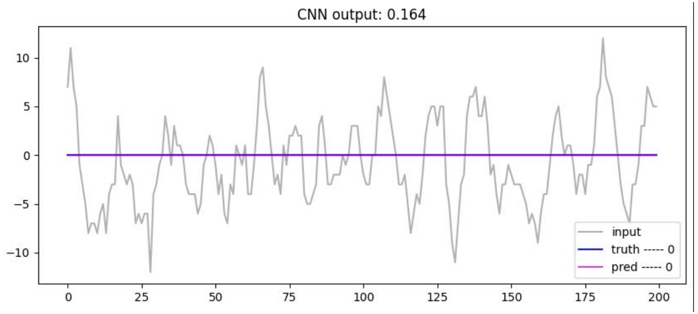
Testing set:
~50k:50k noise:signal

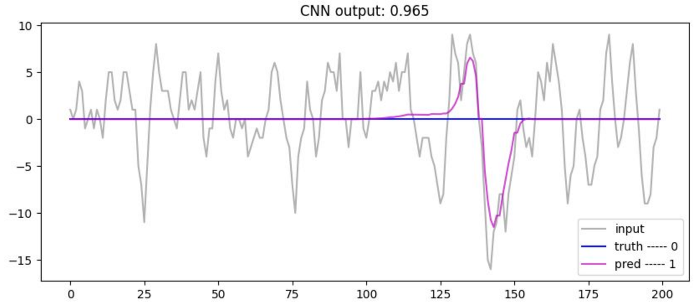
Has signals



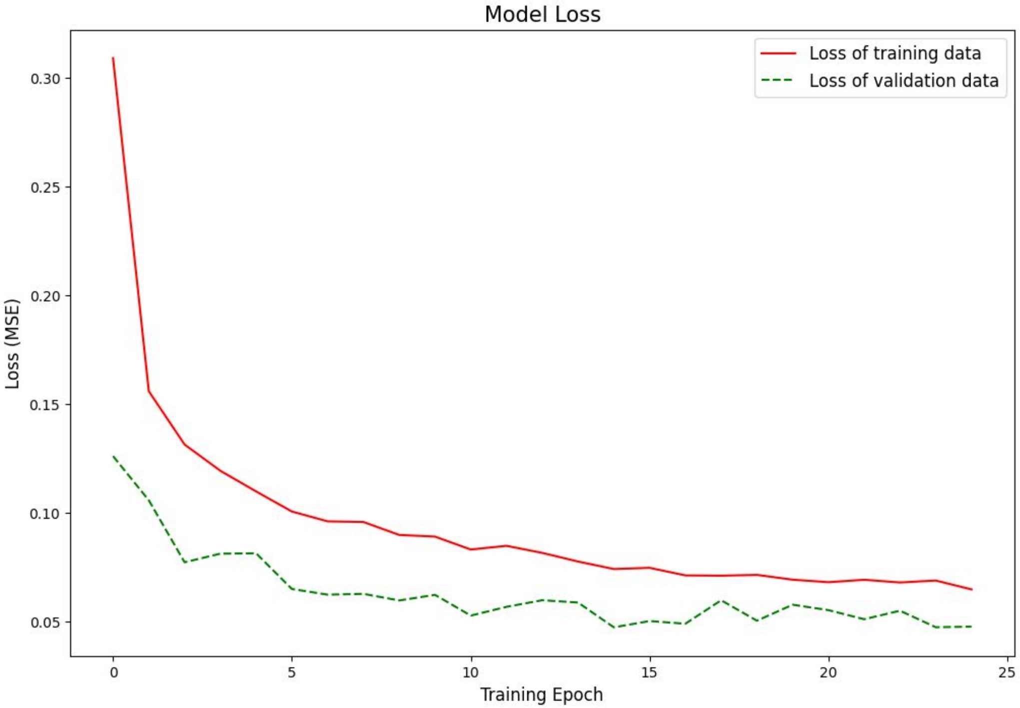


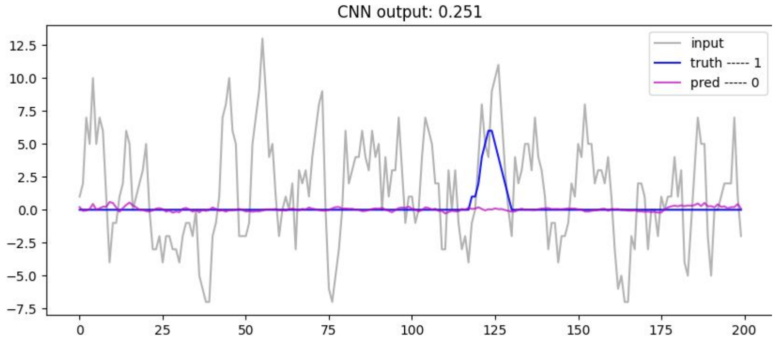


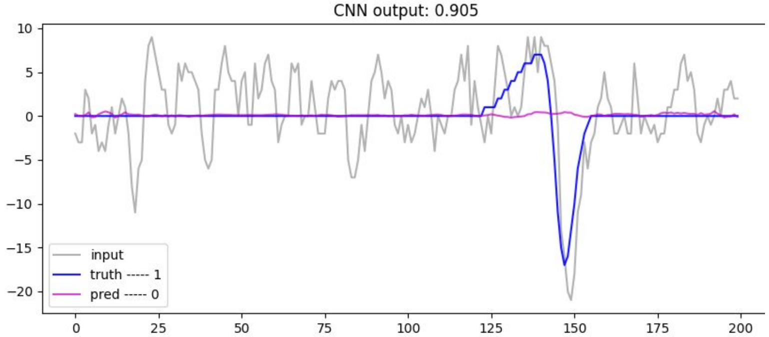




NOISE







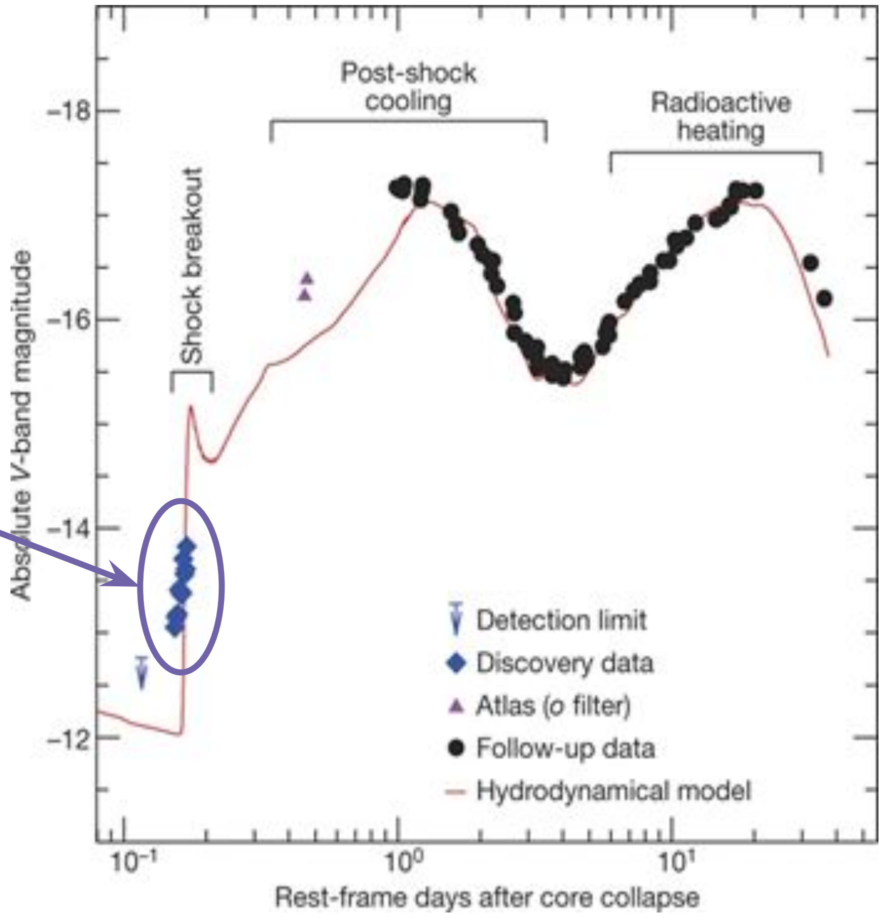
Model is too complex for the dataset.
Overfitting while training

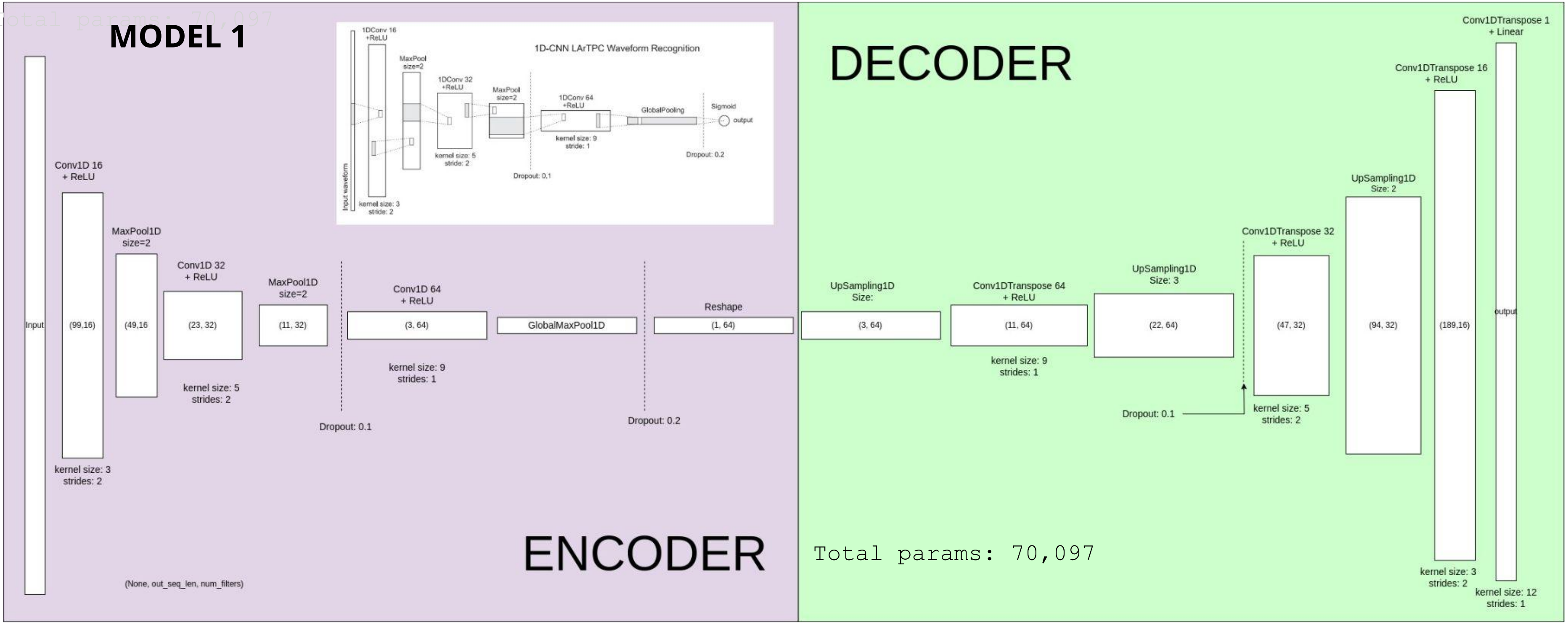
Background: Motivation

LEGACY SURVEY OF SPACE AND TIME @ RUBIN OBSERVATORY

- Scan entire visible sky every few nights for 10 years
- Unparalleled tool for study of transients – supernovas, kilonovas
- **Discovery Data**
  - Happens in the first few hours of a Supernova!
  - Only managed to observe due to amateur astronomer happening to be looking at the right spot!

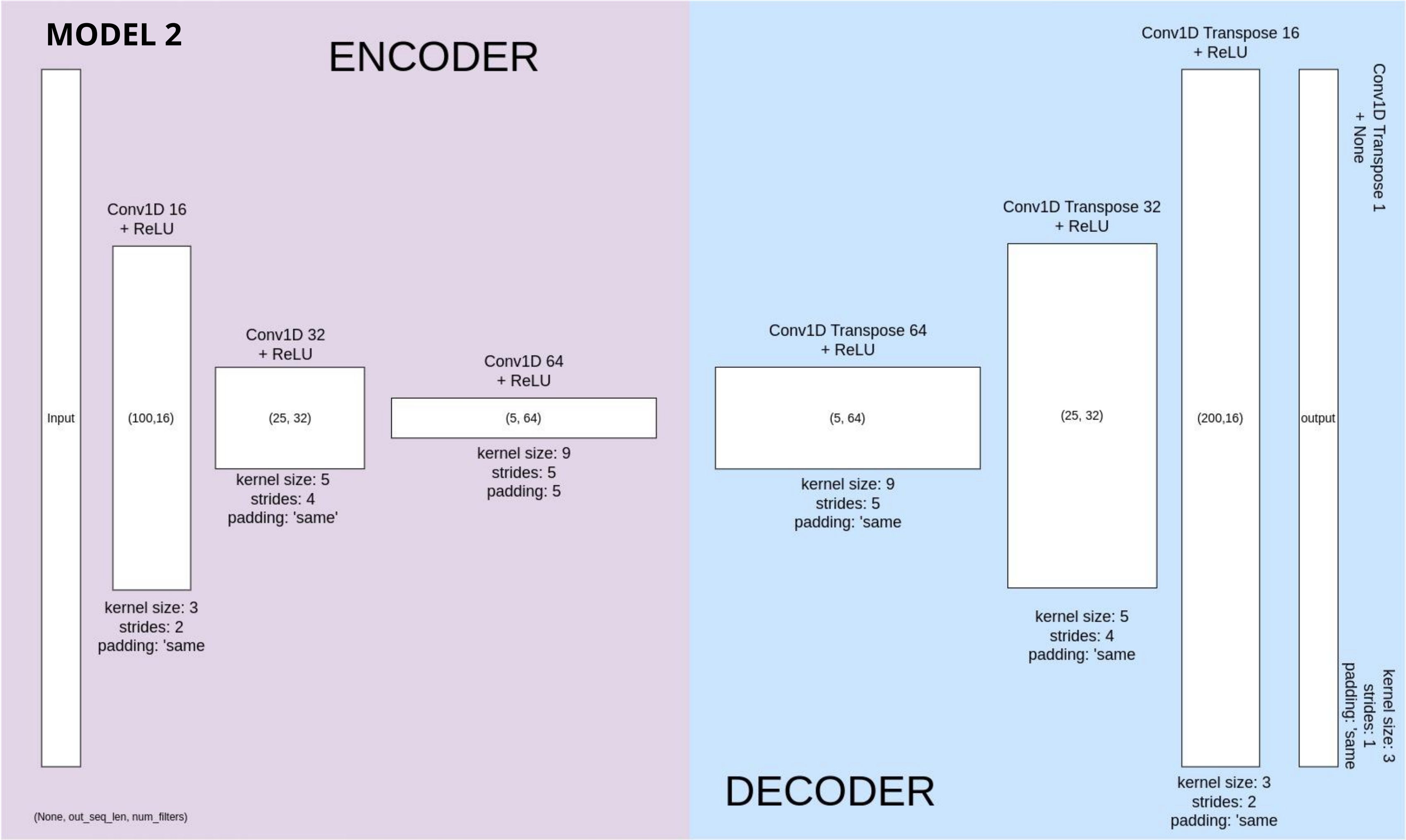
Time is *critical* for some events, so we can perform Multi Messenger Astronomy (MMA) to coordinate different instruments, like the LSST, to better observe and understand these events in real time



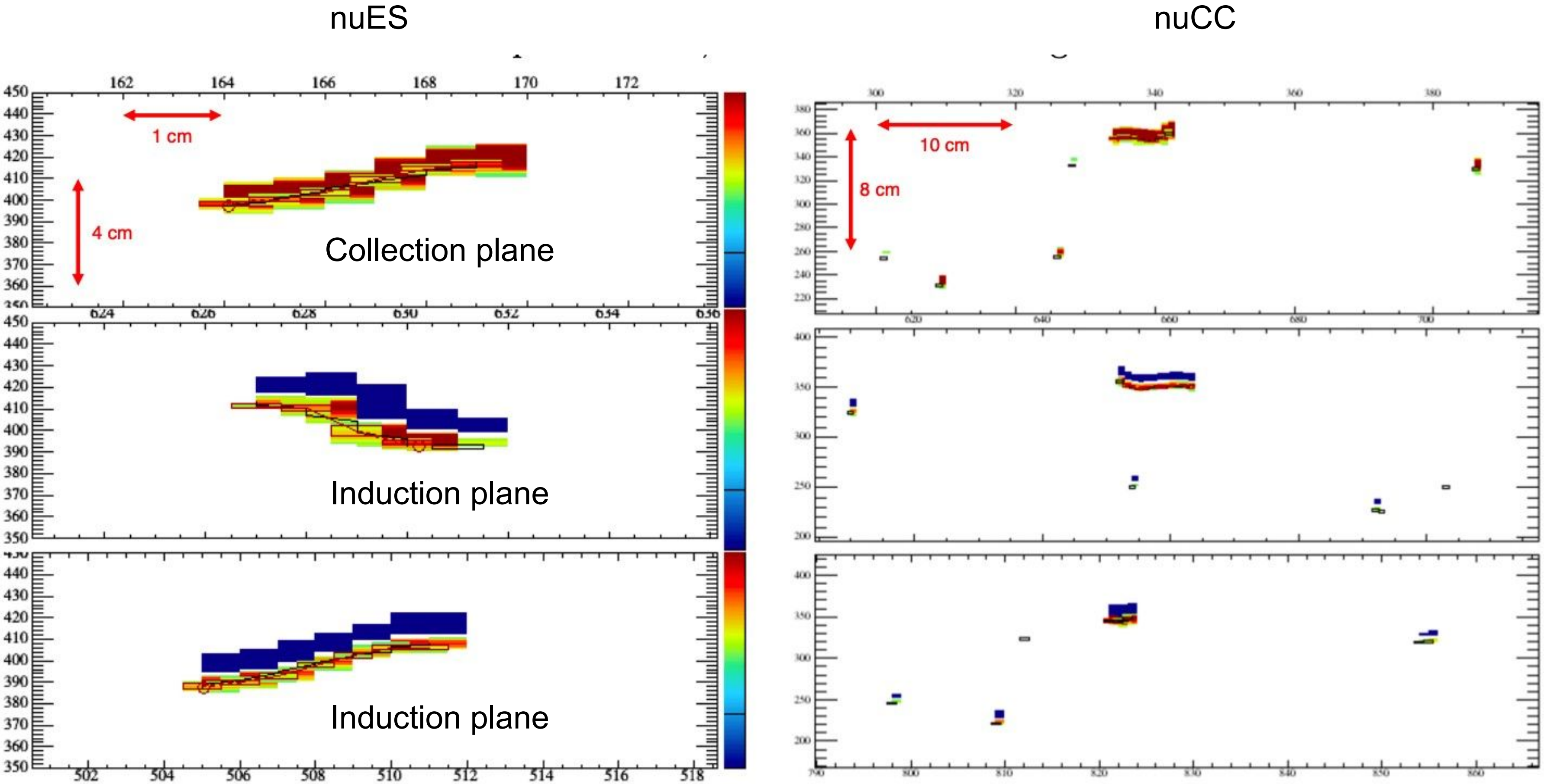


Encoder: is **original 1D-CNN** *without last layer with sigmoid function* (replaced with Reshape layer)

Decoder: Mirrored* version of encoder, ***Conv1D Transpose*** used inplace of Conv1D. Includes an additional Conv1D Transpose at the end


MODEL 2
ENCODER
Input
Conv1D 16
+ ReLU
(100,16)
kernel size: 3
strides: 2
padding: 'same
Conv1D 32
+ ReLU
(25, 32)
kernel size: 5
strides: 4
padding: 'same'
Conv1D 64
+ ReLU
(5, 64)
kernel size: 9
strides: 5
padding: 5
(None, out_seq_len, num_filters)
DECODER
Conv1D Transpose 64
+ ReLU
(5, 64)
kernel size: 9
strides: 5
padding: 'same
Conv1D Transpose 32
+ ReLU
(25, 32)
kernel size: 5
strides: 4
padding: 'same
Conv1D Transpose 16
+ ReLU
(200,16)
kernel size: 3
strides: 2
padding: 'same
Conv1D Transpose 1
+ None
output
kernel size: 3
strides: 1
padding: 'same

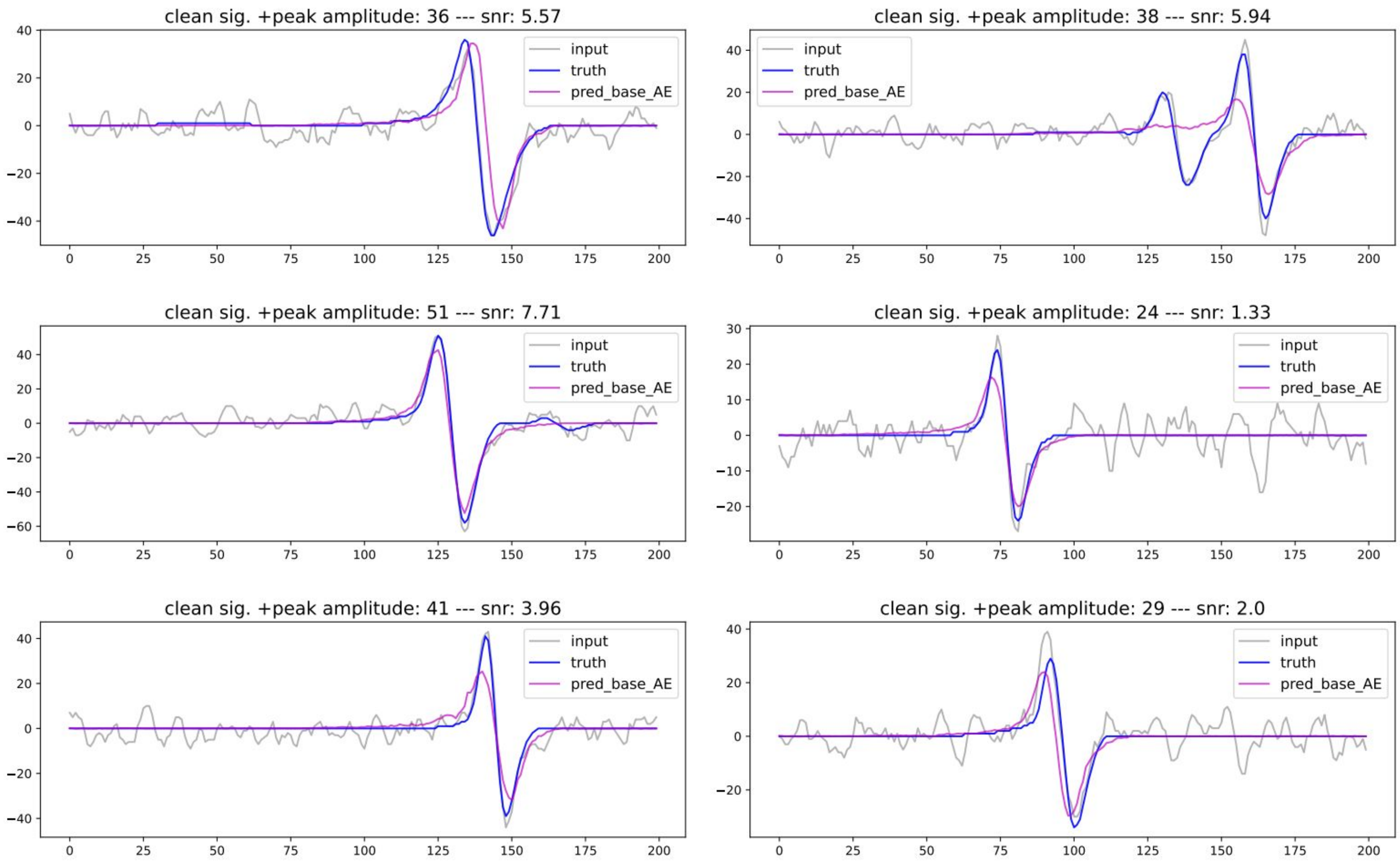
2D projections of wire vs time



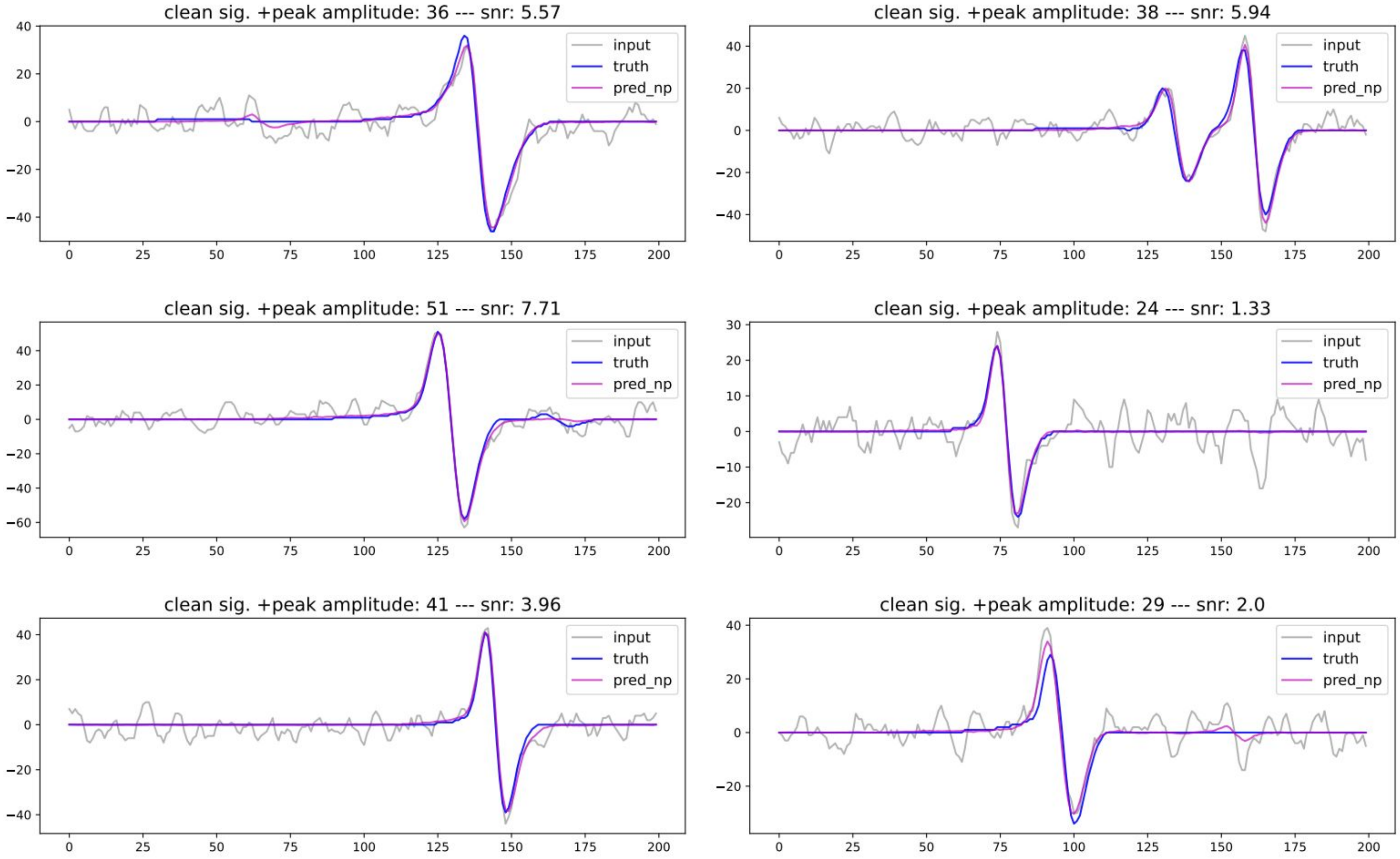
Too good to be true?

… yeah :/

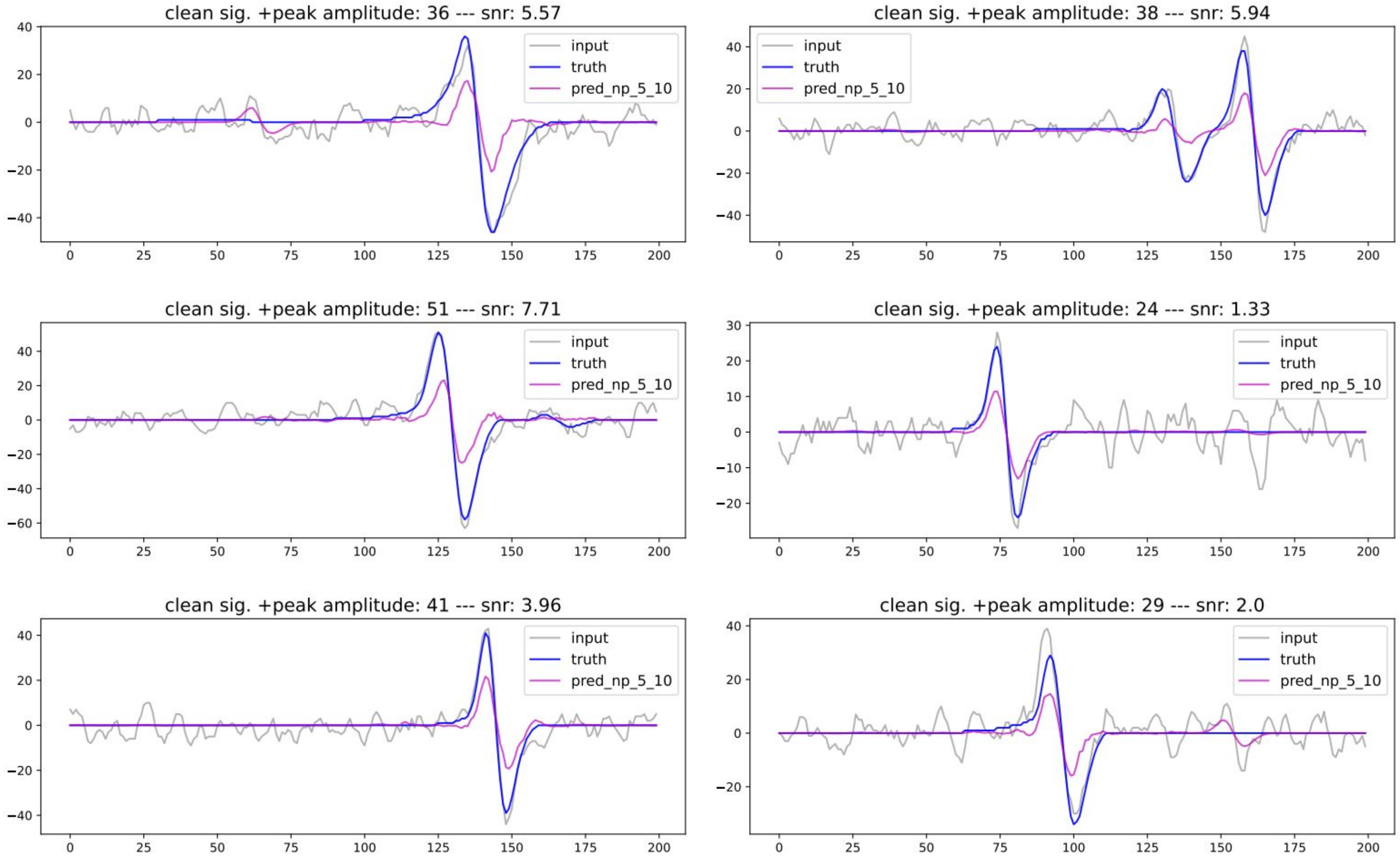
Higher ADC (ADC > 22)


clean sig. +peak amplitude: 36 --- snr: 5.57
clean sig. +peak amplitude: 38 --- snr: 5.94
clean sig. +peak amplitude: 51 --- snr: 7.71
clean sig. +peak amplitude: 24 --- snr: 1.33
clean sig. +peak amplitude: 41 --- snr: 3.96
clean sig. +peak amplitude: 29 --- snr: 2.0
input
truth
pred_base_AE

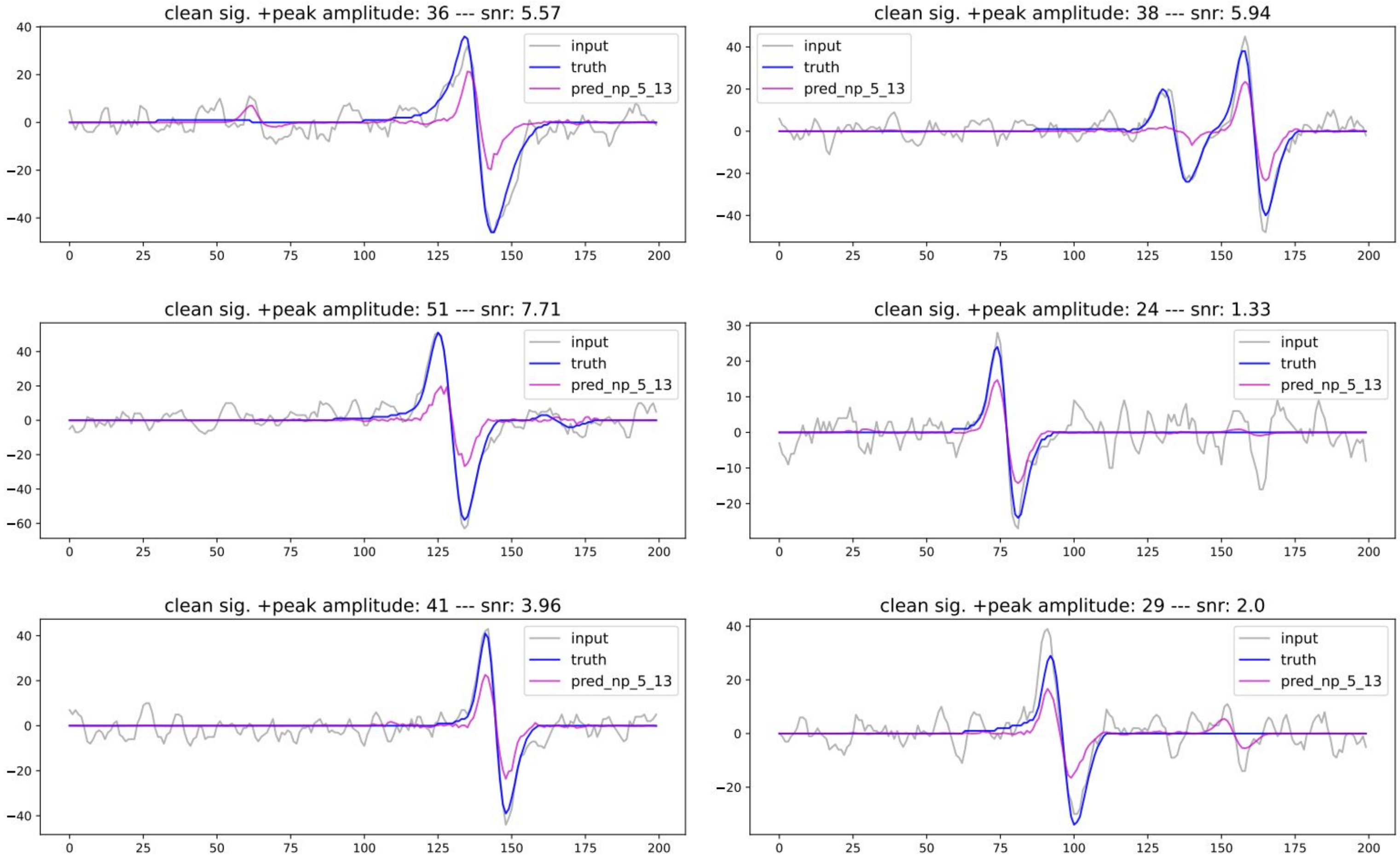
Model 1



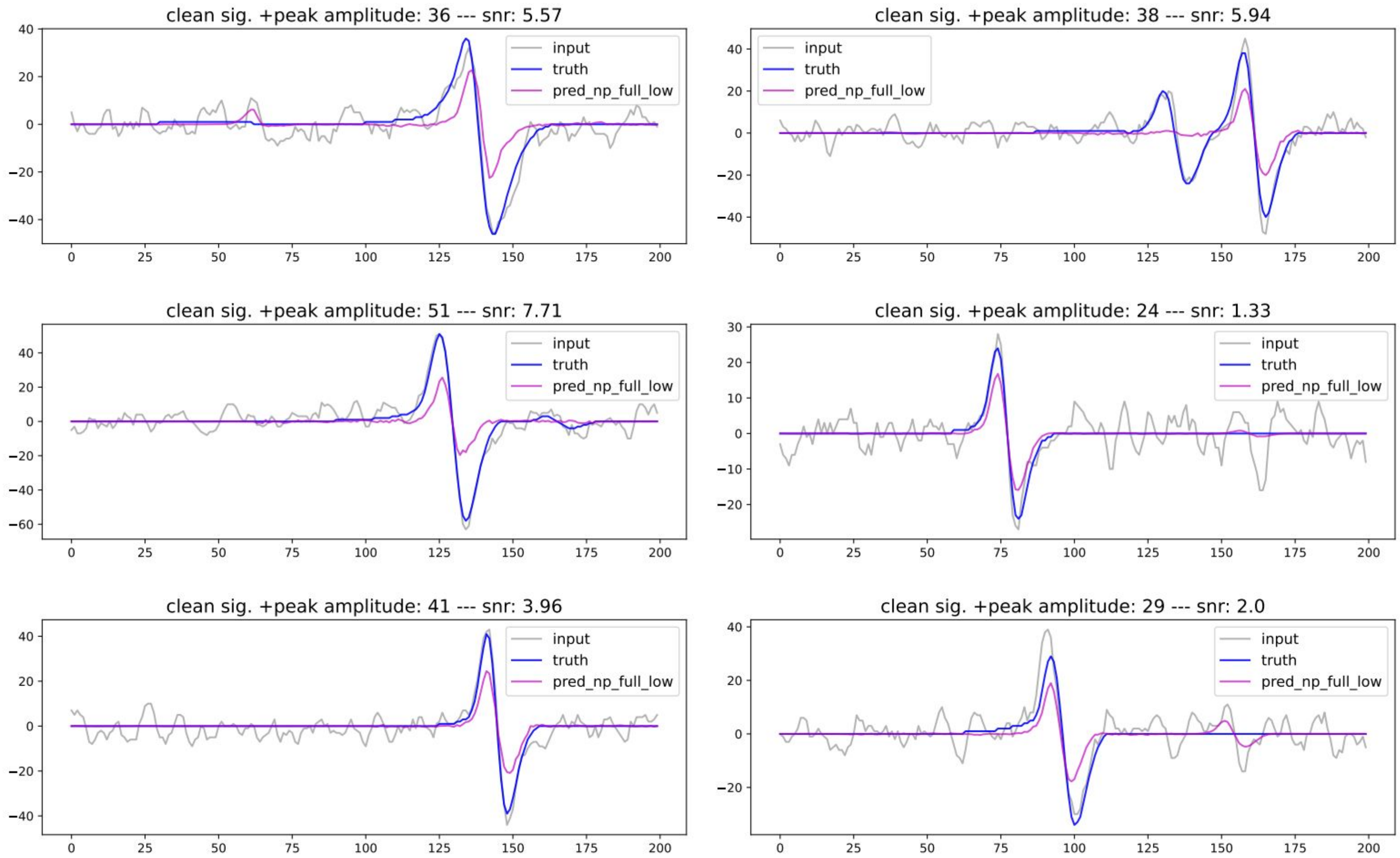
Model 2 trained with nu_cc/nu_es



Model 2 trained with Ar39 5_10



Model 2 trained with Ar39 5_13



Model 2 trained with Ar39 + nu_cc/nu_es

Notes

Latent space analysis

- Look into how the clusters are being formed

Training samples

- Train on Ar39 and nu_cc/nu_es separately and compare
- Balance dataset more (for higher ADC)

More interesting to explore lower adc count samples now

- Want models that is robust in different SNRs
- Apply to real noise data instead of simulation

MC generation

- NU_ES/NU_CC
  - Marley Generator
- Noise
  - Wirecell, dune far detector
- Ar39
  - Decay Zero