SBND TPC Electronics Pulser Calibrations

Lynn Tung

WireCell Meeting
September 5, 2024

Overview

- Electronics response pulser calibrations procedure
  - Calibration Results
  - Calibration Validation
- Using the updated model
  - RC Filter tuning
  - debugging

Electronics Response Calibration Procedure

1. Obtain pulser data for charge-injection values within the linear region (DAC 1-10)

2. Within a single charge-injection value (one run):

   A. Perform peak finding for many waveforms, results in 60-70 pulses

   B. Obtain the "average pulse response" to remove noise/cosmic signals

      - median waveform is used, less sensitive to cosmic ionization signals
      - the error of each data point = quartile deviation (less sensitive to outliers)

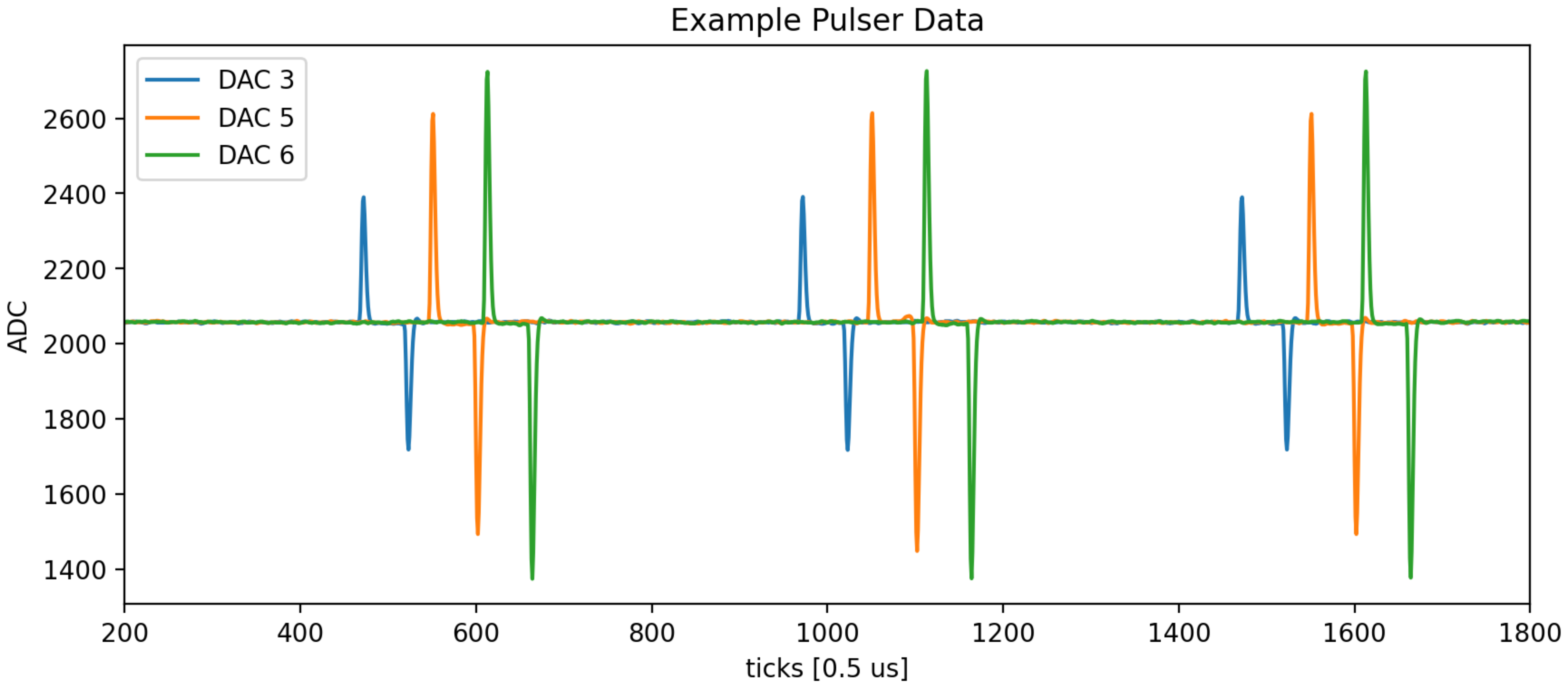
   C. Obtain the amplitude and shaping time from fitting the electronics response function

3. Extract the **gain** from the DAC vs. fitted amplitude

4. Extract the **shaping time** as a weighted average from all DAC values

Electronics Response Calibration Procedure

1. Obtain pulser data for charge-injection values within the linear region (DAC 1-10)



Electronics Response Calibration Procedure

1. Obtain pulser data for charge-injection values within the linear region (DAC 1-10)

2. Within a single charge-injection value (one run):

   A. Perform peak finding for many waveforms, results in 60-70 pulses

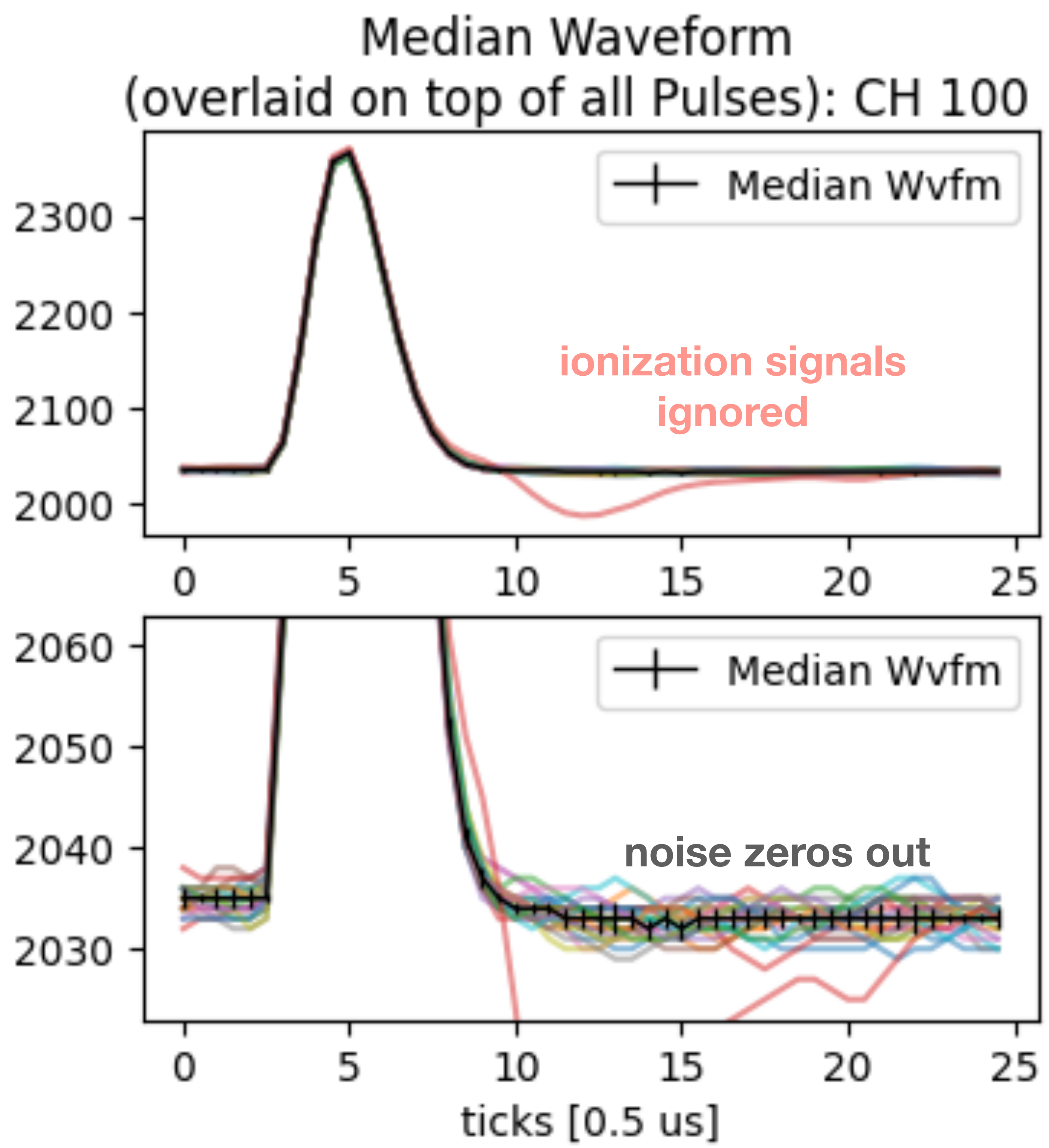
   B. Obtain the "average pulse response" to remove noise/cosmic signals

      - median waveform is used, less sensitive to cosmic ionization signals
      - the error of each data point = quartile deviation

   C. Obtain the amplitude and shaping time from fitting the electronics response function

3. Extract the **gain** from the DAC vs. fitted amplitude

4. Extract the **shaping time** as a weighted average from all DAC values



Electronics Response Calibration Procedure

1. Obtain pulser data for charge-injection values within the linear region (DAC 1-10)

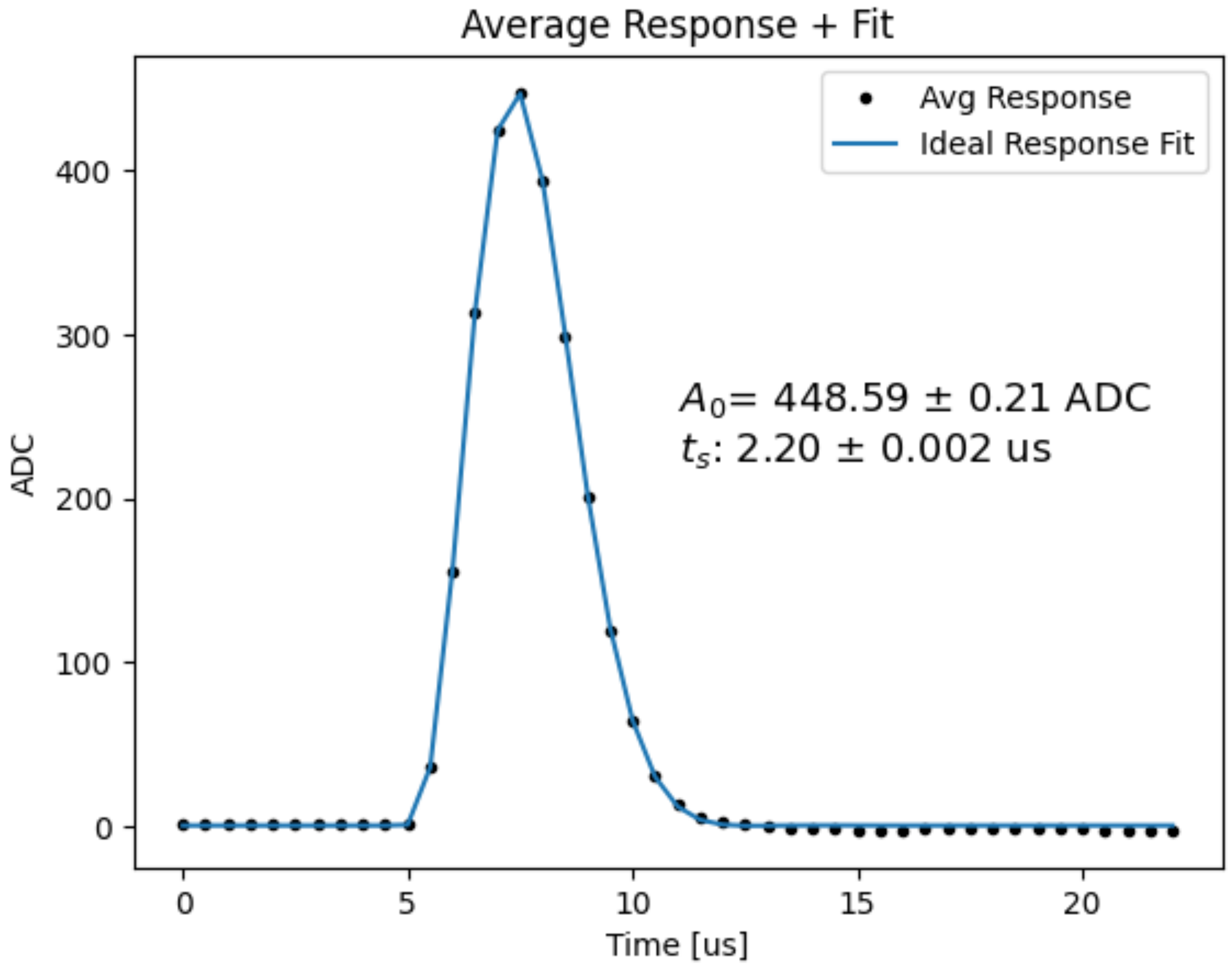
2. Within a single charge-injection value (one run):

   A. Perform peak finding for many waveforms, results in 60-70 pu

   B. Obtain the "average pulse response" to remove noise/cosmic

   - median waveform is used, less sensitive to cosmic ionization signals
   - the error of each data point = quartile deviation

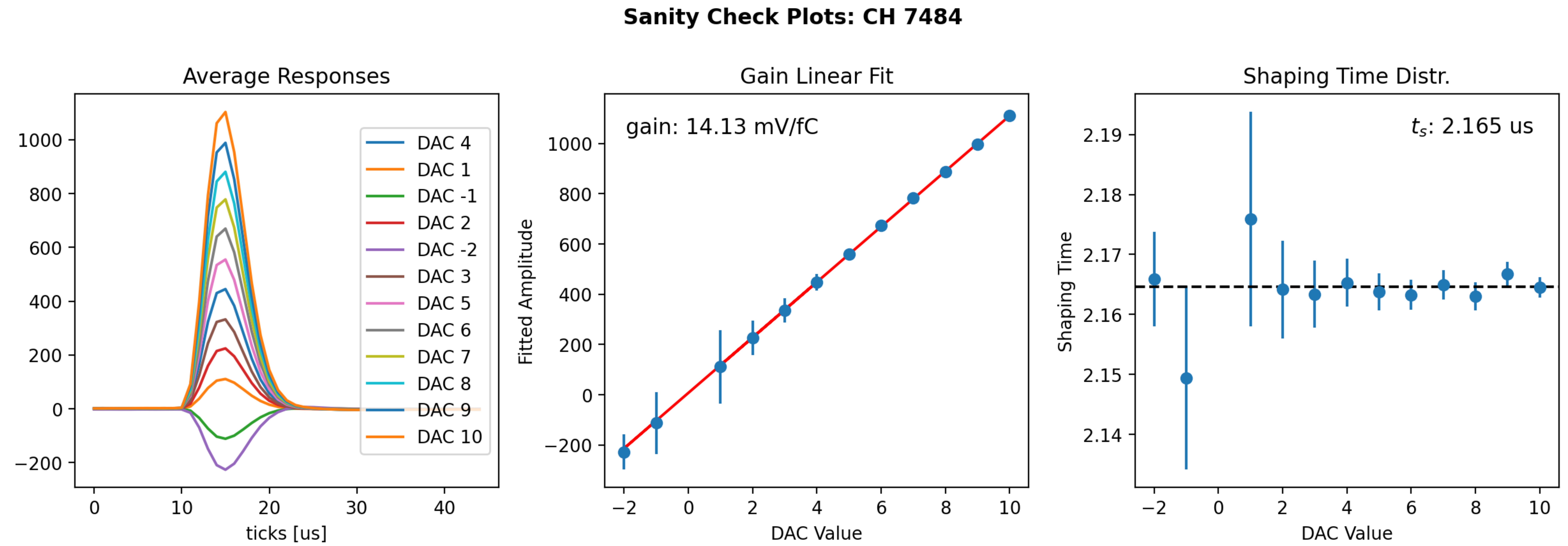
   C. Obtain the amplitude and shaping time from fitting the electronics response function



3. Extract the **gain** from the DAC vs. fitted amplitude

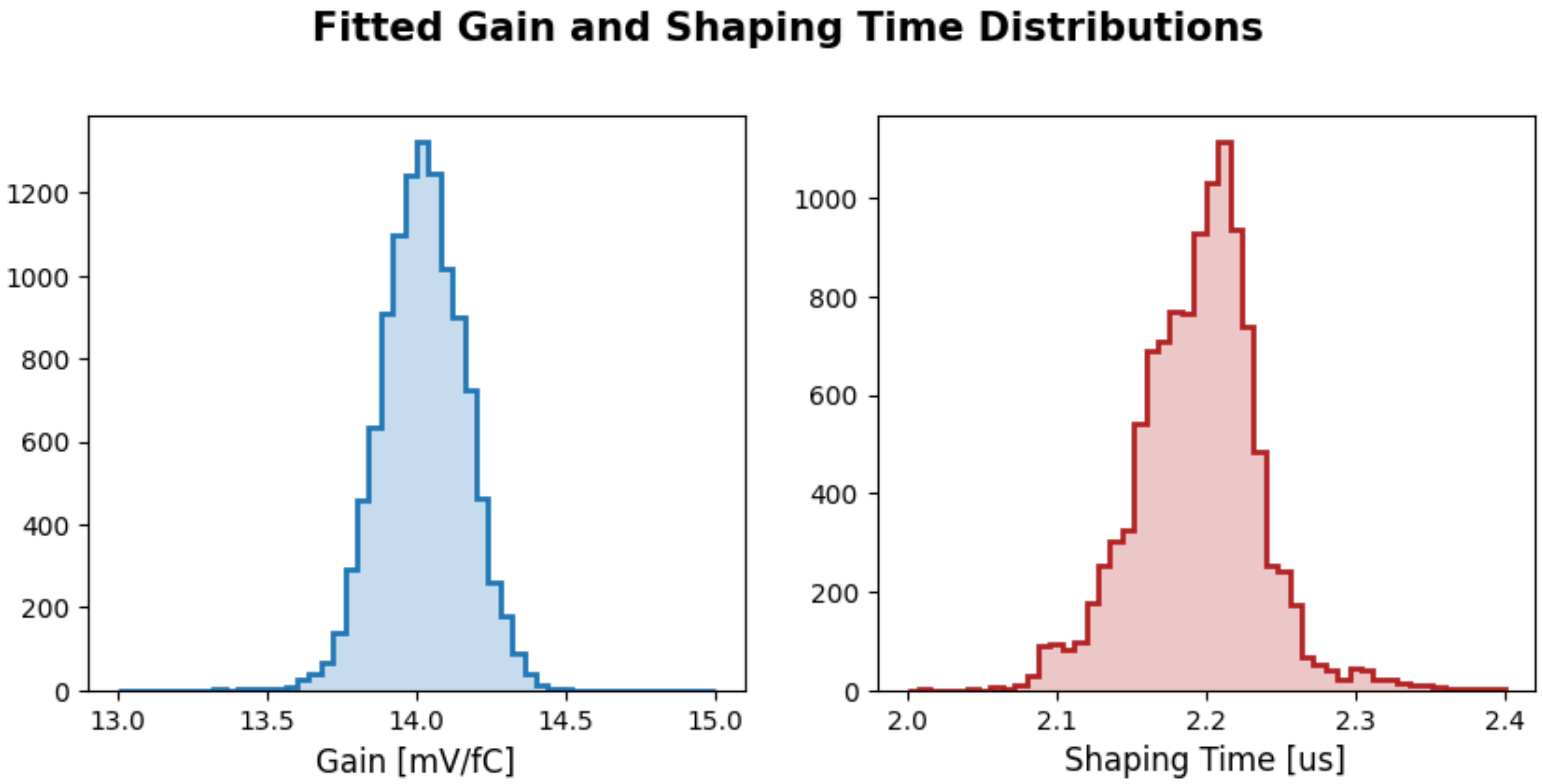
4. Extract the **shaping time** as a weighted average from all DAC values

Electronics Response Calibration Procedure



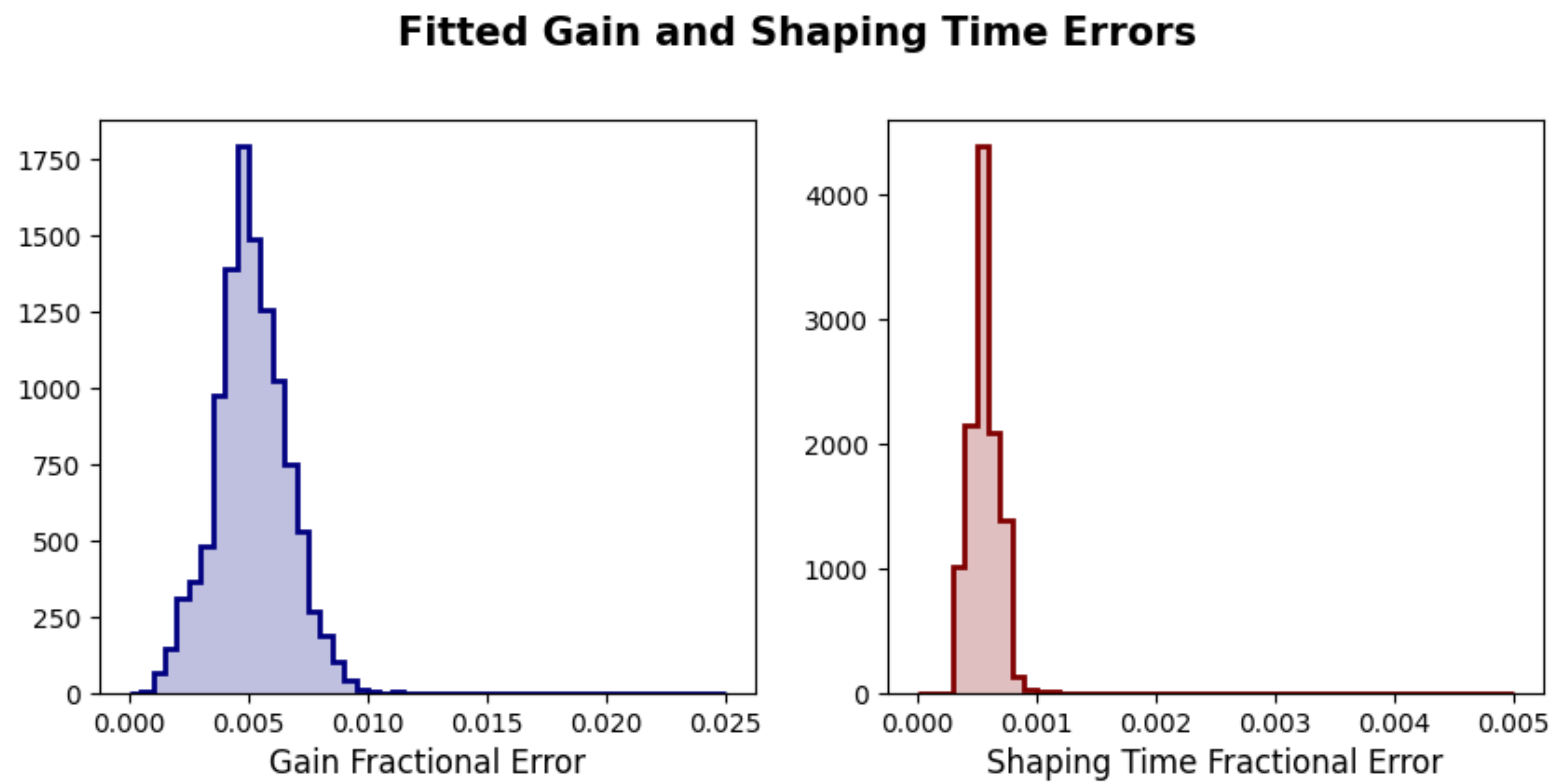
3. Extract the **gain** from the DAC vs. fitted amplitude
4. Extract the **shaping time** as a weighted average from all DAC values

Calibration Values



- obtain gain and shaping time for all channels!

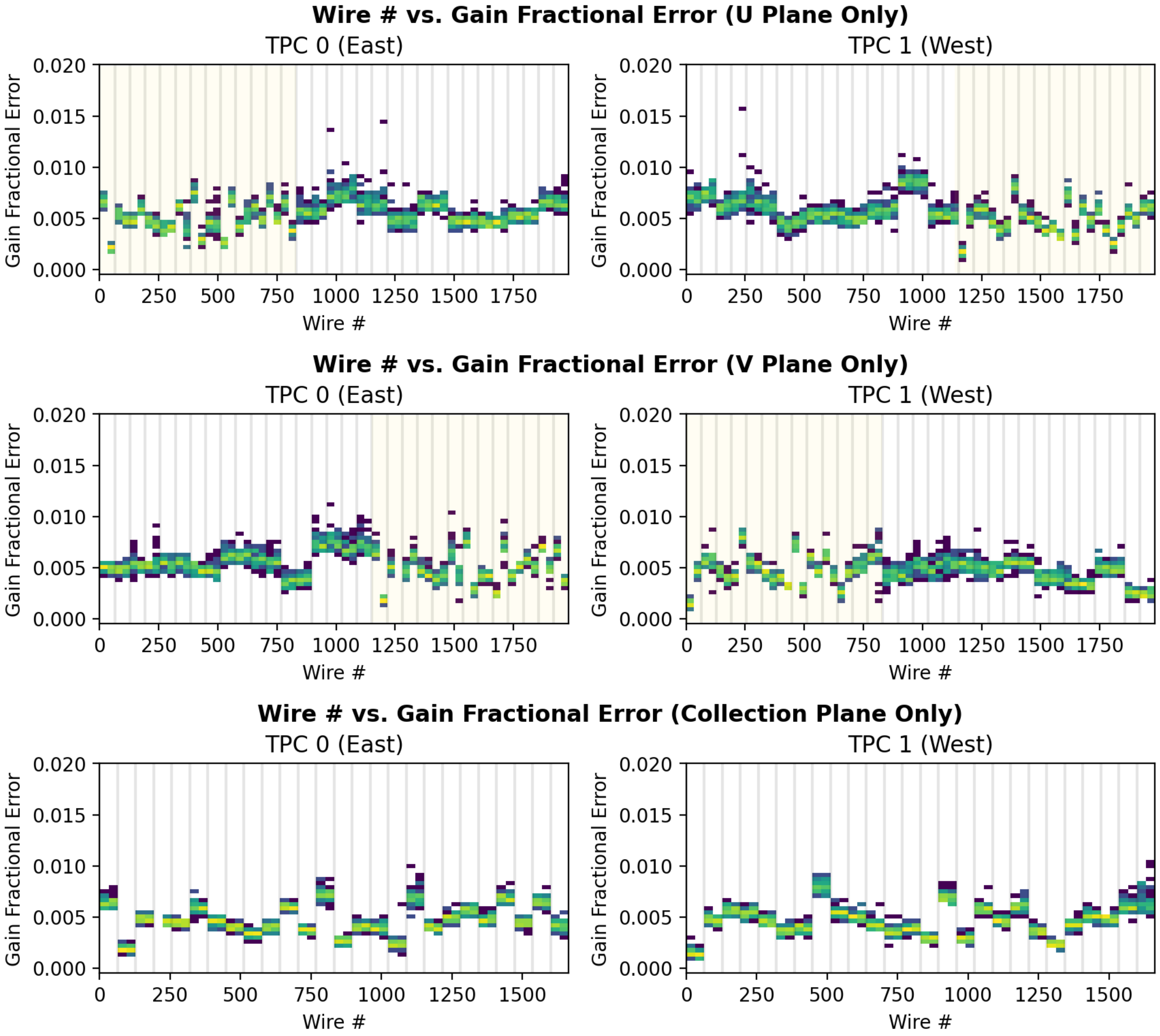
Calibration Value Errors



- distributions of the fractional error for both calibration parameters
  - error on the gain obtained directly from the linear fit, error on the shaping time is the error on the weighted average

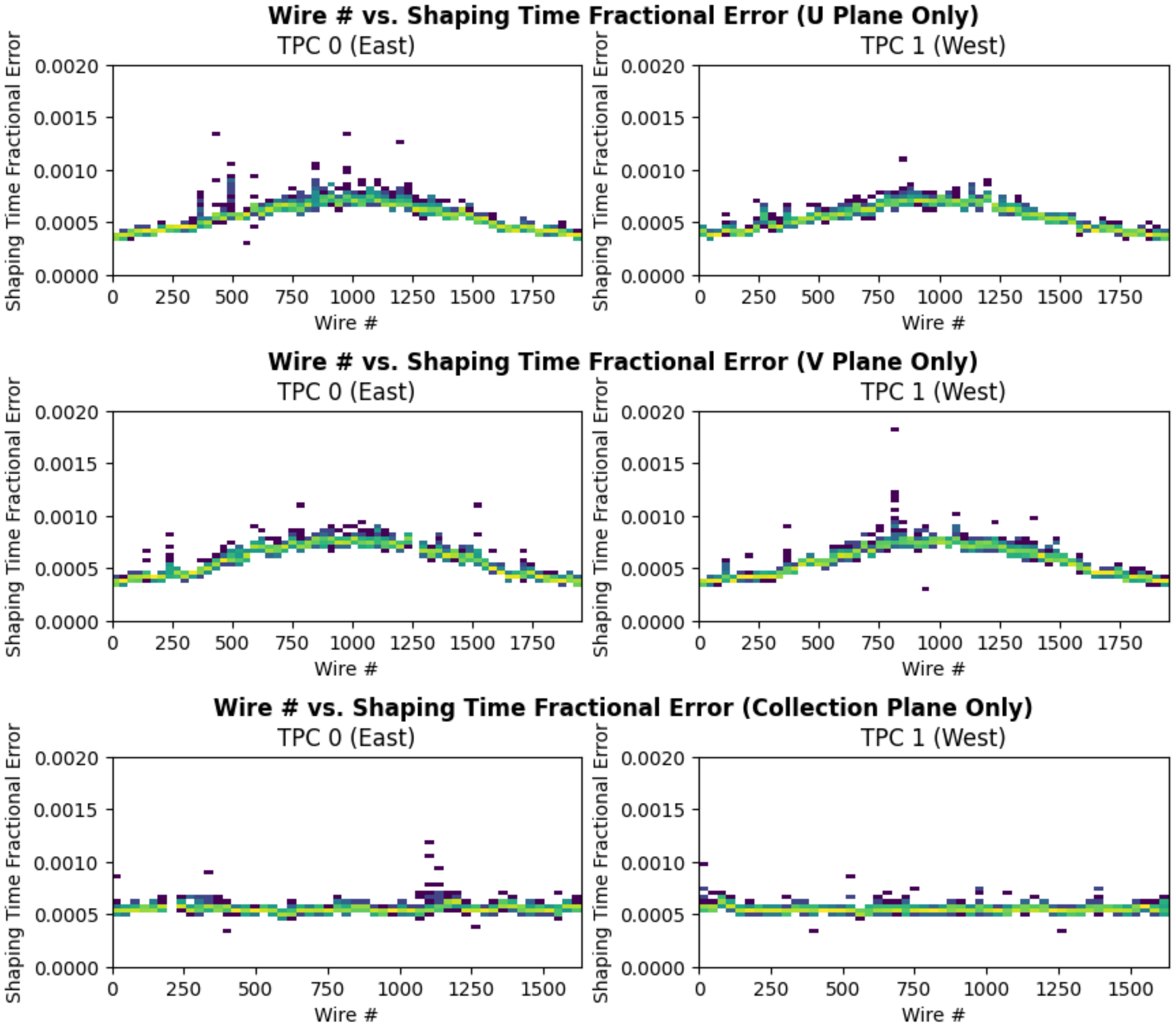
Fit Performance vs. Wire

- **gain fractional error**
- vertical lines separate groups of 64 channels
- induction planes with CE installed at the TPC sides seem to have 32 wire grouping vs those installed on the TPC top, which seem to have 128 wire grouping
- collection planes have grouping by 64 channels

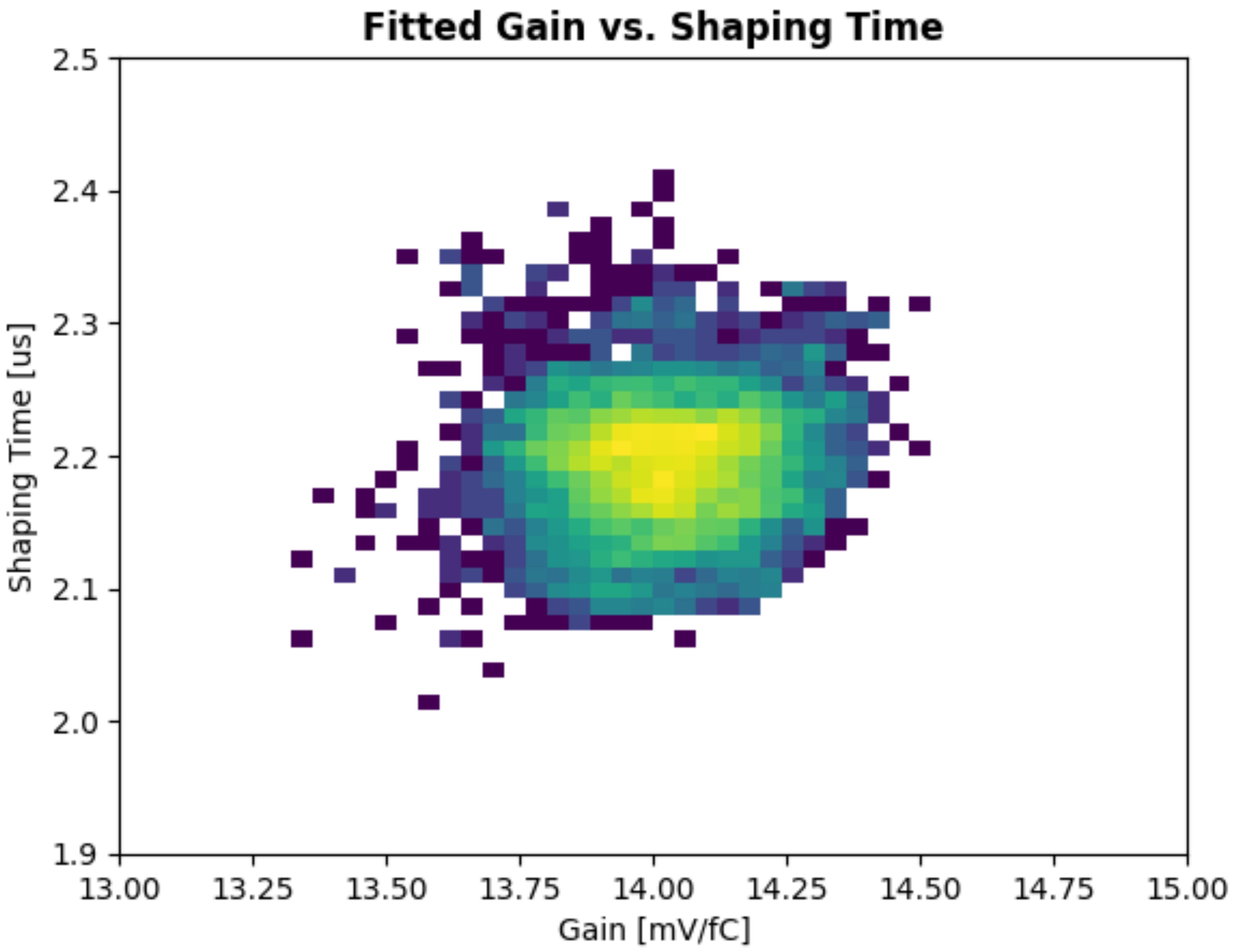


Fit Performance vs. Wire

- **shaping time fractional error**
- no clear wire grouping, but clear wire length dependency!



Sanity Check!

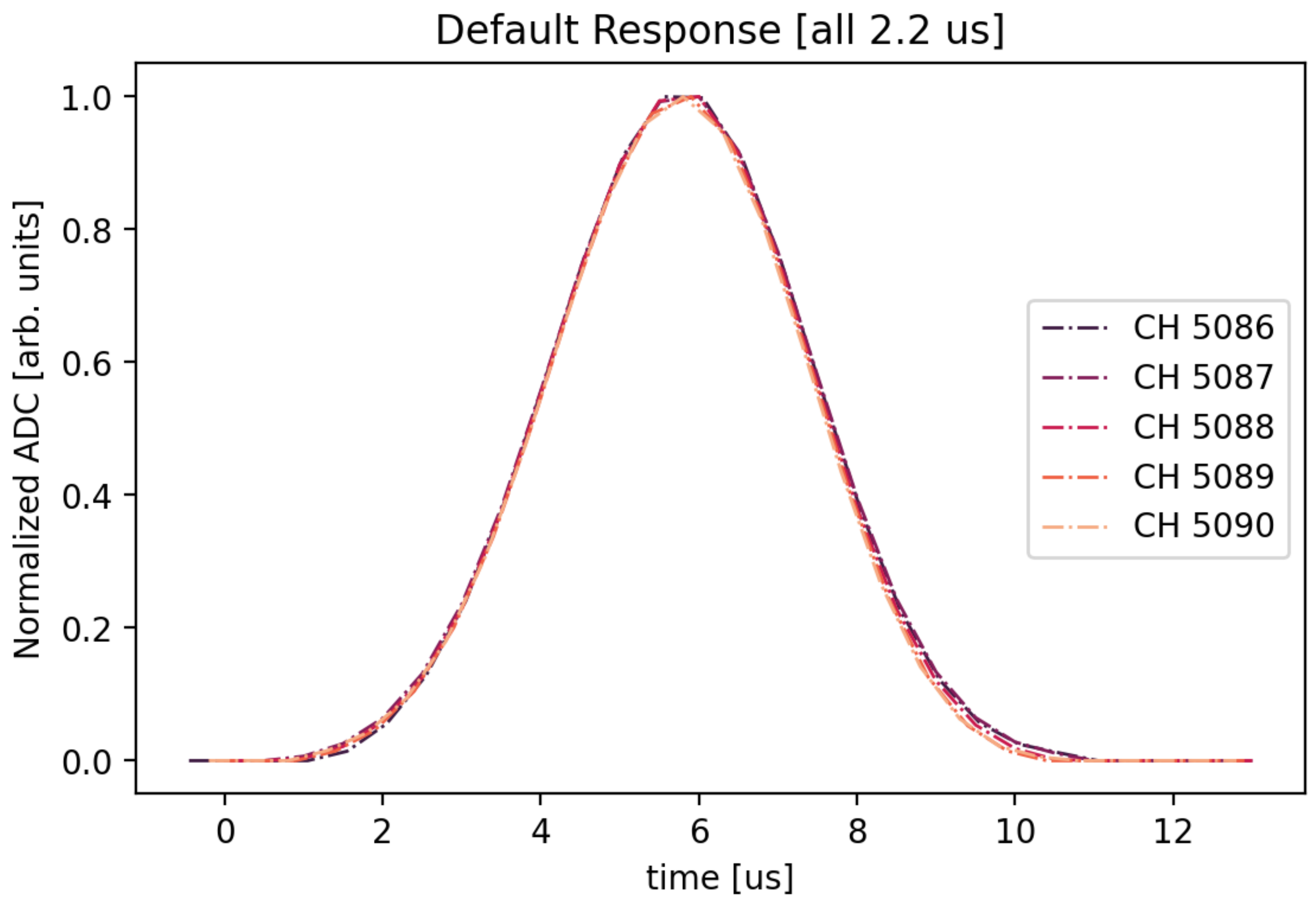


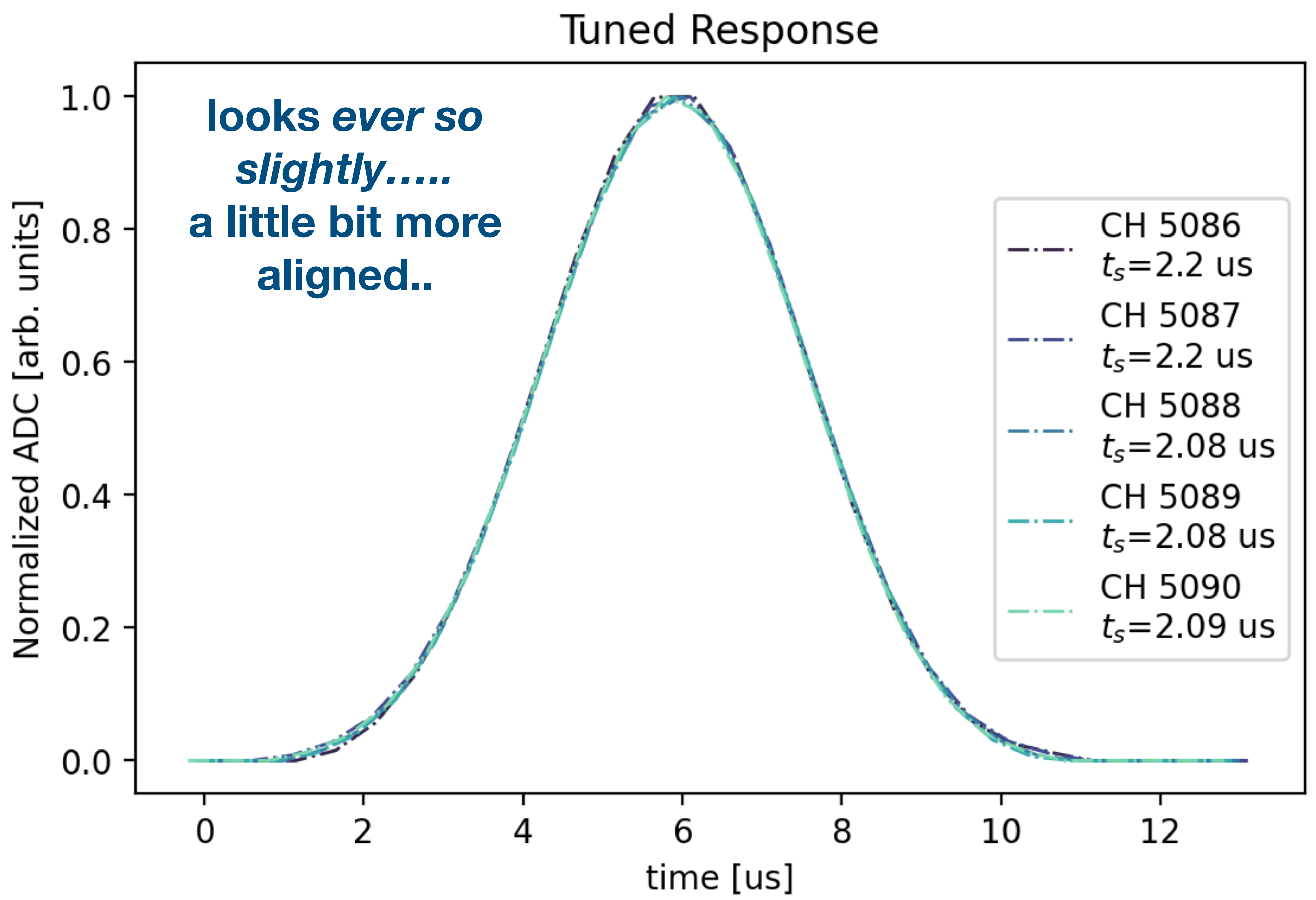
- we don't expect the gain and shaping time to be clearly correlated, so this looks good

Validation with Calibrated Values

- validation with data is somewhat challenging; the improvement that we expect to see is *at most* on the few percent level, more likely even less after the inherent smearing of signal processing

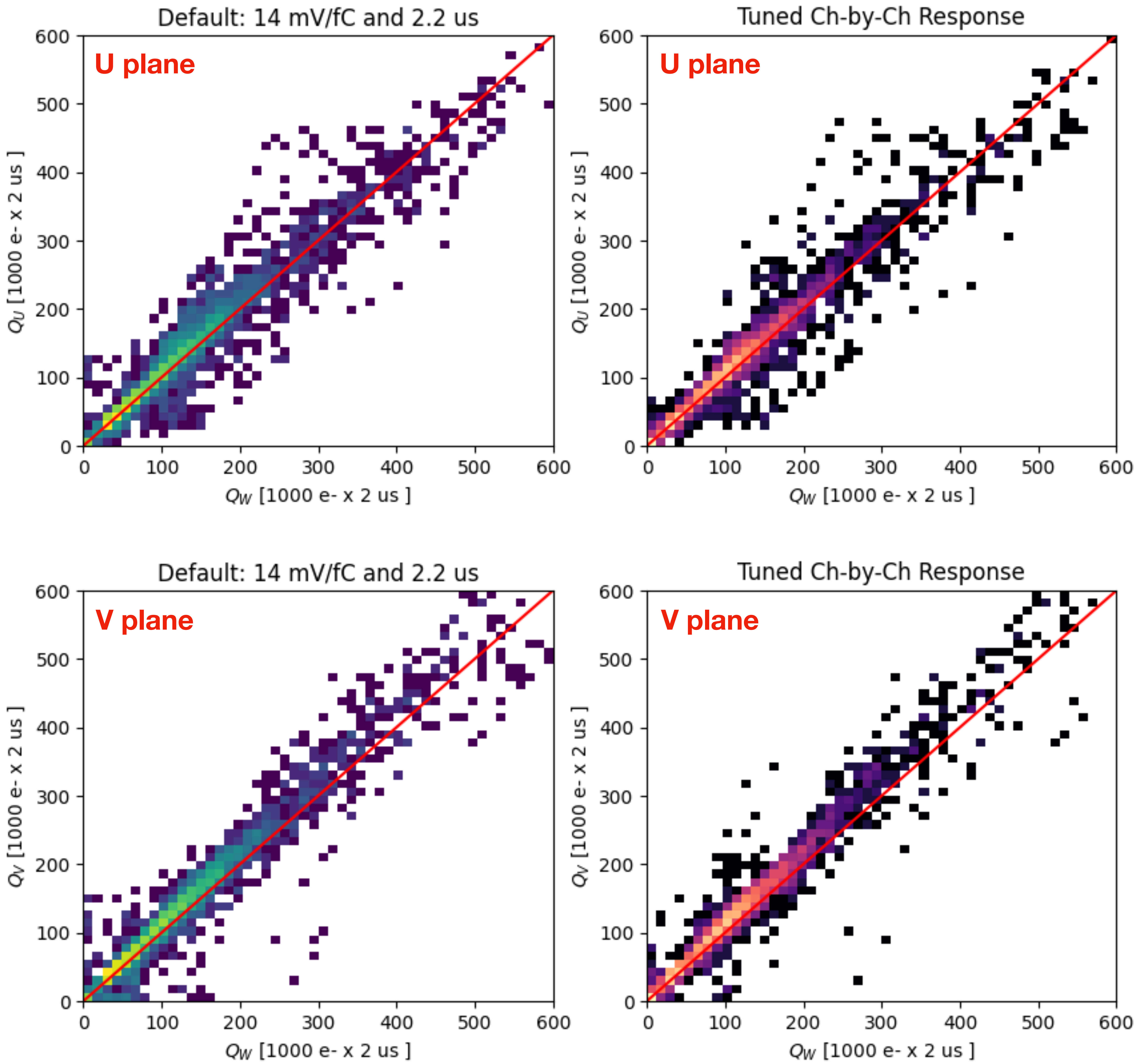
    - validation with full reconstruction, accurate dx correction, and larger statistics may help!





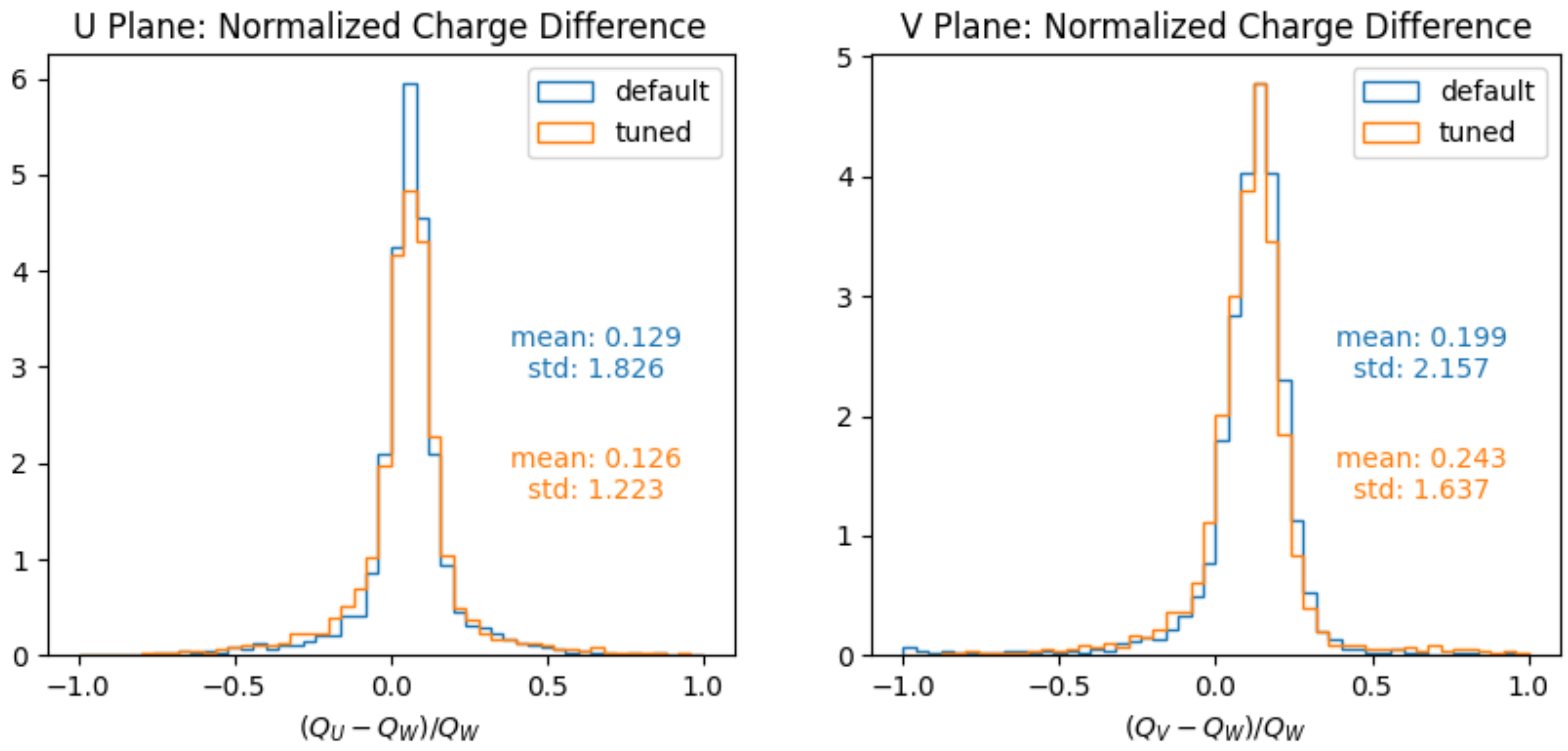
Calibration Validation

- some very basic validation using waveforms only:
  - matching the integrated deconvolved charge over time slices (4 ticks = 2 us), we can compare the extracted charge from the induction planes to the collection plane



- **outcome**: both the default and the tuned comparisons show good matching (with some bias)
  - using the tuned response leads to better charge resolution!

Calibration Validation

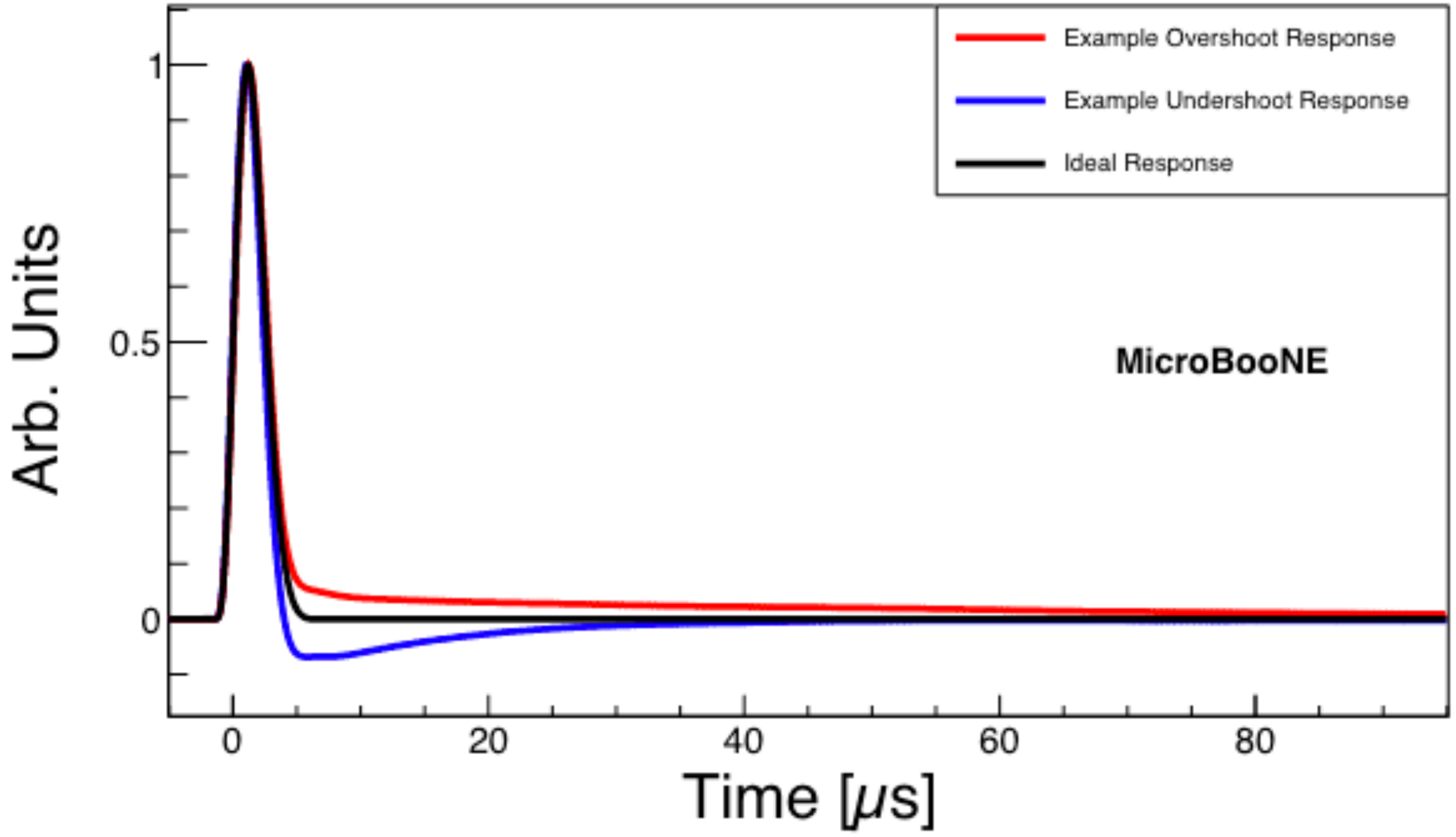


- the charge agreement between the induction planes and collection planes has overall improved using the tuned electronics response
  - caveat: the mean and standard deviation are not great metrics due to the outliers in the distribution, but just using them as a first look!

Updated Model
(Imperfect Pole-Zero Cancellation)

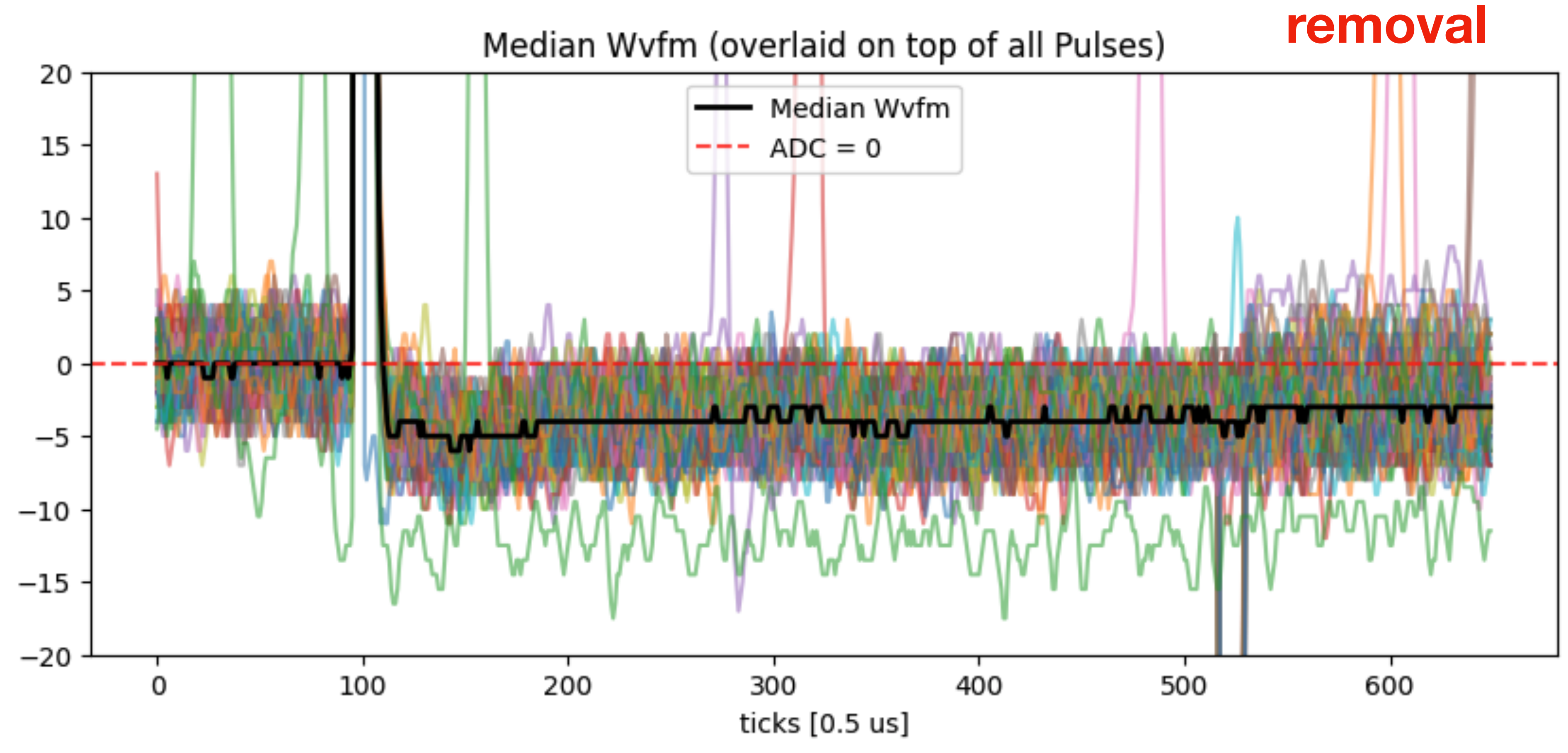
Imperfect Pole-Zero Cancellation

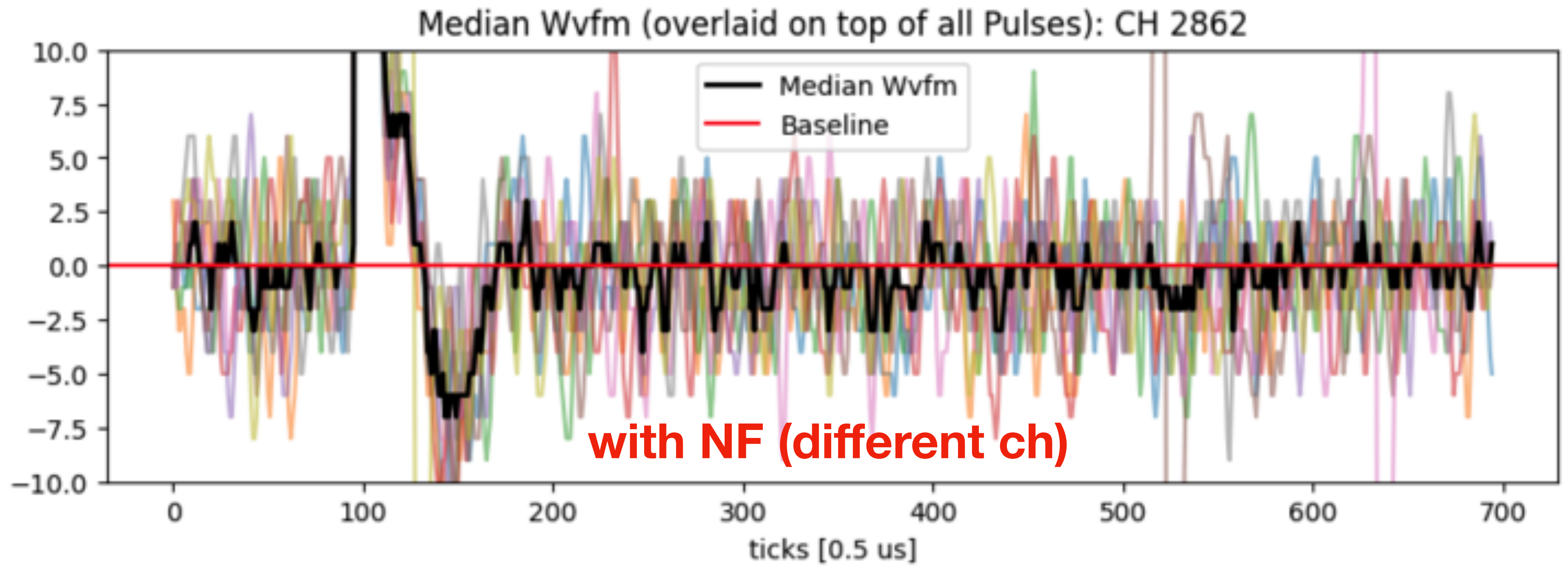
- although this model works well in many cases, the ideal electronics response does not account for some effects, such as *undershoot* and *overshoot* tails due to imperfect pole-zero cancellation

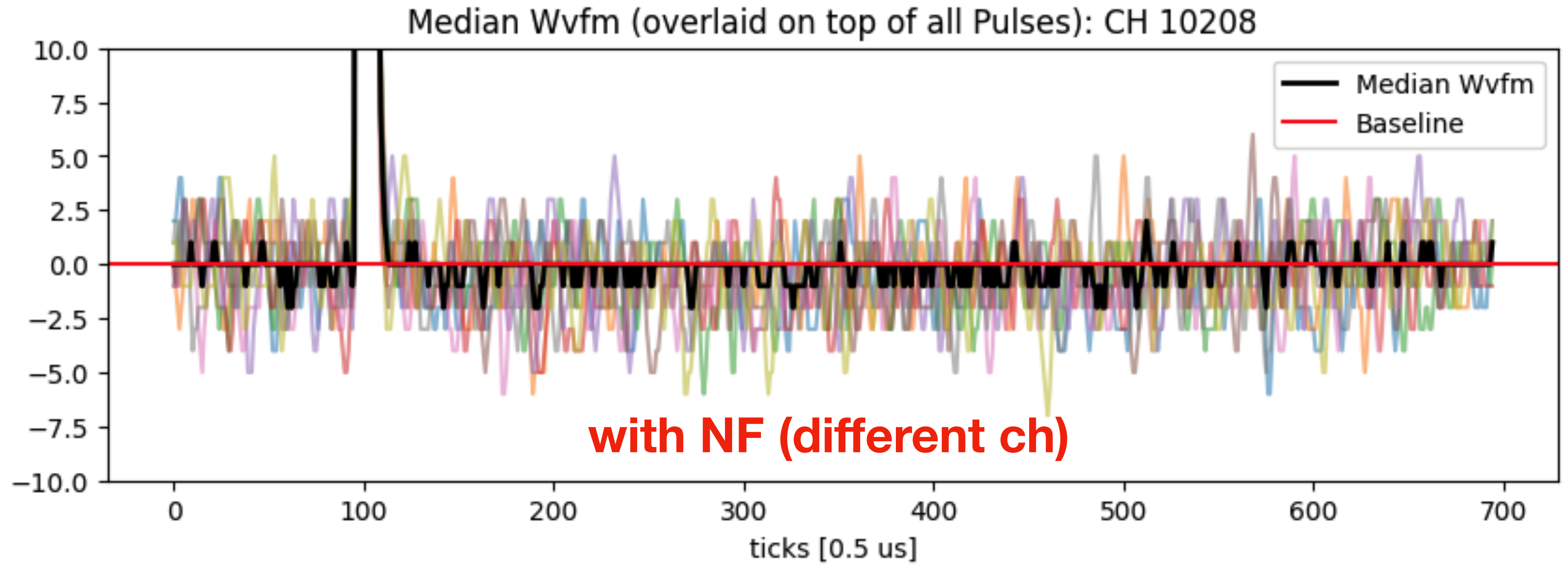


but first… RC effect

without RC tail removal





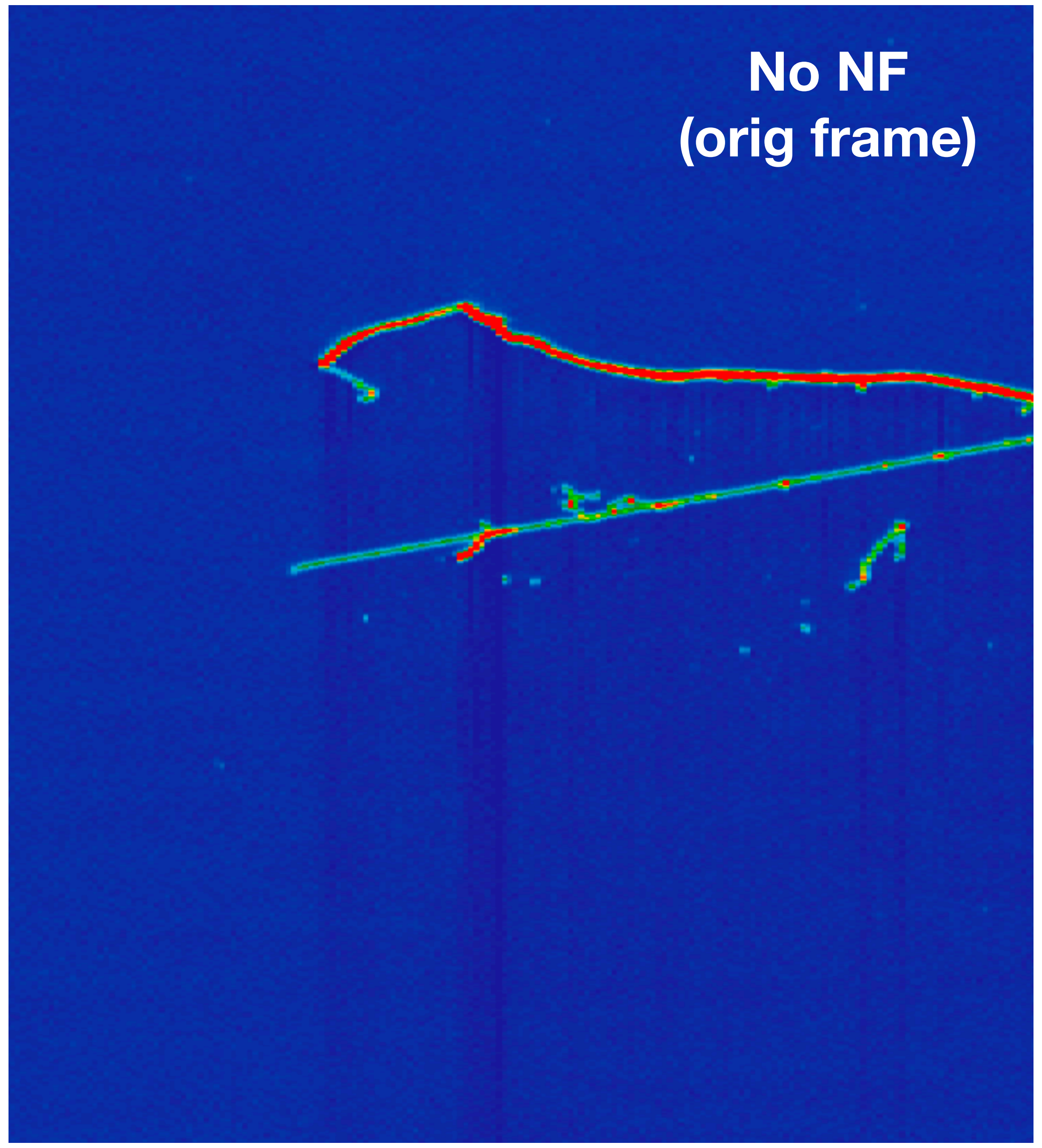


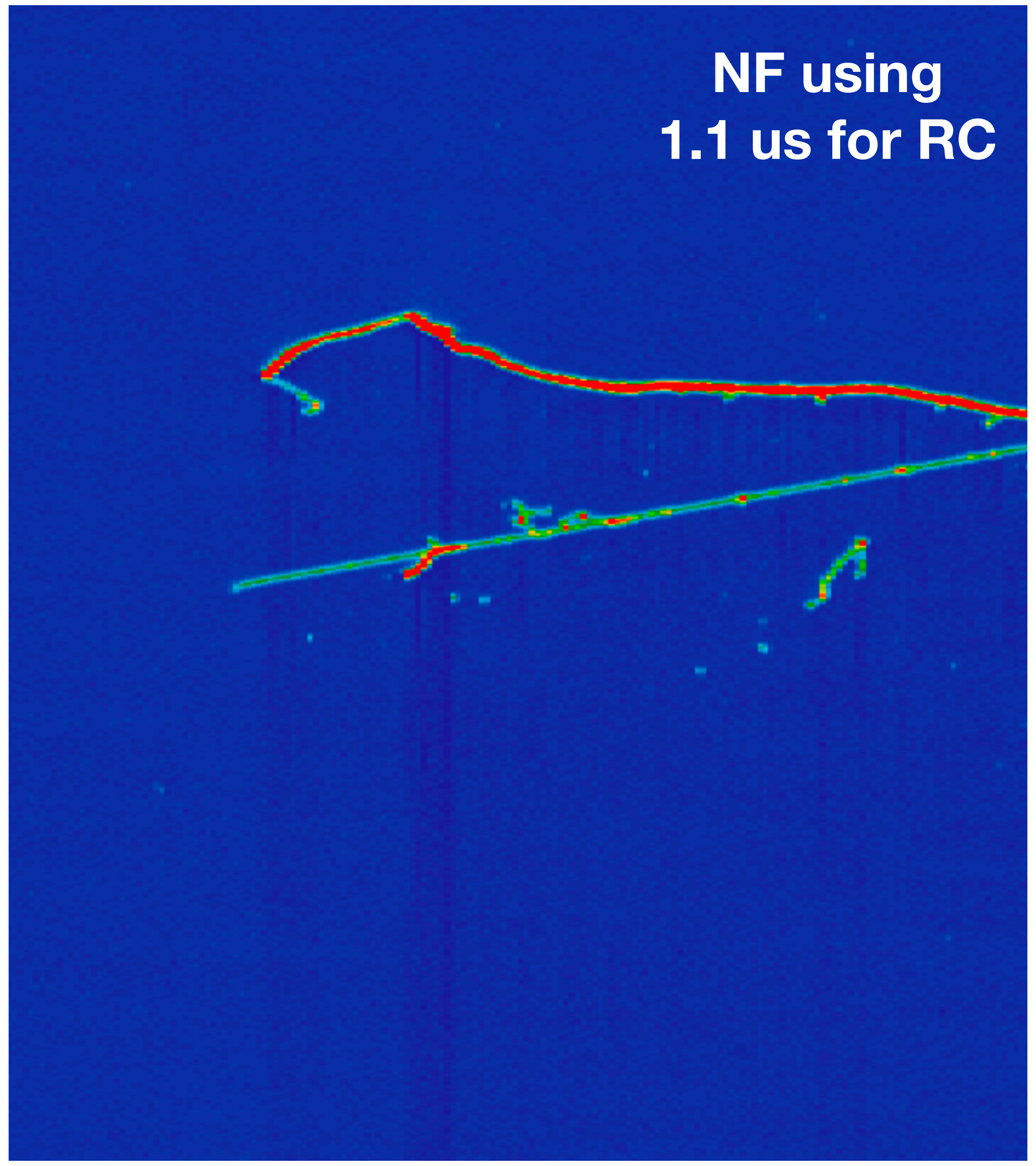
- when inspecting waveforms for small undershoots in the tail, noticed that nearly all had a small undershoot with very long time constant → RC tail!

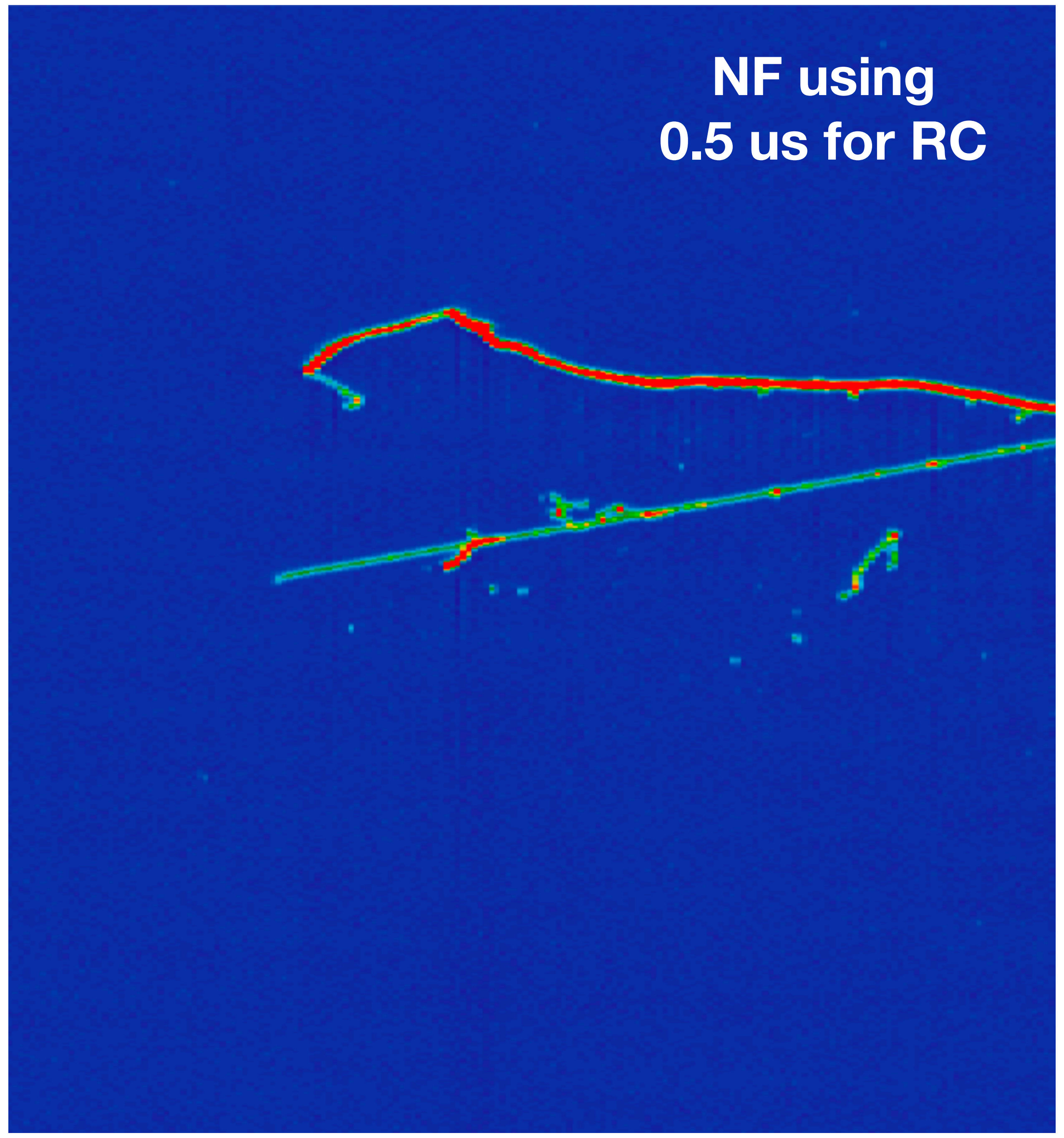
- need to remove the RC tail to properly characterize possible undershoot/ overshoot effects due to pole-zero cancellation

- using NF waveforms (with coherent noise removal turned off)

RC effect (cont.)







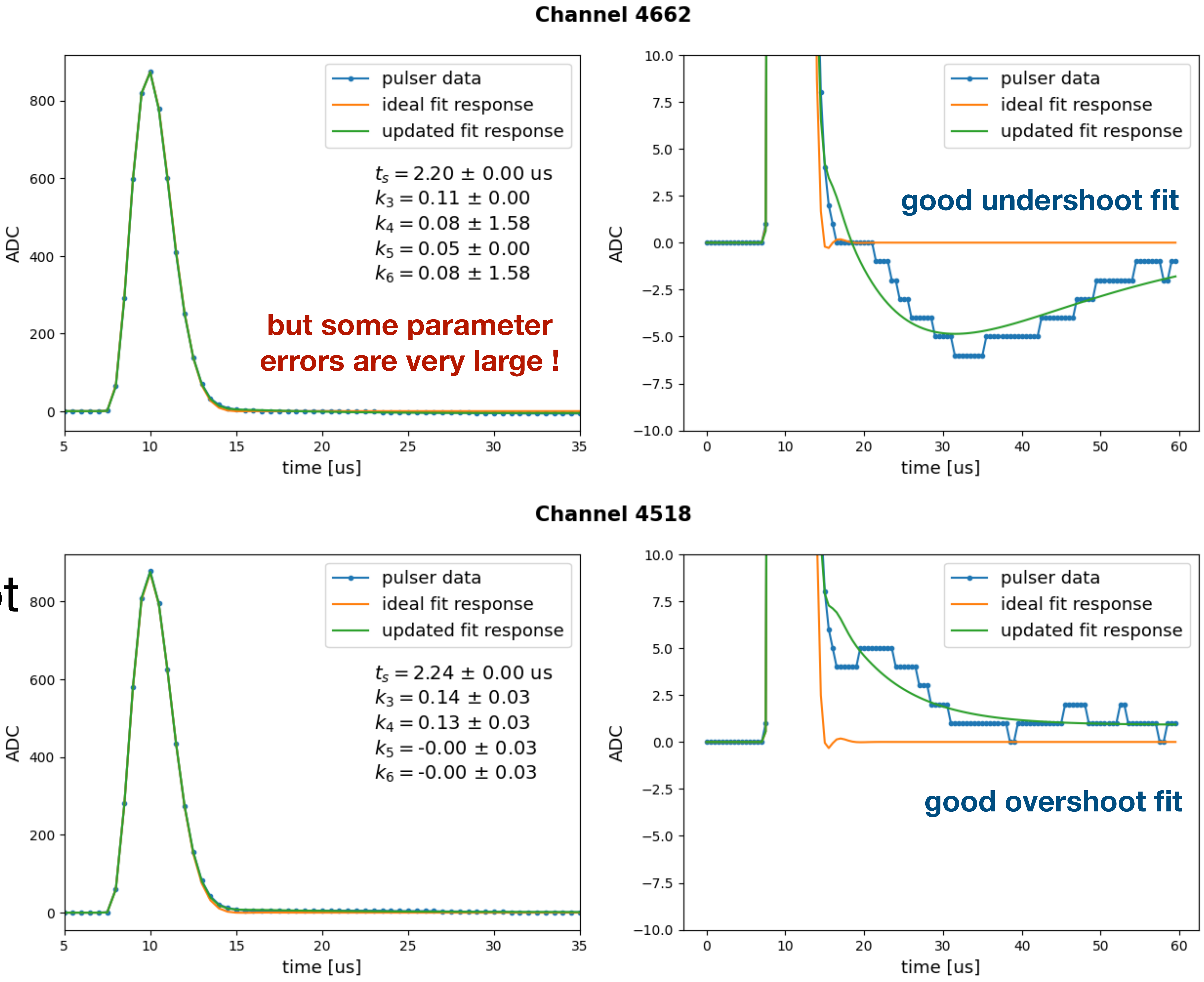
- the value of the RC time constant needed to remove the observed tail was very different from the expected
  - Shanshan reported **1.1 us** from his tests
  - after testing many values and comparing waveforms (backup) and event displays, **0.5 us** seemed to be most effective in removing the observed tail

Updated Response Model

- for shaping studies with updated model, using 100+ pulses with DAC 8

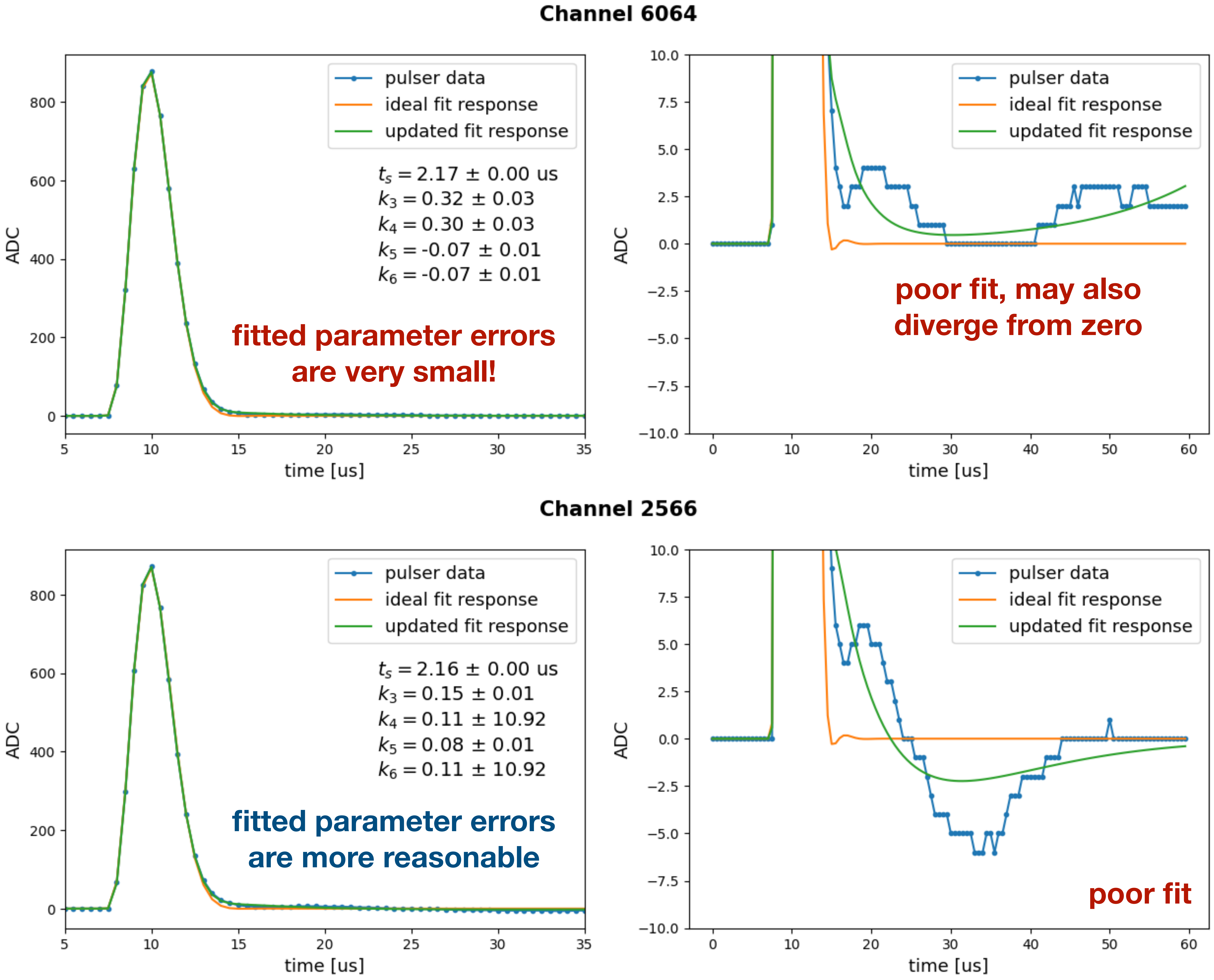
- the updated response model can account for some features we also see in our pulser data

  - the undershoot and overshoot effects in SBND pulser data are *very small* compared to the main response



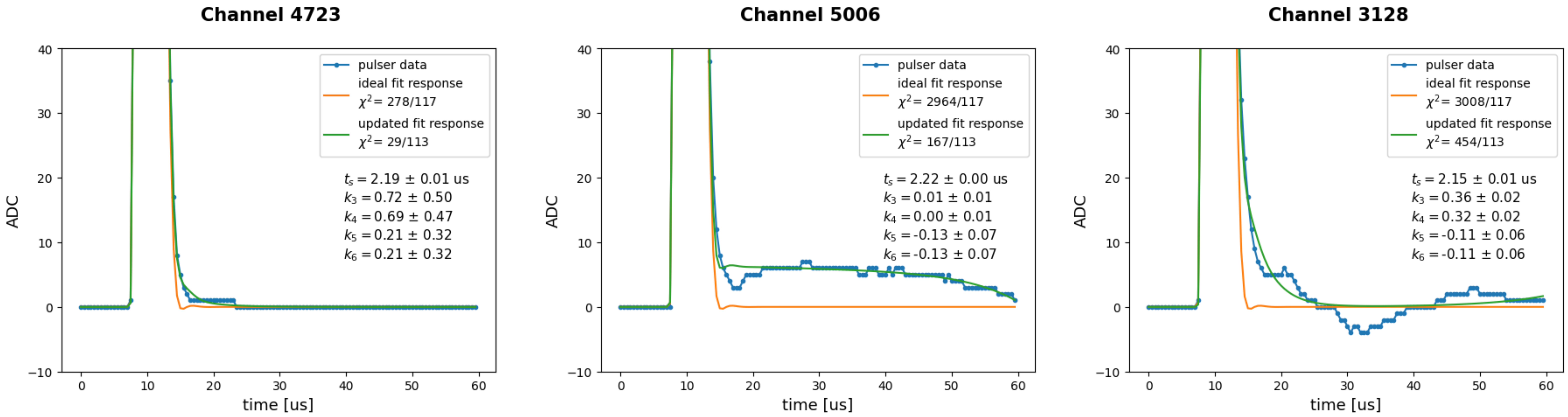
Updated Response Model

- will see features such as multiple over/undershoots, or features with very long time-scales
  - imperfect pole-zero cancellation model doesn't model these well
- the errors in the fitted parameters do not necessarily reflect the goodness-of-fit



Updated Response Model

- still in the process of determining a metric and procedure for deciding when to use ideal response and when to use the updated model for each channel
  - uncertainties in updated model fit do not always reflect goodness-of-fit



Updated Response Model: debugging

- when first attempting to use ParamsPerChannelResp correction with fitted pole values, the SP stage starts to output *empty frames*

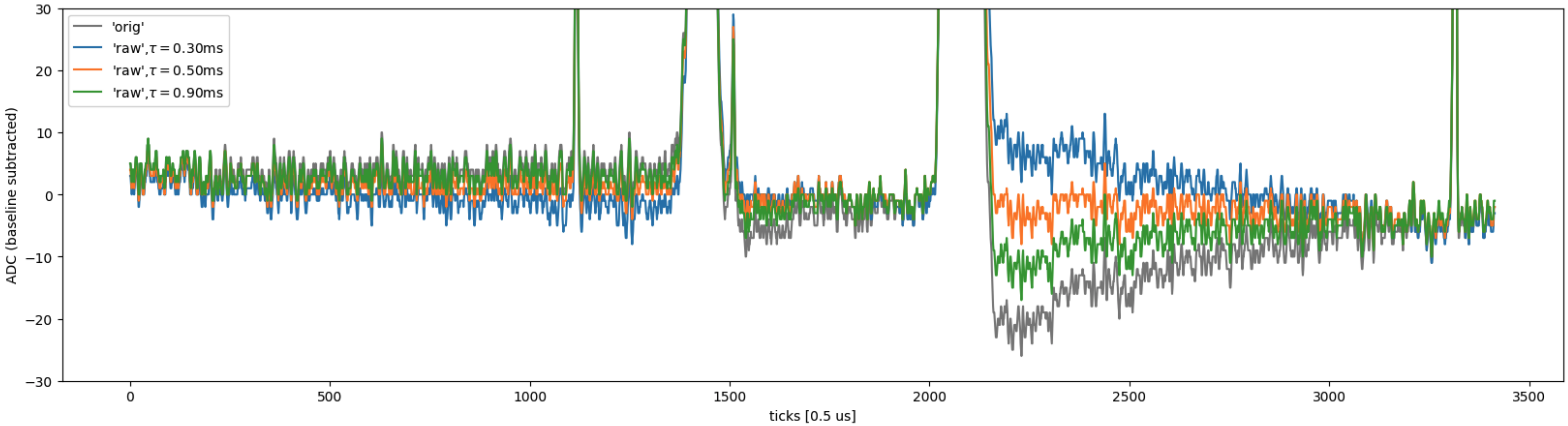
```
[20:47:21.226] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 init nticks=3415 tbinmin=0 tbinmax=3415
[20:47:21.266] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 load plane index: 0, ntraces=5632, input bad regions: 1
[20:47:21.450] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 applying ch-by-ch electronics response correction
[20:47:24.892] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 load plane index: 1, ntraces=5632, input bad regions: 1
[20:47:25.047] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 applying ch-by-ch electronics response correction
[20:47:28.688] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 load plane index: 2, ntraces=5632, input bad regions: 0
[20:47:28.899] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 applying ch-by-ch electronics response correction
[20:47:30.343] D [sigproc ] <OmnibusSigProc:apa1sigproc1> 1664 empty rows out of size=(1664,3415)
[20:47:30.930] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 save plane index: 0, Qtot=56797706 Qloss=-6266144, 4613
indices spanning [0,4612] "wiener"
[20:47:31.315] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 save plane index: 0, Qtot=50996746 Qloss=-1695435, 2494
indices spanning [4613,7106] "gauss"
[20:47:31.775] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 save plane index: 1, Qtot=63567394 Qloss=-7627208, 3926
indices spanning [7107,11032] "wiener"
[20:47:32.107] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 save plane index: 1, Qtot=54804406 Qloss=-1516344, 2110
indices spanning [11033,13142] "gauss"
[20:47:32.343] D [sigproc ] <OmnibusSigProc:apa1sigproc1> 1664 empty rows out of size=(1664,3415)
[20:47:32.404] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 wiener save plane index: 2 empty
[20:47:32.656] D [sigproc ] <OmnibusSigProc:apa1sigproc1> 1664 empty rows out of size=(1664,3415)
[20:47:32.728] D [sigproc ] <OmnibusSigProc:apa1sigproc1> call=0 gauss save plane index: 2 empty
```

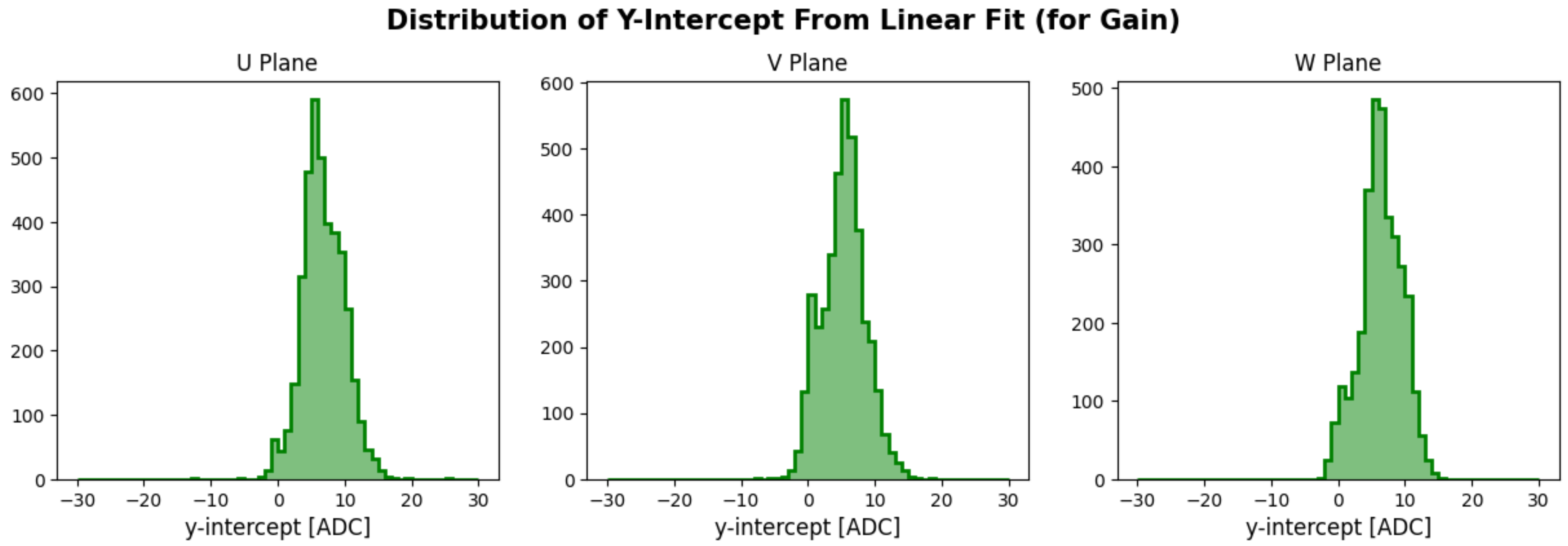
- SP runs normally if all pole values are their default (using ideal response), where k3=k4=0.1, and k5=k6=0.0

Summary

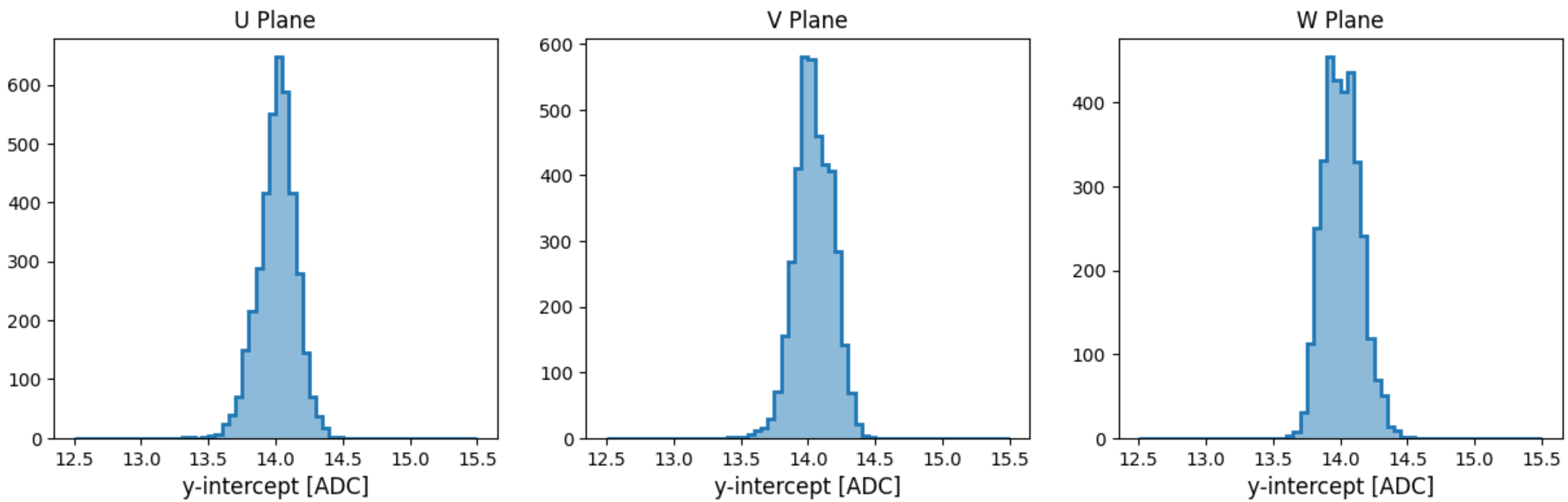
- First pass of the electronics response calibration has been performed using 100 kV pulser data!
  - work-in-progress: using an updated model to better characterize some very minor shape differences
- Very early validation shows improvement in the extracted charge agreement between the induction and collection panes

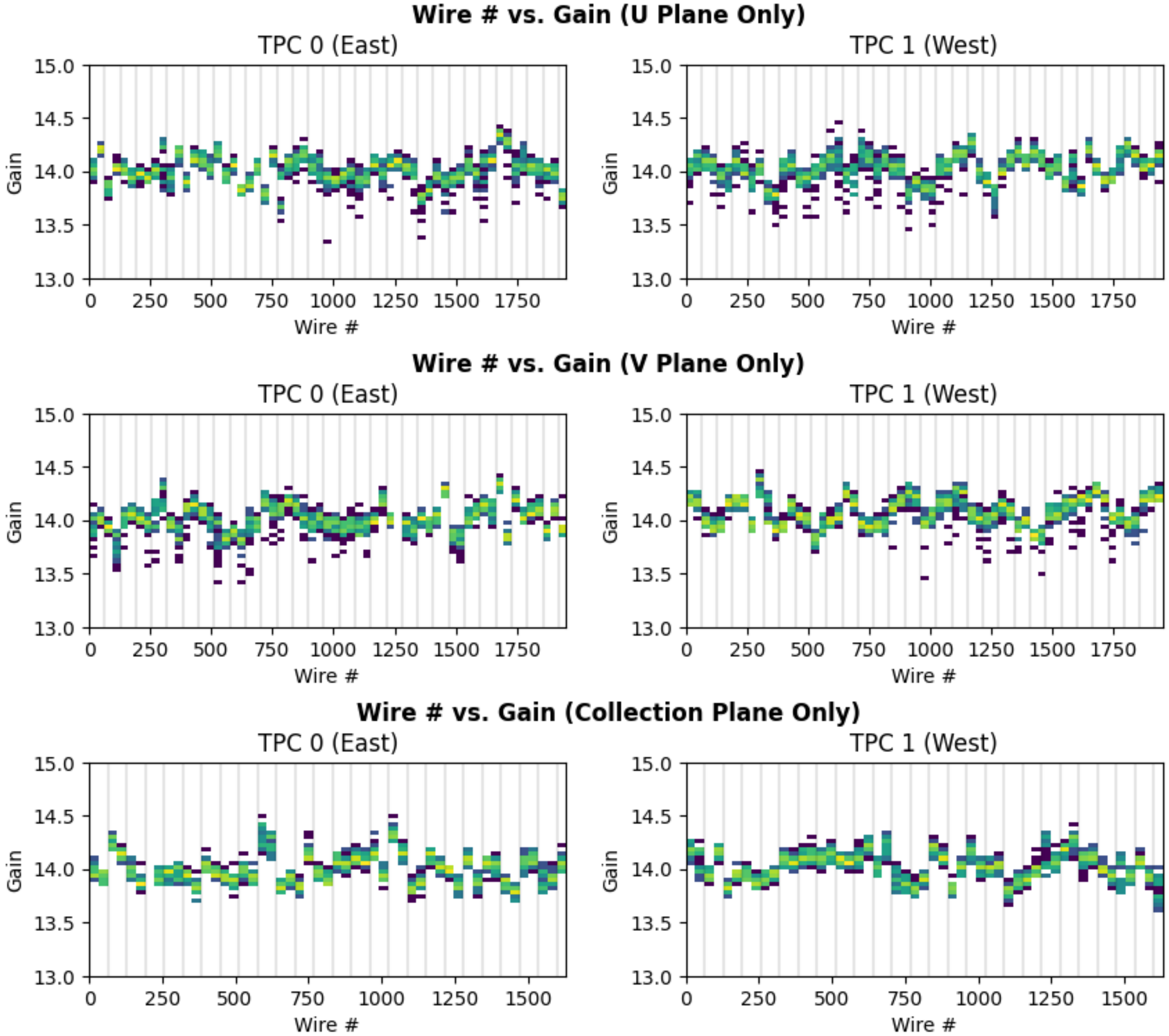
backup


'orig'
'raw',τ = 0.30ms
'raw',τ = 0.50ms
'raw',τ = 0.90ms
ADC (baseline subtracted)
30
20
10
0
−10
−20
−30
0
500
1000
1500
2000
2500
3000
3500
ticks [0.5 us]

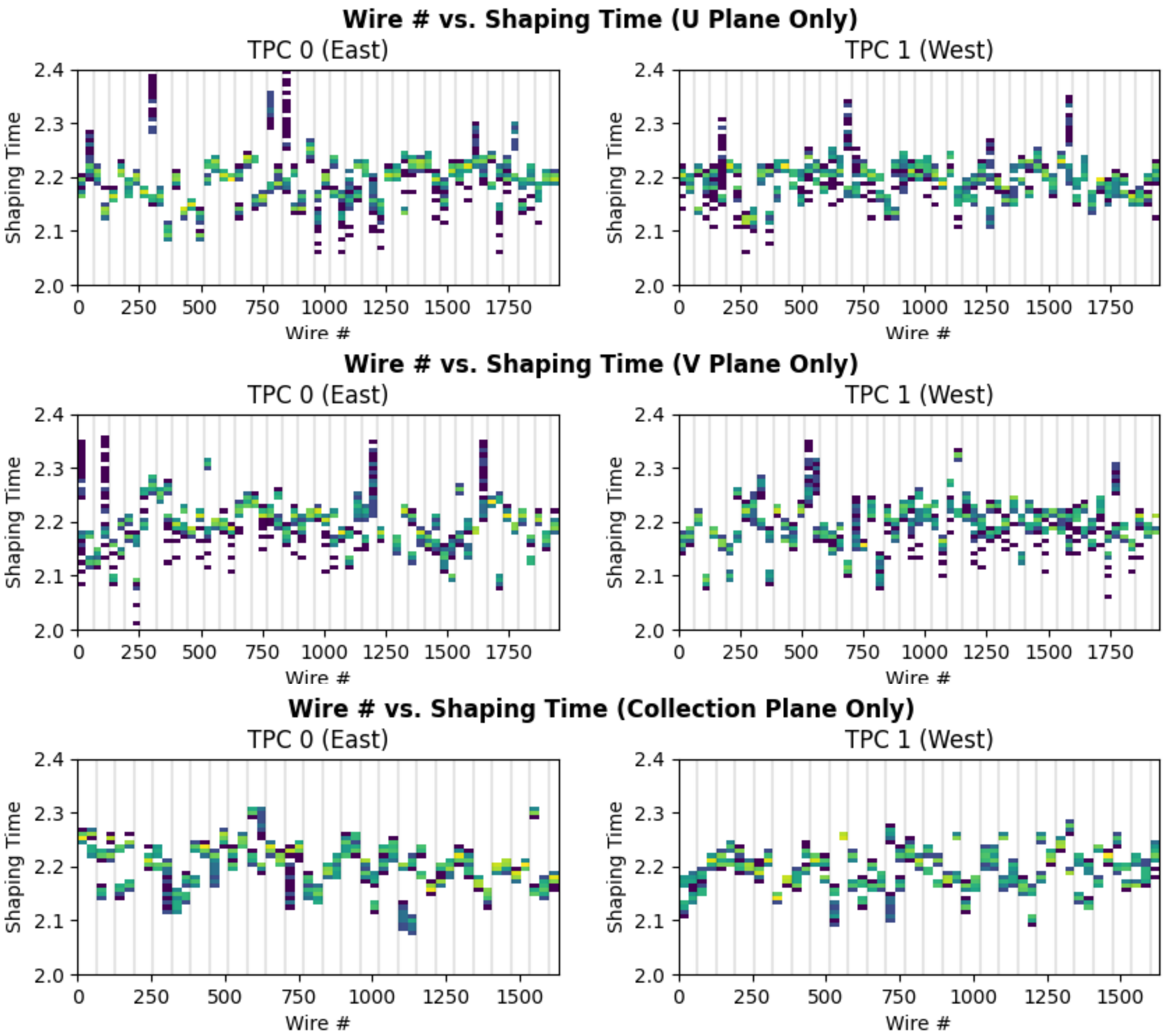

Distribution of Y-Intercept From Linear Fit (for Gain)
U Plane
V Plane
W Plane
y-intercept [ADC]
y-intercept [ADC]
y-intercept [ADC]
0
100
200
300
400
500
600
−30
−20
−10
0
10
20
30

Distribution of Gains Per Plane





Wire # vs. Gain (U Plane Only)
TPC 0 (East)
TPC 1 (West)
Wire # vs. Gain (V Plane Only)
TPC 0 (East)
TPC 1 (West)
Wire # vs. Gain (Collection Plane Only)
TPC 0 (East)
TPC 1 (West)
Gain
Wire #
15.0
14.5
14.0
13.5
13.0
0
250
500
750
1000
1250
1500
1750



Wire # vs. Shaping Time (U Plane Only)
TPC 0 (East)
TPC 1 (West)
Shaping Time
Wire #
Wire # vs. Shaping Time (V Plane Only)
TPC 0 (East)
TPC 1 (West)
Shaping Time
Wire #
Wire # vs. Shaping Time (Collection Plane Only)
TPC 0 (East)
TPC 1 (West)
Shaping Time
Wire #

Inverse Laplace Transformation of T(s)

```
In[3]:= InverseLaplaceTransform[A / (p0 + s) / (pi1^2 + (pr1 + s)^2) / (pi2^2 + (pr2 + s)^2) * (k3 + s) / (k4 + s) * (k5 + s) / (k6 + s), s, t]

Out[11]= A (e^(-k4 t) (-k3 k4 + k4^2 + k3 k5 - k4 k5) / ((k4 - k6) (k4 - p0) (k4^2 + pi1^2 - 2 k4 pr1 + pr1^2) (k4^2 + pi2^2 - 2 k4 pr2 + pr2^2)) + e^(-k6 t) (-k3 k5 + k3 k6 + k5 k6 - k6^2) / ((k4 - k6) (k6 - p0) (k6^2 + pi1^2 - 2 k6 pr1 + pr1^2) (k6^2 + pi2^2 - 2 k6 pr2 + pr2^2)) +
 e^(-p0 t) (-k3 k5 + k3 p0 + k5 p0 - p0^2) / ((k4 - p0) (-k6 + p0) (p0^2 + pi1^2 - 2 p0 pr1 + pr1^2) (p0^2 + pi2^2 - 2 p0 pr2 + pr2^2)) + (e^(-pr1 t) (pi1 ((pi1^2 + pr1^2) (2 k6 (pi1^2 + pr1^2) (pr1 - pr2) + k6 p0 (-pi1^2 + pi2^2 - pr1^2 + pr2^2) + (pi1^2 + pr1^2) (pi1^2 - pi2^2 + (pr1 - pr2) (2 p0 - 3 pr1 + pr2)) +
 k5 (2 pi1^2 (-2 pr1 + pr2) + p0 (pi1^2 - pi2^2 - 3 pr1^2 + 4 pr1 pr2 - pr2^2) + 2 pr1 (pi2^2 + 2 pr1^2 - 3 pr1 pr2 + pr2^2) + k6 (pi1^2 - pi2^2 + (pr1 - pr2) (2 p0 - 3 pr1 + pr2)))) + k4 ((pi1^2 + pr1^2) (2 (pi1^2 + pr1^2) (pr1 - pr2) + p0 (-pi1^2 + pi2^2 - pr1^2 + pr2^2)) +
 k5 (2 k6 (pi1^2 + pr1^2) (pr1 - pr2) - k6 p0 (pi1^2 - pi2^2 + pr1^2 - pr2^2) + (pi1^2 + pr1^2) (pi1^2 - pi2^2 + (pr1 - pr2) (2 p0 - 3 pr1 + pr2))) + k6 (-pi1^4 + pi1^2 (pi2^2 - 2 pr1^2 + 2 p0 pr2 + pr2^2) + pr1 (pr1 (pi2^2 - pr1^2 + pr2^2) - 2 p0 (pi2^2 - pr1 pr2 + pr2^2)))) +
 k3 (-(pi1^2 + pr1^2) (4 pi1^2 pr1 - 2 pi2^2 pr1 - 4 pr1^3 - 2 pi1^2 pr2 + 6 pr1^2 pr2 - 2 pr1 pr2^2 + p0 (-pi1^2 + pi2^2 + 3 pr1^2 - 4 pr1 pr2 + pr2^2) + k6 (-pi1^2 + pi2^2 - (pr1 - pr2) (2 p0 - 3 pr1 + pr2))) + k5 (-pi1^4 + pi1^2 (pi2^2 - 4 p0 pr1 + 10 pr1^2 + 2 p0 pr2 - 8 pr1 pr2 + pr2^2) +
 k6 (2 pi1^2 (-2 pr1 + pr2) + p0 (pi1^2 - pi2^2 - 3 pr1^2 + 4 pr1 pr2 - pr2^2) + 2 pr1 (pi2^2 + 2 pr1^2 - 3 pr1 pr2 + pr2^2)) + pr1 (2 p0 (pi2^2 + 2 pr1^2 - 3 pr1 pr2 + pr2^2) - pr1 (3 pi2^2 + 5 pr1^2 - 8 pr1 pr2 + 3 pr2^2))) + k4 (2 k6 (pi1^2 + pr1^2) (pr1 - pr2) +
 k6 p0 (-pi1^2 + pi2^2 - pr1^2 + pr2^2) + (pi1^2 + pr1^2) (pi1^2 - pi2^2 + (pr1 - pr2) (2 p0 - 3 pr1 + pr2)) + k5 (2 pi1^2 (-2 pr1 + pr2) + p0 (pi1^2 - pi2^2 - 3 pr1^2 + 4 pr1 pr2 - pr2^2) + 2 pr1 (pi2^2 + 2 pr1^2 - 3 pr1 pr2 + pr2^2) + k6 (pi1^2 - pi2^2 + (pr1 - pr2) (2 p0 - 3 pr1 + pr2)))))
 Cos[pi1 t] - ((pi1^2 + pr1^2) ((pi1^2 + pr1^2) (p0 (pi1^2 - pi2^2 - (pr1 - pr2)^2) + pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (-3 pr1 + 2 pr2)) + k6 (pi1^4 + (p0 - pr1) pr1 (pi2^2 + (pr1 - pr2)^2) - pi1^2 (pi2^2 - p0 pr1 + 2 p0 pr2 - 2 pr1 pr2 + pr2^2)) +
 k5 (-pi1^4 + (p0 - pr1) pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (pi2^2 - 3 p0 pr1 + 6 pr1^2 + 2 p0 pr2 - 6 pr1 pr2 + pr2^2) + k6 (p0 (pi1^2 - pi2^2 - (pr1 - pr2)^2) + pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (-3 pr1 + 2 pr2)))) +
 k4 ((pi1^2 + pr1^2) (pi1^4 + (p0 - pr1) pr1 (pi2^2 + (pr1 - pr2)^2) - pi1^2 (pi2^2 - p0 pr1 + 2 p0 pr2 - 2 pr1 pr2 + pr2^2)) + k5 ((pi1^2 + pr1^2) (p0 (pi1^2 - pi2^2 - (pr1 - pr2)^2) + pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (-3 pr1 + 2 pr2)) +
 k6 (pi1^4 + (p0 - pr1) pr1 (pi2^2 + (pr1 - pr2)^2) - pi1^2 (pi2^2 - p0 pr1 + 2 p0 pr2 - 2 pr1 pr2 + pr2^2))) + k6 ((pi1^2 + pr1^2) (pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (pr1 - 2 pr2)) - p0 (pi1^4 + pr1^2 (pi2^2 + (pr1 - pr2)^2) - pi1^2 (pi2^2 - 2 pr1^2 + 2 pr1 pr2 + pr2^2)))) +
 k3 ((pi1^2 + pr1^2) (-pi1^4 + (p0 - pr1) pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (pi2^2 - 3 p0 pr1 + 6 pr1^2 + 2 p0 pr2 - 6 pr1 pr2 + pr2^2) + k6 (p0 (pi1^2 - pi2^2 - (pr1 - pr2)^2) + pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (-3 pr1 + 2 pr2))) + k5 (5 pi1^4 pr1 - 3 pi1^2 pi2^2 pr1 - 10 pi1^2 pr1^3 + pi2^2 pr1^3 +
 pr1^5 - 2 pi1^4 pr2 + 12 pi1^2 pr1^2 pr2 - 2 pr1^4 pr2 - 3 pi1^2 pr1 pr2^2 + pr1^3 pr2^2 - p0 (pi1^4 + pr1^2 (pi2^2 + (pr1 - pr2)^2) - pi1^2 (pi2^2 + 6 pr1^2 - 6 pr1 pr2 + pr2^2)) + k6 (-pi1^4 + (p0 - pr1) pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (pi2^2 - 3 p0 pr1 + 6 pr1^2 + 2 p0 pr2 - 6 pr1 pr2 + pr2^2))) +
 k4 ((pi1^2 + pr1^2) (p0 (pi1^2 - pi2^2 - (pr1 - pr2)^2) + pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (-3 pr1 + 2 pr2)) + k6 (pi1^4 + (p0 - pr1) pr1 (pi2^2 + (pr1 - pr2)^2) - pi1^2 (pi2^2 - p0 pr1 + 2 p0 pr2 - 2 pr1 pr2 + pr2^2)) +
 k5 (-pi1^4 + (p0 - pr1) pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (pi2^2 - 3 p0 pr1 + 6 pr1^2 + 2 p0 pr2 - 6 pr1 pr2 + pr2^2) + k6 (p0 (pi1^2 - pi2^2 - (pr1 - pr2)^2) + pr1 (pi2^2 + (pr1 - pr2)^2) + pi1^2 (-3 pr1 + 2 pr2)))))) Sin[pi1 t])) /
 (pi1 (k4^2 + pi1^2 - 2 k4 pr1 + pr1^2) (k6^2 + pi1^2 - 2 k6 pr1 + pr1^2) (p0^2 + pi1^2 - 2 p0 pr1 + pr1^2) (pi1^4 - 2 pi1^2 (pi2^2 - (pr1 - pr2)^2) + (pi2^2 + (pr1 - pr2)^2)^2)) +
 (e^(-pr2 t) (-pi2 (k4 (-(pi2^2 + pr2^2) (p0 (pi1^2 - pi2^2 + pr1^2 - pr2^2) - 2 (pr1 - pr2) (pi2^2 + pr2^2)) + k5 ((pi1^2 - pi2^2 + (2 p0 + pr1 - 3 pr2) (pr1 - pr2)) (pi2^2 + pr2^2) + 2 k6 (pr1 - pr2) (pi2^2 + pr2^2) + k6 p0 (-pi1^2 + pi2^2 - pr1^2 + pr2^2)) +
 k6 (pi2^4 - pi2^2 (2 p0 pr1 + pr1^2 - 2 pr2^2) - pi1^2 (pi2^2 + pr2 (-2 p0 + pr2)) - (pr1 - pr2) pr2 (-2 p0 pr1 + pr2 (pr1 + pr2)))) + (pi2^2 + pr2^2) ((pi1^2 - pi2^2 + (2 p0 + pr1 - 3 pr2) (pr1 - pr2)) (pi2^2 + pr2^2) +
 2 k6 (pr1 - pr2) (pi2^2 + pr2^2) + k6 p0 (-pi1^2 + pi2^2 - pr1^2 + pr2^2) + k5 (k6 (pi1^2 - pi2^2 + (2 p0 + pr1 - 3 pr2) (pr1 - pr2)) + p0 (pi1^2 - pi2^2 + pr1^2 - 4 pr1 pr2 + 3 pr2^2) - 2 (pi2^2 (pr1 - 2 pr2) + pr2 (pi1^2 + pr1^2 - 3 pr1 pr2 + 2 pr2^2)))) +
 k3 ((pi2^2 + pr2^2) (k6 (pi1^2 - pi2^2 + (2 p0 + pr1 - 3 pr2) (pr1 - pr2)) + p0 (pi1^2 - pi2^2 + pr1^2 - 4 pr1 pr2 + 3 pr2^2) - 2 (pi2^2 (pr1 - 2 pr2) + pr2 (pi1^2 + pr1^2 - 3 pr1 pr2 + 2 pr2^2))) + k5 (pi2^4 - 2 p0 pi2^2 pr1 - pi2^2 pr1^2 + 4 p0 pi2^2 pr2 + 8 pi2^2 pr1 pr2 - 2 p0 pr1^2 pr2 - 10 pi2^2 pr2^2 +
 6 p0 pr1 pr2^2 + 3 pr1^2 pr2^2 - 4 p0 pr2^3 - 8 pr1 pr2^3 + 5 pr2^4 - pi1^2 (pi2^2 + (2 p0 - 3 pr2) pr2) + k6 p0 (pi1^2 - pi2^2 + pr1^2 - 4 pr1 pr2 + 3 pr2^2) - 2 k6 (pi2^2 (pr1 - 2 pr2) + pr2 (pi1^2 + pr1^2 - 3 pr1 pr2 + 2 pr2^2))) + k4 ((pi1^2 - pi2^2 + (2 p0 + pr1 - 3 pr2) (pr1 - pr2))
 (pi2^2 + pr2^2) + 2 k6 (pr1 - pr2) (pi2^2 + pr2^2) + k6 p0 (-pi1^2 + pi2^2 - pr1^2 + pr2^2) + k5 (k6 (pi1^2 - pi2^2 + (2 p0 + pr1 - 3 pr2) (pr1 - pr2)) + p0 (pi1^2 - pi2^2 + pr1^2 - 4 pr1 pr2 + 3 pr2^2) - 2 (pi2^2 (pr1 - 2 pr2) + pr2 (pi1^2 + pr1^2 - 3 pr1 pr2 + 2 pr2^2))))) Cos[pi2 t] +
 ((pi2^2 + pr2^2) ((pi2^2 + pr2^2) (p0 (pi1^2 - pi2^2 + (pr1 - pr2)^2) - (pi1^2 + (pr1 - pr2)^2) pr2 + pi2^2 (-2 pr1 + 3 pr2)) + k6 (-pi2^4 - (p0 - pr2) (pr1 - pr2)^2 pr2 + pi2^2 (2 p0 pr1 + pr1^2 - p0 pr2 - 2 pr1 pr2) + pi1^2 (pi2^2 + pr2 (-p0 + pr2))) + k5
 (-pi1^2 pi2^2 + pi2^4 - 2 p0 pi2^2 pr1 - pi2^2 pr1^2 - p0 pi1^2 pr2 + 3 p0 pi2^2 pr2 + 6 pi2^2 pr1 pr2 - p0 pr1^2 pr2 + pi1^2 pr2^2 - 6 pi2^2 pr2^2 + 2 p0 pr1 pr2^2 + pr1^2 pr2^2 - p0 pr2^3 - 2 pr1 pr2^3 + pr2^4 + k6 (p0 (pi1^2 - pi2^2 + (pr1 - pr2)^2) - (pi1^2 + (pr1 - pr2)^2) pr2 + pi2^2 (-2 pr1 + 3 pr2)))) +
 k4 (-k6 (pi2^2 + pr2^2) ((pi1^2 + (pr1 - pr2)^2) pr2 + pi2^2 (-2 pr1 + pr2)) + k6 p0 (pi2^4 + (pr1 - pr2)^2 pr2^2 - pi2^2 (pr1^2 + 2 pr1 pr2 - 2 pr2^2) + pi1^2 (-pi2^2 + pr2^2)) + (pi2^2 + pr2^2) (-pi2^4 - (p0 - pr2) (pr1 - pr2)^2 pr2 + pi2^2 (2 p0 pr1 + pr1^2 - p0 pr2 - 2 pr1 pr2) +
 pi1^2 (pi2^2 + pr2 (-p0 + pr2))) + k5 ((pi2^2 + pr2^2) (p0 (pi1^2 - pi2^2 + (pr1 - pr2)^2) - (pi1^2 + (pr1 - pr2)^2) pr2 + pi2^2 (-2 pr1 + 3 pr2)) + k6 (-pi2^4 - (p0 - pr2) (pr1 - pr2)^2 pr2 + pi2^2 (2 p0 pr1 + pr1^2 - p0 pr2 - 2 pr1 pr2) + pi1^2 (pi2^2 + pr2 (-p0 + pr2))))) +
 k3 ((pi2^2 + pr2^2) (-pi1^2 pi2^2 + pi2^4 - 2 p0 pi2^2 pr1 - pi2^2 pr1^2 - p0 pi1^2 pr2 + 3 p0 pi2^2 pr2 + 6 pi2^2 pr1 pr2 - p0 pr1^2 pr2 + pi1^2 pr2^2 - 6 pi2^2 pr2^2 + 2 p0 pr1 pr2^2 + pr1^2 pr2^2 - p0 pr2^3 - 2 pr1 pr2^3 + pr2^4 +
 k6 (p0 (pi1^2 - pi2^2 + (pr1 - pr2)^2) - (pi1^2 + (pr1 - pr2)^2) pr2 + pi2^2 (-2 pr1 + 3 pr2))) - k5 (p0 pi1^2 pi2^2 - p0 pi2^4 - 2 pi2^4 pr1 + p0 pi2^2 pr1^2 - 3 pi1^2 pi2^2 pr2 + 5 pi2^4 pr2 - 6 p0 pi2^2 pr1 pr2 - 3 pi2^2 pr1^2 pr2 - p0 pi1^2 pr2^2 + 6 p0 pi2^2 pr2^2 +
 12 pi2^2 pr1 pr2^2 - p0 pr1^2 pr2^2 + pi1^2 pr2^3 - 10 pi2^2 pr2^3 + 2 p0 pr1 pr2^3 + pr1^2 pr2^3 - p0 pr2^4 - 2 pr1 pr2^4 + pr2^5 + k6 (-pi2^4 + (p0 - pr2) (pr1 - pr2)^2 pr2 + pi1^2 (pi2^2 + (p0 - pr2) pr2) + pi2^2 (2 p0 pr1 + pr1^2 - 3 p0 pr2 - 6 pr1 pr2 + 6 pr2^2))) +
 k4 ((pi2^2 + pr2^2) (p0 (pi1^2 - pi2^2 + (pr1 - pr2)^2) - (pi1^2 + (pr1 - pr2)^2) pr2 + pi2^2 (-2 pr1 + 3 pr2)) + k6 (-pi2^4 - (p0 - pr2) (pr1 - pr2)^2 pr2 + pi2^2 (2 p0 pr1 + pr1^2 - p0 pr2 - 2 pr1 pr2) + pi1^2 (pi2^2 + pr2 (-p0 + pr2))) + k5 (-pi1^2 pi2^2 + pi2^4 - 2 p0 pi2^2 pr1 -
 pi2^2 pr1^2 - p0 pi1^2 pr2 + 3 p0 pi2^2 pr2 + 6 pi2^2 pr1 pr2 - p0 pr1^2 pr2 + pi1^2 pr2^2 - 6 pi2^2 pr2^2 + 2 p0 pr1 pr2^2 + pr1^2 pr2^2 - p0 pr2^3 - 2 pr1 pr2^3 + pr2^4 + k6 (p0 (pi1^2 - pi2^2 + (pr1 - pr2)^2) - (pi1^2 + (pr1 - pr2)^2) pr2 + pi2^2 (-2 pr1 + 3 pr2)))))) Sin[pi2 t])) /
 (pi2 (pi1^4 - 2 pi1^2 (pi2^2 - (pr1 - pr2)^2) + (pi2^2 + (pr1 - pr2)^2)^2) (k4^2 + pi2^2 - 2 k4 pr2 + pr2^2) (k6^2 + pi2^2 - 2 k6 pr2 + pr2^2) (p0^2 + pi2^2 - 2 p0 pr2 + pr2^2)))
```