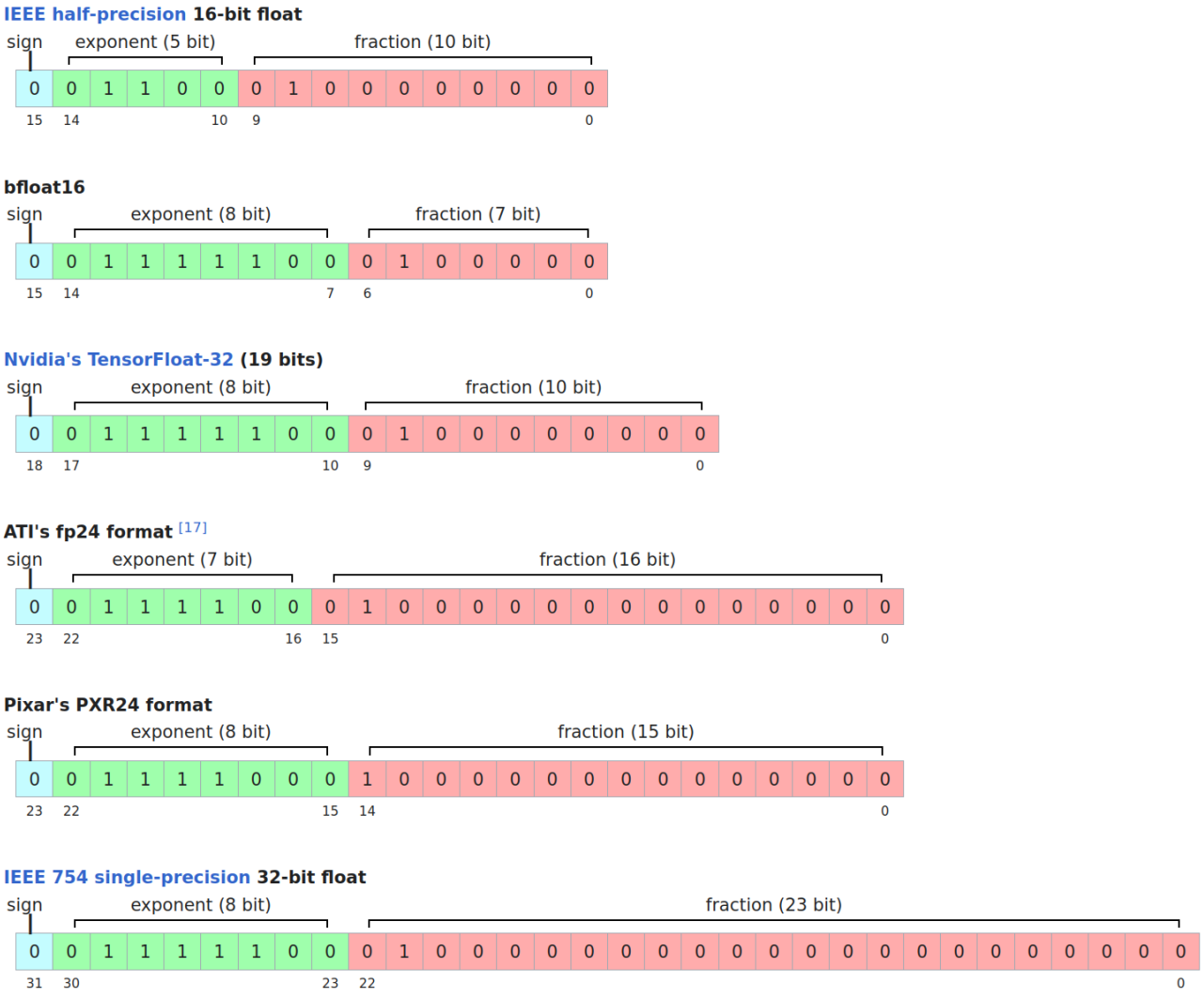
Reduced precision floating points artifacts

Mikael Mieskolainen, 23/08/2025

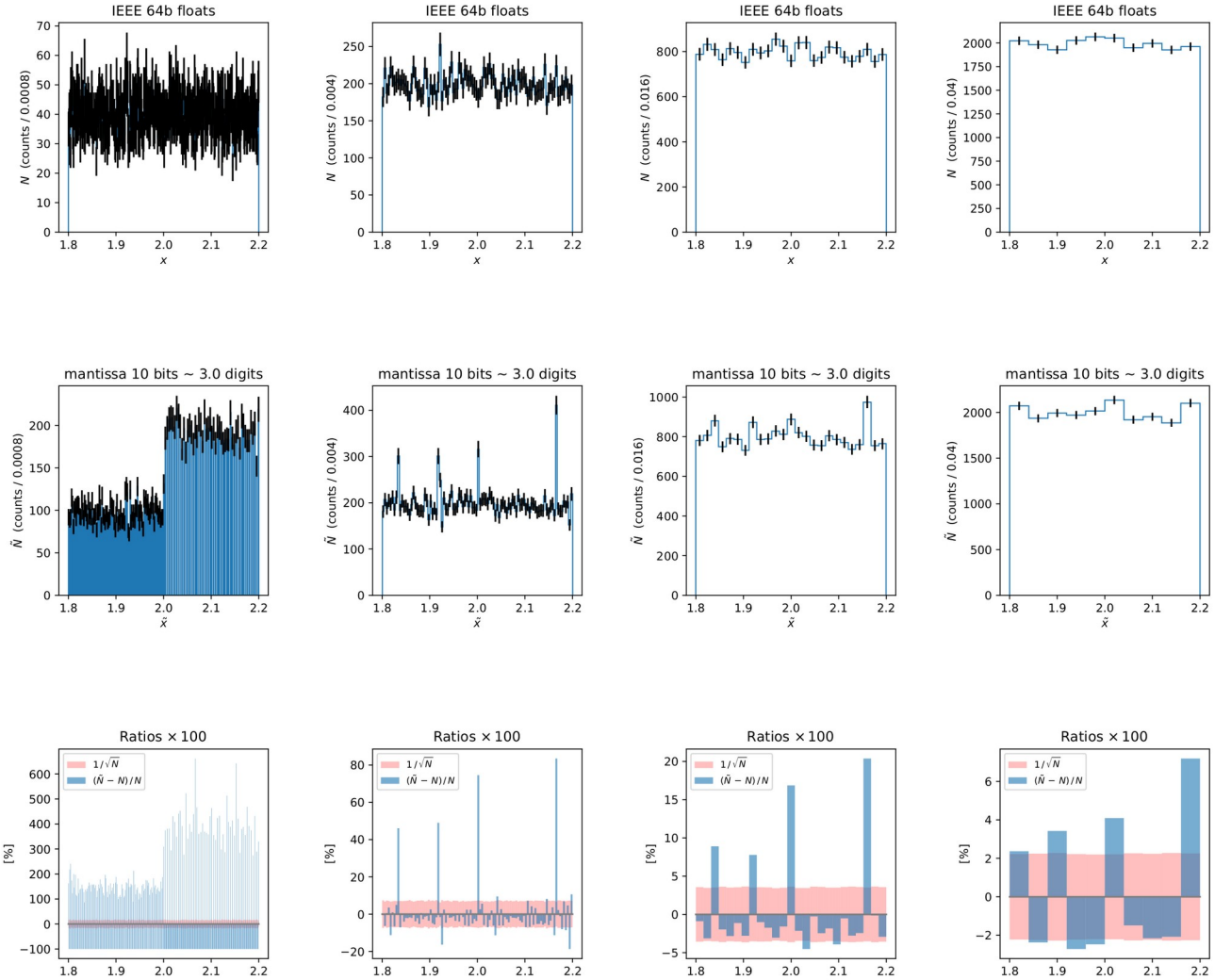
Something to remember with NanoAOD reduced mantissa bit precision variables. Original MiniAOD is float32.

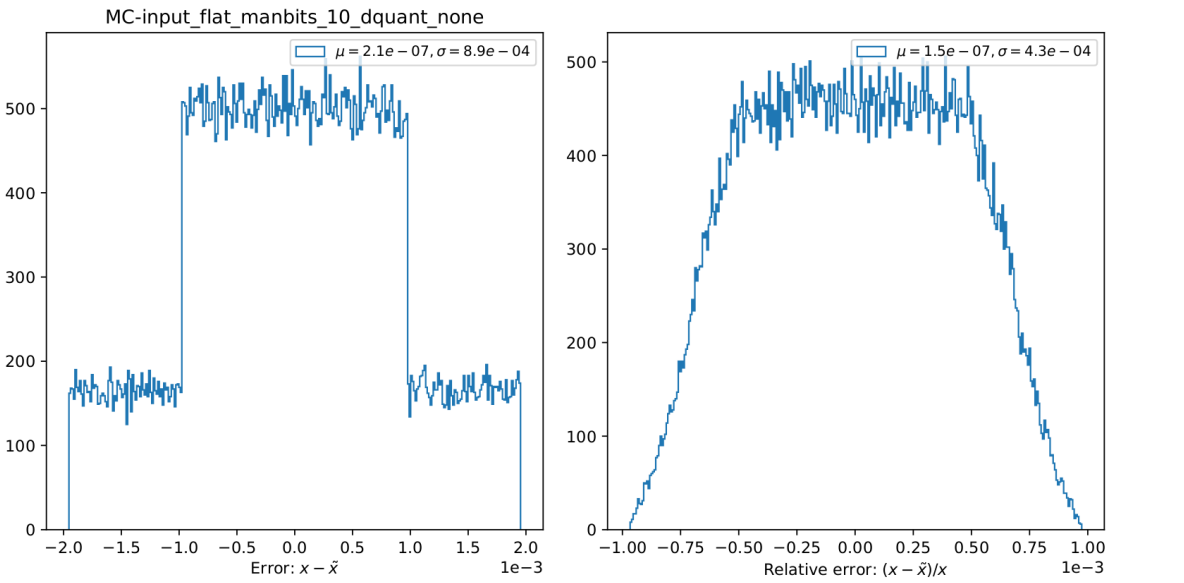
Reproduce:
https://github.com/mieskolainen/icenet/blob/master/icefit/mantissa.py



Numpy default is np.float64 (1 + 11 + 52 = 64). AI/ML can handle much lower precision in the model parameters (such as bfloat16 from Google) because training can compensate, but not at the input x level necessarily. E.g. normalizing flows can be especially sensitive to this.

EXAMPLE: reduce to 10 mantissa (fraction) bits, the scale dependent loss of absolute precision gives a step e.g. at x = 2.0. This is visible if histogram binning is very fine, but periodic convolution peaks appear even at wide binning …





DEQUANTIZATION: add relative std Gaussian noise (10^{-3}) into 10 mantissa bit floats as *a post fix* → steps and peaks removed, but the price to pay is in precision (resolution). A proper fix is to keep enough bits ~ IEEE float32 are reasonably safe. Some numerical computation applications require even float128.