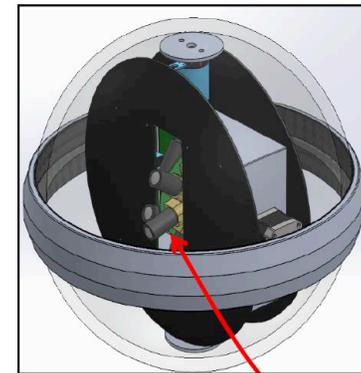


Readout Instrumentation & DAQ for Future Detectors

- Large-volume neutrino detectors can benefit from
 - Moving readout electronics as close to sensor as possible
 - ▶ Minimizes analog signal pathlength \Rightarrow improves signal quality
 - ▶ Simplifies cabling/installation
 - ▶ Distributes computational load for pulse extraction
 - ▶ No HV near humans
 - Encapsulation of photosensors
 - ▶ Reduces implosion risk (PMTs); isolates sensor from chemically aggressive detector fluids
 - ▶ Penetrator-less design for power & data would simplify installation and maintenance
 - ▶ Vessel contains some radioactive contaminants (e.g., radon)
 - ▶ Flexible: standardized platform to house different light sensors, other sensors (T , \vec{B} , \vec{a} , $p\dots$), calibration devices
 - ▶ Can employ local cooling of sensors to reduce intrinsic dark rate



Multi-PMT module (Hyper-K)



Pencil-beam DOM (IceCube)

Readout Instrumentation & DAQ for Future Detectors

● Commercial Off-the-Shelf vs. Custom ASIC

- Up to $O(5k)$ channels, COTS can be cost-competitive

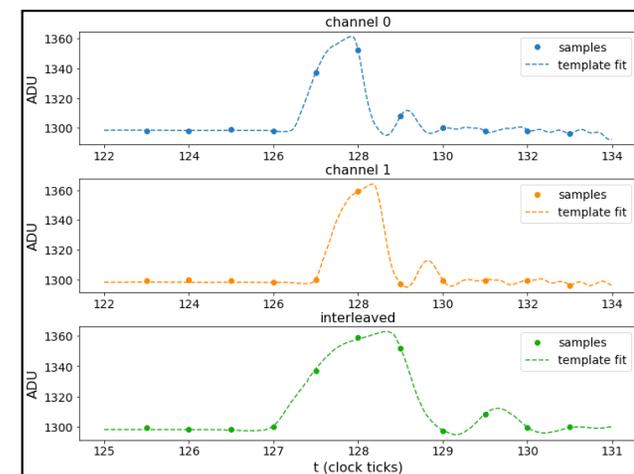
- ▶ With 250 MSPS ADCs (@\$20)

- ~ 100 ps timing resolution
- ~ 100 p.e. dynamic range
- triggerless, zero deadtime, deep buffers

- ▶ Can interleave: 2×250 MSPS cheaper than 500 MSPS ADC

- Above $O(5k)$ channels, custom ASIC very likely cheaper

- ▶ Large channel count amortizes large up-front engineering cost
- ▶ ASICs using switched capacitor arrays or analog q/t extraction exist & work



Interleaving w/template fit (WATCHMAN)

Firmware for Future Detectors

- Firmware for managing data flow and for pulse extraction is complex w/O(10-100k) lines
 - Real firmware expertise is limited
 - Learning curve is steep
- Tasks are often similar from neutrino experiment to experiment
 - Manage signal data
 - ▶ Apply calibrations
 - ▶ Buffer to on-board memory
 - ▶ Package and ship downstream
 - For readouts with waveform digitization, firmware used to extract pulse info (t, Q)
 - ▶ Potentially after applying DSP algorithms to reduce noise
- A public firmware library would reduce time for design, testing, etc. of both firmware itself and the electronics it is used to test/verify
 - Would also allow for early optimization of FPGA size, saving design effort and per-channel cost