DUNE-SP Data Selection Summary
DAQ Consortium

1 DUNE Data Selection Requirements and Overview

The most critical requirement for the DUNE Data Selection system (the DUNE “trigger”) is that it have an efficiency for triggering on interactions by beam neutrinos within the detector’s active volume that is high enough that it is a sub-dominant effect on DUNE’s physics sensitivity. The energy threshold—based on *visible* energy deposit—is expected to be 100 MeV, low enough for all current long-baseline oscillation analyses, but this is an achievable number and the ultimate number could be lower.

A second important requirement is that DUNE trigger on bursts of events consistent with neutrino interactions from a supernova explosion in the Milky Way. The “fake” rate for such bursts is targeted to be less than 1/month, and the target efficiency is that we are able to detect such burst for more than 95% of the Milky Way galaxy.

Additional requirements are that DUNE will trigger with high efficiency on non-beam, high-energy events such as nucleon decay candidates, atmospheric neutrinos, and cosmic rays, with the same threshold as beam events, 100 MeV of visible energy.

We also have as a requirement that DUNE is capable of triggering on low-energy events such as solar neutrinos and single supernova neutrino events, with a threshold around 10 MeV. Lower thresholds are possible, and will depend on the level of radiological backgrounds in DUNE, and slightly higher thresholds will not compromise the primary physics missions of DUNE.

All of these requirements are derived from higher-level requirements of the DUNE physics program.

The DUNE Data Selection System satisifies these requirements with a hierarchical design. The design allows great flexibility for changing trigger criteria and allows for development of increased sophistication for triggering on challenging low-energy events. The slow nature of the TPC, the full digitization of every front-end signal, and the expected low trigger rate for the primary events of interest, allows the Data Selection system to operate entirely as software processing on waveforms from the TPC and the Photon Detection System (PDS). Figure 1 shows a block diagram of the data selection system.

The first stage of Data Selection is generation of *Trigger Primitives*, which for the TPC are provided for each “hit” collection wire. Trigger Primitives provide summary information such as the start time of the pulse on a collection wire, the total charge or peak of the

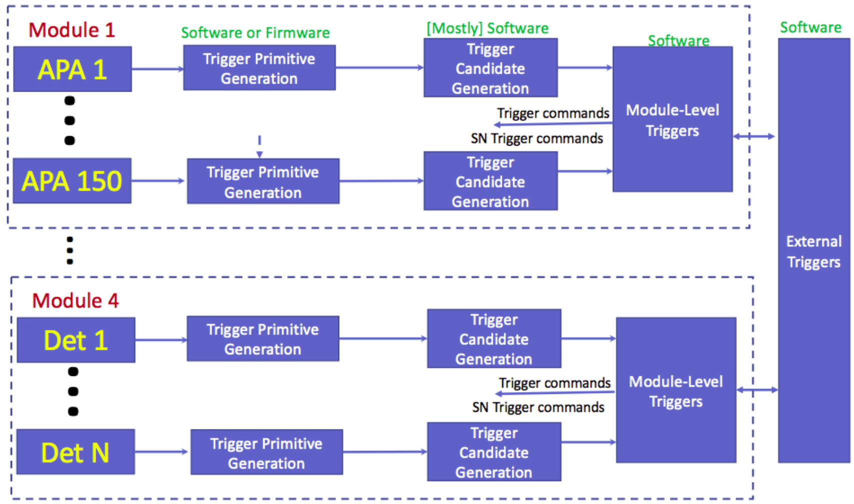

Figure 1: Simple block diagram of data selection system

collection-wire pulse, and the time the pulse stays above some pre-determined threshold. Trigger primitives from the PDS are also generated, in PDS hardware.

Trigger Primitives are streamed from the front-end DAQ readout to an "APA-level" system that examines those for one APA (or a very small number of APAs) and determines whether the Primitives from that APA constitute a potential *Trigger Candidate*. Trigger Candidates are generated by looking for clusters of hit wires in a narrow time window, the total charge of one or a cluster of hits wires, or a large time-over-threshold. Trigger Candidates will be generated with both a "low" and "high"-energy threshold. The former will be used for supernova burst triggering, the latter for triggers on beam, atmospheric neutrinos, cosmics, and other high-energy events. Thus, a Trigger Candidate includes a rudimentary estimate of its energy, so that further downstream a decision can be made on whether a particular Trigger Candidate should be counted as part of a possible supernova burst, or worthy of triggering an entire 10 ktonne module in its own right. For example, a candidate with an estimated energy of 100 MeV would trigger an entire module, while one with an estimated energy of 3.5 MeV would only be considered as potentially part of a supernova burst.

Trigger candidates are passed upstream to a "Module Level" trigger that collects candidates and makes the final decision on whether a valid module-wide trigger has occured, or a possible supernova burst. The Module Level trigger passes *Trigger Commands* back down to the DAQ readout, which then saves a pre-determined amount of data depending on the type of Module Level trigger that has occurred. The Module Level also takes input from an "External Trigger," which may include information from other modules, or even other detectors, and it sends information to the External Trigger so that other modules or detectors can be triggered.

An important point for the purposes of this document and our planning, is that for physics-generated Module Level triggers (those that are triggering based on activity in the detector) we read out an entire 10 ktonne module for a time that is about twice the maximum drift time (a total of 5.4 ms). The reason for this approach is firstly that gammas and neutrons from the initial interaction carry important energy information, and the neutrons can travel quite far in the detector after they thermalize. Secondly, this approach simplifies our read out because it requires no development of zero-suppression algorithms near the front-end of the DAQ, where processing resources are in high demand.

Not shown in Figure 1 is the final stage of data selection, the “High Level Trigger” or “HLT.” The High Level Trigger acts on events that have already been triggered, read out, and built by the Event Builder. The HLT will serve several purposes. First, it can limit the total triggered data rate that goes to disk, by applying more refined selection criteria that will remove events that are clearly uninteresting. For example, instrumentally-generated events (like the “streamers” seen in ProtoDUNE) might lead to Trigger Candidates and a Module Level Trigger, but on more detailed inspection be clearly non-physics events. The HLT can also be used to reduce the triggered data set through identification of interesting activity—for any given neturino interaction, most APAs will see nothing but noise and radiological backgrounds. Thus saving that information does not improve the DUNE physics program. Beacuse the HLT is in the DAQ back-end and operates on event-built data, and the rate of these is low, developing sophisticated algorithms to eliminate empty APAs from the data stream is possible.

There are several critical parameters that impact data selection, and we summarize them below.

Table 1:

Parameter	Value
ADC Dynamic Range	12 bits
ADC Sampling Rate	2 MHz
Supernova Burst Readout Window	30 s
Total Number of Channels/Module	384000
Number of Channels/APA	2560
Event Builder Throughput	800 MB/s/core

2 TPC Trigger Primitive Generation

Details of Trigger Primitive generation can be found in Phil Rodrigues' technical note on DocDB #11236. Trigger Primitives are created in the front-end of the DAQ system, and for the single-phase detector they are generated purely on collection wire data. Although in principle Trigger Primitives can be generated from induction wire data, induction waveforms tend to be noisier, and are bipolar, and from a triggering standpoint they are not necessary.

Trigger Primitive generation operates on "raw" collection-wire waveform data. Data from the detector passes through the DUNE Warm Interface Board (WIB) where it is translated from copper to optical signals. The waveform data then flows to the front-end of the DAQ. For each channel, the front-end of the DAQ must remove the baseline (pedestal), filter noise possibly including coherent noise, and then look for signals above a threshold that is low enough—roughly 1/4 MIP-equivalent—that we will have good signal efficiency for events of interest, but high enough that the rate of "hits" that are due purely to noise or radiologicals, are tolerable. At zero threshold, the rate of ^{39}Ar in DUNE is expected to be about 100 Hz per collection wire, and a rate this high does not cause any problems for downstream algorithms. Once a "hit" is found on a wire, the channel number, time of the threshold crossing, pulse integral ("summed ADC"), and time-over-threshold is recorded and sent out as a Trigger Primitive.

The algorithm for removing the pedestal is detailed in DocDB #11236. It begins by finding the median of the waveform baseline using a "frugal streaming" approach that approximates the median at very low computational cost. Such a method is intrinsically a high-pass filter and thus, without adjustment, when operating in the region of a true signal it effectively differentiates it, losing sensitivity. A "lookahead" approach suggested by Brett Viren has been implemented that stops updating the pedestal when it becomes clear that a potential signal has been found. Figure 2, from #DocDB 11236, shows raw FD single-phase waveforms generated in Monte Carlo, with the frugal streaming algorithm's baseline indicated.

Based on both MiroBooNE and ProtoDUNE, it is likely that there will be some degree of coherent noise in the waveforms, which is dangerous because it can lead to a high rate of false hits, and simple filtering will not remove it if it is below the anticipated bandwidth of physics signals (about $0 - 200$ kHz). The MicroBooNE Collaboration has implemented a filtering approach that finds the median of waveforms of nearby channels in a particular time tick and subtracts this value from each individual channel. Additionally, "harmonic" noise is removed with a fine-grained notch filter with notches about 1 kHz or narrower. The danger in coherent noise removal is sensitivity to isochronous tracks, but few events have entirely isochronous tracks with the possible exception of some low-energy events. Nevertheless, we

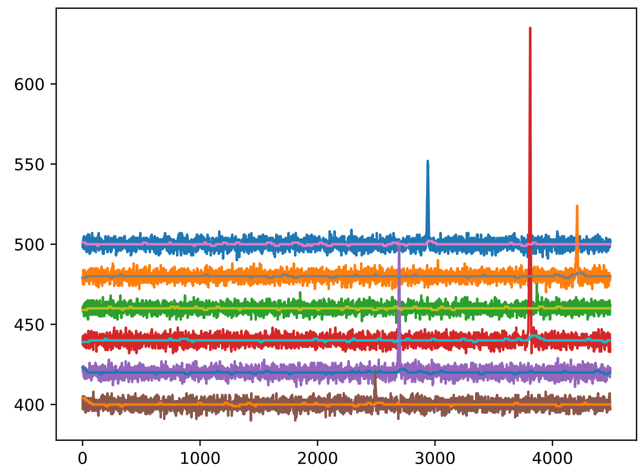

Figure 2: Raw FD MC waveforms plus pedestal estimated with frugal pedestal algorithm.

will need to test such algorithms, using ProtoDUNE data and simulation as testbeds, to determine how necessary it is and what its impact will be.

The removal of non-coherent noise has an advantage for Trigger Primitive generation that it may not for more complete downstream analyses. While the bandwidth of the DUNE electronics is set to be large enough that detailed structure in each waveform can be used for precision reconstruction, at the Trigger Primitive level high frequency information is not necessary, as the bulk of the signals have waveforms whose fundamentals lie well below 200 kHz (the system bandwidth goes up to 1 MHz for the 2 MHz sampling rate). Therefore removing broad-spectrum noise increases the signal/noise, making hit detection easier at this level.

We have used in our Trigger Primitive generation a finite impulse response (FIR) filter tuned to be a low-pass filter. More complete details of the algorithm can be found in DocDB #11236. We have restricted ourselves to a relatively simple FIR filter of just 7 taps, although we have explored filters with other complexities. DUNE waveforms are integer-valued, while the filter coefficients nominally can take on floating-point values that sum to 1.0. To allow the filtering to be done in fast integer arithmetic, the filter coefficients are rounded to two decimal places (by multiplying by 100 and rounding). Figure 3 from DocDB#11236 shows the effects of the 7-tap IR filter on Monte Carlo waveforms, indicating the large increase in signal-to-noise.

After pedestal subtraction and filtering, the Trigger Primitive algorithm searches for “hits” on each collection wire. The algorithm is by intention very simple, and is done

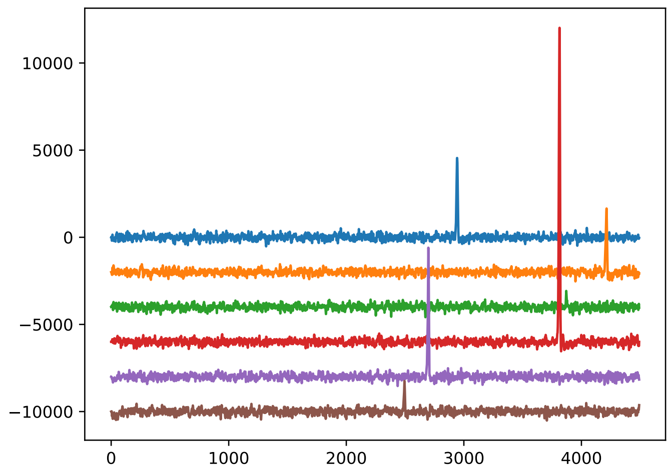

Figure 3: FD MC waveforms after pedestal subtraction and filtering with a 7-tap FIR filter.

as a simple threshold-discriminator. Once a sample in the waveform crosses a predetermined threshold in ADC counts, a "hit" is started and the pulse is integrated until it again drops below threshold. The time-over-threshold is recorded, as is the start time of the hit. These three pieces of information—pulse integral, start time, and time-over-threshold—form a nearly orthogonal "basis set" for the various track geometries in the detector. As Figure 4 shows, tracks that travel along the beam direction and are therefore perpendicular to the collection wires, have the largest number of adjacent "hits" (indicated by channel number and start time). Tracks that travel vertically, along the collection wires, have a large pulse integral (charge) on one or a few wires (depending on the level of diffusion). Tracks that are normal to the wire plane create long pulse widths (time-over-threshold), as well as having large charge on one or a small number of wires. Real tracks are linear combinations of these basis tracks, and thus combinations of the information in the trigger primitives can be used to find them efficienctly. For tracks that produce very long pulses (tracks that travel normal to the wire plane for long distances), a trigger primitive "time out" will be used that will send a primitive once after a pulse remains above threshold for a certain duration, and then send another if the pulse continues to be above threshold. For such tracks, therefore, many Trigger Primitives may be sent. Figure 5, taken from DocDB #11236, shows on the left the efficiency for detecting hits in MIP-equivalent units and in total number of drift electrons, as a function of Trigger Primitive threshold, for the Monte Carlo simulation's (LArSoft's) default noise levels (no coherent noise is simulated). A MIP energy deposition near the wire

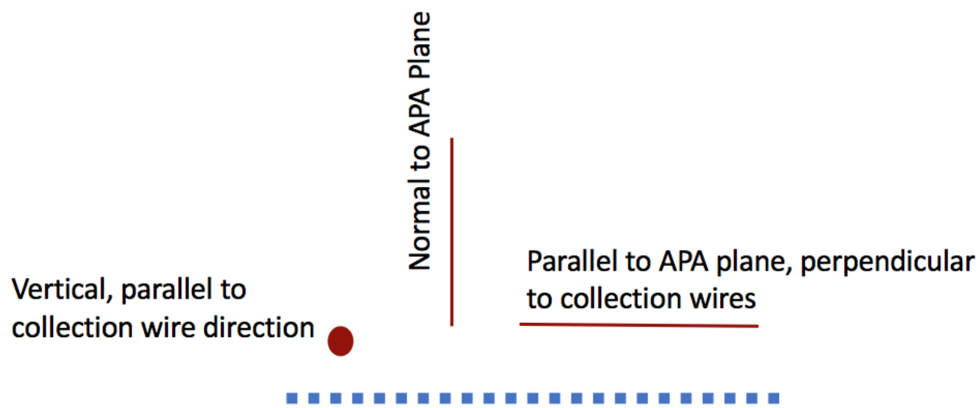

Figure 4: Cartoon of "basis set" of track geometries. Tracks normal to the APA plane produce very long pulses (high time-over-threshod), tracks that are parallel to the APA plane but perpendicular to the collection wires (indicated by blue dots) hit many wires, and tracks parallel to the collection wires produce large charges on a single or a few wires.

plane (the anode) produces a pulse height that is about 100 ADC counts above pedestal. In the Monte Carlo, which makes assumptions about the electron lifetime in DUNE (worse than what was achieved in ProtoDUNE), the same MIP at the cathode leads to a pulse whose height is about 40 ADC counts. So as can be seen in Figure 5, a threshold of 10 ADC counts is 1/4 of a MIP for deposits that originate at the cathode. The right side of Fig. 5 shows the impact of threshold on simulated radiological decays and supernova signal interactions ("Marley events"). At the 1/4 MIP threshold of about 10 ADC counts, the ^{39}Ar dominates the rate, at about 50 kHz/APA. Noise does not appear in Fig. 5 because the impact of the 7-tap low-pass FIR filter.

A test of the robustness to noise was done by increasing the noise rate to about 50% above the LArSoft default (and higher than the noise level on most of ProtoDUNE's APAs). Figure 6 shows the same plots as Fig. 5 with this increased noise. We can see that with this very high level of noise, we start to see noise-only Trigger Primitives being generated, but by the 1/4 MIP threshold of 10 ADC counts, the rate is already below the dominant radiologicals.

Trigger Primitives can be generated in a front-end FPGA, a GPU, or a CPU. We have tested this in all three hardware scenarios and have found that in all cases (with enough resources), Trigger Primitive generation can keep up with the detector data rate. Generation within CPUs may be the easiest to implement, given the rest of the DAQ architecture, and the algorithm described above has been tested on CPUs and found that with 4 cores/APA to keep up, the algorithm can keep up. Figure 7 produced by Phil Rodrigues shows the full

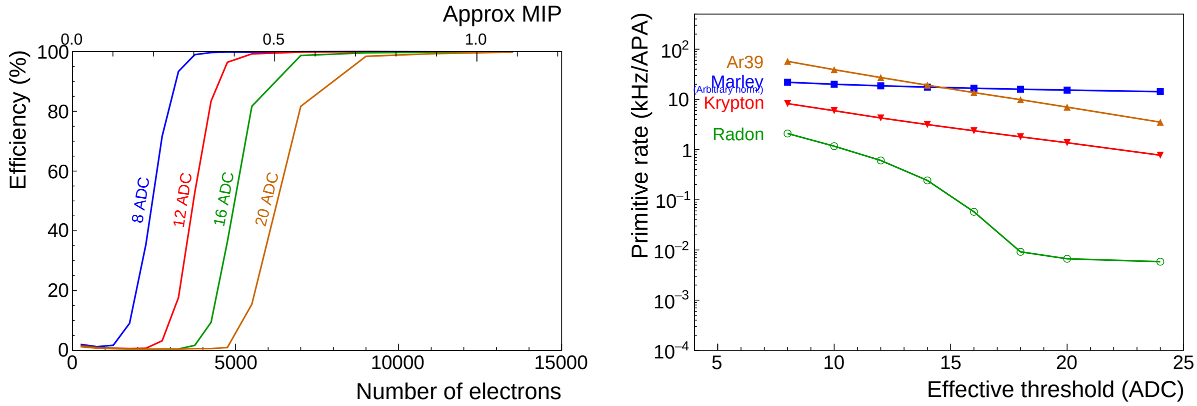

Figure 5: Efficiency and hit rate, default noise. The rate of hits due to noise (ie, not matched with any true energy deposition) is too low to show up on this plot.

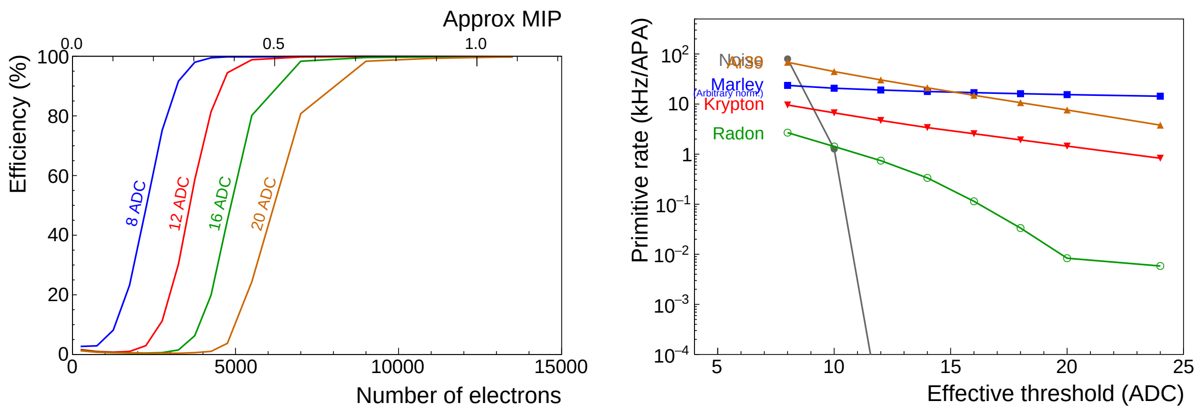

Figure 6: Efficiency and hit rate, with default noise increased by 50%.

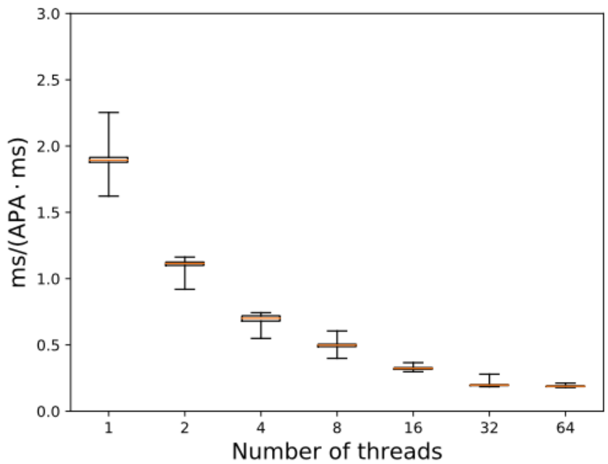

Figure 7: Time for Trigger Primitive generation in milliseconds of processing over APA milliseconds, as a function of the number of threads. The test was done on an existing Xeon Gold 6140 system. The boxes indicate the median and interquartile range of times for various iterations of the test, and the whiskers indicate the minimum and maximum after 1000 repeats of thet test. By 4 threads (cores), the algorithm can keep up with detector data rate.

curve of milliseconds/APA millisecond for the algorithm and the fluctuations in time over many tests, as a function of the number of running threads (assuming one thread/core). The tests were done on an existing system at CERN, a Xeon Gold 6140 with 72 cores, and the start time was with data already in memory formatted as `short ints`.

Tests of Trigger Primitive generation on very early ProtoDUNE data have also been done, as shown in Figure 8. ProtoDUNE is very different than DUNE because of its surface location, and this can be seen in the "floor" on the trigger primitive rate above a threshold of about 16 ADC counts. In addition, for the data shown in Fig. 8, there are known "noisy" channels that have not yet been removed. Although the total rates are much higher than we anticipate for DUNE, the algorithm itself seems to be working and as ProtoDUNE continues to run and improve its noise understanding, we expect to see these rates come down somewhat.

3 TPC Trigger Candidate Generation

Trigger Candidates are created from Trigger Primitives, by clustering hits in time and space. The Trigger Candidates are created in CPUs, with two APAs mapped into a single multi-core server. A Trigger Candidate includes the list of primitives that went into creating it, a tag as to whether this candidate is considered "high energy" and thus likely a beam event,

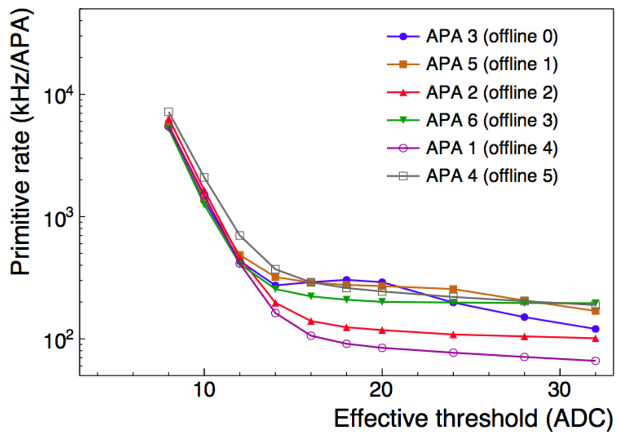

Figure 8: Trigger Primitive rate from early ProtoDUNE data. The "floor" on the rate above about 16 ADC counts is due in part to the high rate of cosmics in ProtoDUNE. In addition, known "noisy" channels have not been removed from these curves.

cosmic ray, or other physics, or "low energy" and thus potentially a supernova neutrino interaction that might be part of a burst. The "threshold" in each cases (high or low energy) is determined by the overall DAQ and Physics requirements of DUNE. At high energies, we want high efficiency by 100 MeV of visible energy deposit, while at low energies we want to ensure high efficiency for supernova burst detection. A decision on a supernova trigger depends on two parameters: the efficiency for detecting *single* low-energy interactions, and the total number of such interactions in a given time window. Thus even for a low single-interaction efficiency, the efficiency for supernova burst detection can be high. Although we ultimately anticipate a single Trigger Candidate algorithm running on Trigger Primitives, for how we have developed slightly distinct algorithms for generating high-energy Trigger Candidates and low-energy (supernova bust interaction) candidates.

In all cases, we aim for algorithms that are as simple as possible, to ensure that the computing resources are affordable and that latencies are within the overall specifications for the DUNE DAQ. More sophisticated approaches will certainly be developed, and if tested successfully will be implemented later in the project.

3.1 High Energy Trigger Candidate Generation

Full details of the generation of high-energy Trigger Candidates can be found in David Last and David Rivera's technical note, DocDB #11215. While the beam timeline will be available to the DAQ (through the DUNE Timing System), we have designed algorithms that will satisfy our requirements without resorting to the use of that timeline. Thus "beam on" and "beam off" triggering is identical. Should there be an advantage for the physics to include the beam timeline in the trigger, the system as designed can accommodate it easily. Triggering *only* on the beam timeline—taking data on every single beam gate—would exceed the overall data volume allocated for DUNE.

The challenge for high-energy Trigger Candidate generation is first ensuring that the rate of Trigger Candidates is low enough that we do not overwhelm the throughout into the event builder, nor wind up with too large a data volume written to disk. We anticipate that the highest rate of high-energy events in DUNE will be cosmic rays at 4500/day, thus any algorithm must lead to Trigger Candidates for all possible signals that are not a significant addition to this rate. A second challenge is achieving high efficiency at the required 100 MeV visible energy threshold. We target a 50% efficiency at 10 MeV of visible energy so as to ensure very high efficiency at 100 MeV, but in fact the definition of efficiency itself, and of visible energy, require some thought, particularly at low energies. DUNE plans to analyze neutrino data to below 1 GeV of neutrino energy, but 1 GeV neutrinos can produce events outside the active volume and have a lepton or other secondaries enter the volume and deposit some energy. Neutrinos may interact within the active volume and have their secondaries leave. In some cases, neutrinos may scatter via a neutral-current interaction, liberating neutrons but nothing else, leading to an apparent total visible energy deposit above 10 MeV, yet such events are not envisioned in any long-baseline oscillation analysis. In addition, 10 MeV of MIP-equivalent energy deposit (like what a muon would deposit in the active volume if it left after 5 cm or so) leads to a different amount of deposited charge, with different topological characteristics, than that of a 10 MeV electron. Thus the efficiency for the hypothetical 10 MeV muon energy deposit could be different than for the 10 MeV electron.

For the purposes of designing and testing Trigger Candidate algorithms, we define efficiency in a given energy bin to be the total number of neutrino interactions within the active volume that leads to a *MIP-equivalent visible energy* of that bin.

To generate Trigger Candidates, we count the number of adjacent wires with Trigger Primitives within a given time window, nominally 50 μs. We take the cluster and then apply the following criteria, independently of one another:

- Total cluster summed ADC $\geq$ 7000 Counts
- Number of Hit Wires $\geq$ 8
- Maximum Summed ADC in cluster $\geq$ 6500 Counts
- Maximum Time Over Threshold in Cluster $\geq$ 45 ticks

If a cluster of adjacent hits passes any one of these thresholds, then a Trigger Candidate is generated.

With these thresholds, the rate of fake triggers from radiologicals is found to be less than 10% of the cosmic-ray trigger rate. Figure 9 shows the trigger efficiency curve for beam neutrino interactions, as a function of visible energy. We see that in the current algorithm we appear to miss the target of 50% efficiency at 10 MeV visible energy, while getting to very high efficiencies near 100 MeV (which is the requirement). The reason for

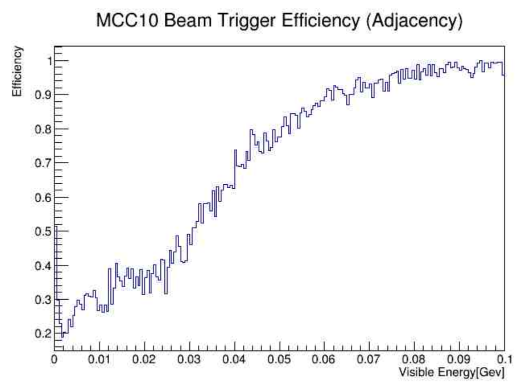

Figure 9: Efficiency as a function of Visible Energy for un-oscillated beam events.

the lower-than-expected efficiency at 10 MeV is due to the fact that for beam neutrinos, the dominant source of 10 MeV visible energy deposits are neutral-current interactions liberating neutrons. The neutrons create a total of 10 MeV of visible energy deposit, but these deposits are scattered across many different wires, and adjacency requirements for clustering are not satisfied. While these events are not likely to be used for primary oscillation analyses, they are nevertheless interesting, and we are investigating ways in which we can improve the efficiency at these low energies. For beam events, we also have the option of lowering some of the thresholds when we expect to see beam, based on the accelerator timeline from the DUNE Timing System, and thus increase efficiency across the entire window.

We have applied the same algorithm to atmospheric neutrino events, and find very similar efficiency curves, although at high energies the efficiency is slightly larger than for beam events.

The generation of Trigger Candidates will be done in CPUs, with two APAs mapped to a single multi-core server. To ensure that we can keep up with data from the detector, we need to be able to generate Trigger Candidates for a given slice of drift time faster than the drift time itself. For example, if we view the data as being divided into time slices equal of the size of the maximum drift window (2.25 ms), we need to generate Trigger Candidates faster than 2.25 ms.

We have benchmarked the performance of the clustering algorithm on Monte Carlo-generated Trigger Primitives, with processing on a single core. We find that for radiological events—the overwhelmingly dominant source of Trigger Primitives—Trigger Candidates are generated about 15 times faster than real time, and thus we can easily keep up with detector data. As more sophistication is added to improve efficiency, we expect the ratio to become less favorable, but there is still plenty of headroom.

3.2 Low-Energy ("Supernova") Candidate Generation

Although in principle the same clustering algorithm can be used for both low-energy (supernova neutrino interaction) Trigger Candidates and high-energy Trigger Candidates, we have developed these independently from one another. There is, however, a possibility that the needs of supernova burst triggering and nominal single-interaction triggering will be different: a series of hundreds of events with energies of 3.5 MeV might constitute a supernova burst for which a high-energy algorithm might be completely mis-optimized. Thus there may not ultimately be a single "unified" algorithm for DUNE.

Nevertheless, generation of Trigger Primitives for single supernova interactions operates in much the same way as clustering does for high-energy events. Groups of adjacent collection-wire hits are grouped together when they are close in time to one another, and then a cut is placed on the total ADC sum of the hits, the minimum number of channels in the cluster, or the width in time of the cluster.

Low efficiencies in triggering on individual events can be tolerated because for a supernova burst, we anticipate hundreds of interactions in the detector. Seeing Trigger Candidates for even a small fraction of these is easily distinguishable from the rate of radiological fakes. Figure 10, made by Alex Booth and Pierre Lasorak, shows the coverage of the galaxy for various single-interaction efficiencies. The orange curve is just the probability of a supernova occuring as a function of distance from Earth (we see the galactic bulge near 10 kpc), and the curves are for different clustering algorithms and the associated rate of fake single-interaction events. We can see that when we aim for a high efficiency (the black curve), the high rate of fakes (19 Hz or so) means we must raise the burst criterion very high to keep to our fake burst rate of 1/month. That higher threshold leads to a poor efficiency for distant supernovae. For the lowest single-interaction efficiency (58%, the yellow curve) we are able to trigger on all supernovae within the galaxy (the yellow curve is indistinguishable from the orange curve).

An important point in this discussion is what was mentioned earlier: if we detect a candidate burst, we take *all* data from a period that is 10 s before the burst is detected, to 20 s afterward. Thus a low single-interaction efficiency does not have an impact on what data we collect for the supernova.

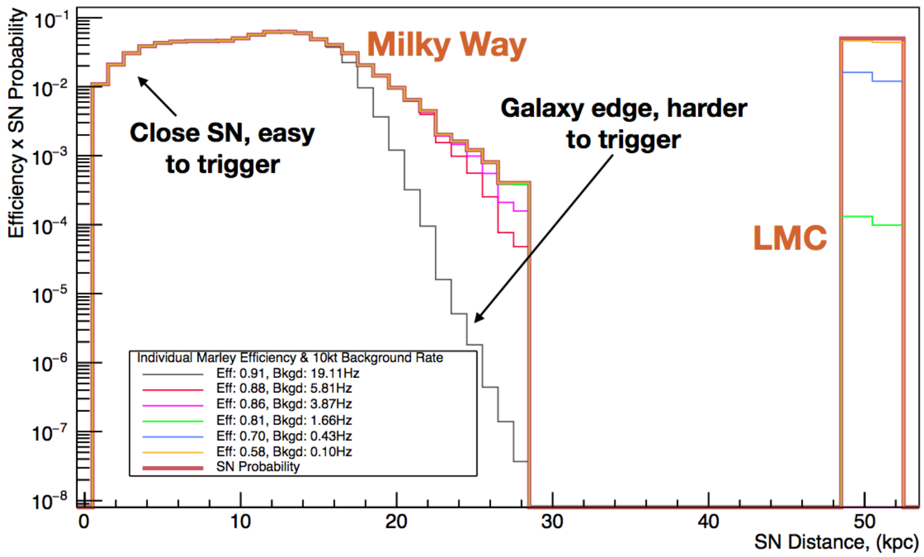

Figure 10: Efficiency of triggering on a galactic supernova as a function of distance. The orange curve shows the probability of a supernova occuring as a function of distance, and the colored curves show different scenarios in which the data selection algorithm is tuned for different efficiencies for a single interactions. We see that when the single-interaction efficiency is low, the rate of backgrounds becomes so low that our efficiency for detecting a burst is very high (near or at unity).

A future path for exploration of SN burst trigger primitives would be to weight them by an estimated event energy. Our first attempts at this, using a machine learning algorithm to reconstruct energy, showed that the intrinsic smearing of the detector (due, for example, to the finite electron lifetime across the drift region) was so large that the weights did not help very much. Work is nevertheless ongoing on this topic.

4 Module Level Trigger

Trigger Candidates are passed downstream to a "Module Level Trigger" that collects such candidates and makes decisions about whether an entire 10 ktonne module should be triggered on a single Trigger Candidate or whether that candidate is part of an ongoing search for supernova bursts. The rate of Trigger Candidates will be low enough that a single server can handle them: the higher-rate low-energy candidates come at a rate of just 0.1 Hz, dominated by neutrons and radon decays.

The Module Level Trigger will maintain a "menu" of currently active triggers, which in addition to high-energy triggers and supernova burst triggers will include calibration source triggers, pulsed and random triggers, Photon Detector System triggers, and prescaled versions of all triggers. Logical combinations of individual triggers can be made as well: for example, a calibration source might provide a tag that is then logically combined with detector activity via Trigger Candidates to form a module-wide trigger.

The Module Level is also responsible for managing partitioning of the detector, and will be able to change its configuration dynamically: for example, if certain APAs are no longer part of a partition, or are known to be non-functioning, the Module Level Trigger will be configured to ignore these, or to include them in a distinct partition.

For all triggers that are not supernova burst triggers, the Module Level Trigger sends trigger commands to the Data Egress Manager that include a timestamp for the trigger and a window duration. As described above, the window duration is nominally twice the duration of the maximum drift time, to ensure that any ambiguities regarding the start and end of a triggered interaction are not relevant. For certain triggers, however, the window may be different: for example, laser events will have a much shorter window because the location and time of the track is known.

Supernova bursts candidates are a special case. Low-energy Trigger Candidates from all active APAs will be accumulated over a 10 second window, and if the number of such low-energy candidates exceeds a pre-determined threshold, a trigger command will be sent via the DEM to tell it to get data from the long-duration event buffer for the 10 s before the time of supernova burst trigger, and will continue to collect data for 20 s after that time. The rate of such burst candidates is targeted to be lower than 1/month, to keep data volumes manageable. It may be possible that such candidates can be reject in a short time by offline analyses, in which case the fake rate can be increased.

With our existing clustering algorithm for low-energy Trigger Candidates described in Section 3, we are able to trigger at greater than 98% efficiency for the entire Milky Way Galaxy. Triggering on the LMC will be more difficult, and studies are underway to understand whether we can do a better job at rejecting low-energy backgrounds and the DUNE radiological task force is working to determine if the sources of radioactivity can be reduced.

The duration of the supernova burst readout window needs to be large enough to deal with the fact that the more distant the supernova, the longer it will take to decide on on a burst candidate. On the other hand, the closer the supernova, the more likely it is that there will be individual supernova neutrino interactions that occur late in time. Our studies have shown that 30 seconds collects all events for supernovae within the galaxy, assuming the supernova models used by DUNE accurately represent the time profile of neutrino emission from real supernovae.

5 External Trigger

The Module Level Trigger is also responsible for sending trigger commands downstream to the global or "External Trigger," which can decide that the trigger from one Module is enough to trigger the entire 40 ktonne DUNE detector. Such a situation might happen, for example, when the accumulated number of low-energy Trigger Candidates in one 10 ktonne module is too few to trigger that module, but collectively the 40 ktonnes together might be above threshold.

The External Trigger will also be responsible for providing information to global supernova neutrino networks, or taking input triggers from other experiments if these are deemed interesting and trustworthy.

6 High-Level Trigger

Once data is read out from the front end through the Data Egress Manager, it goes to the Event Builder which combines it with information from the data selection system, such as the trigger type. Once the data is built, it will pass through the High-Level Trigger (HLT) before being written to disk. The HLT can include sophisticated algorithms that employ techniques like machine learning to either tag events as being part of a particular class (proton decay rather than beam events, for example) for later analysis, or it can remove events that have passed the earlier, much simpler, stages of data selection. It may also be used to reduce the size of the data set, by removing APAs that are empty of any useful information for physics. Unlike the earlier stages of data selection in the single-phase detector, the HLT will be able to use the induction wire information in its selection algorithms, and this will help in either filtering or reducing the data set after event building.

For triggering on low-energy events such as solar neutrinos, the HLT could prove very valuable if a way could be found to eliminate the dominant source of background, which are neutrons. Such events will look very different in detail than solar neutrino events, but at the earlier stages of data selection these details are much harder to use.