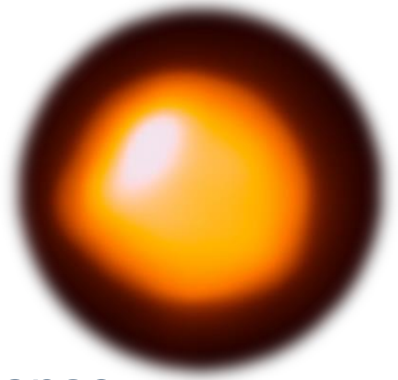
Supernova triggering at DUNE from machine-learning based clustering

Dennis Lindebaum
On behalf of the DUNE Collaboration

10.04.24

Betelgeuse

Betelgeuse:
ALMA
(ESO/NAOJ/NRAO)/
E. O'Gorman/P. Kervella

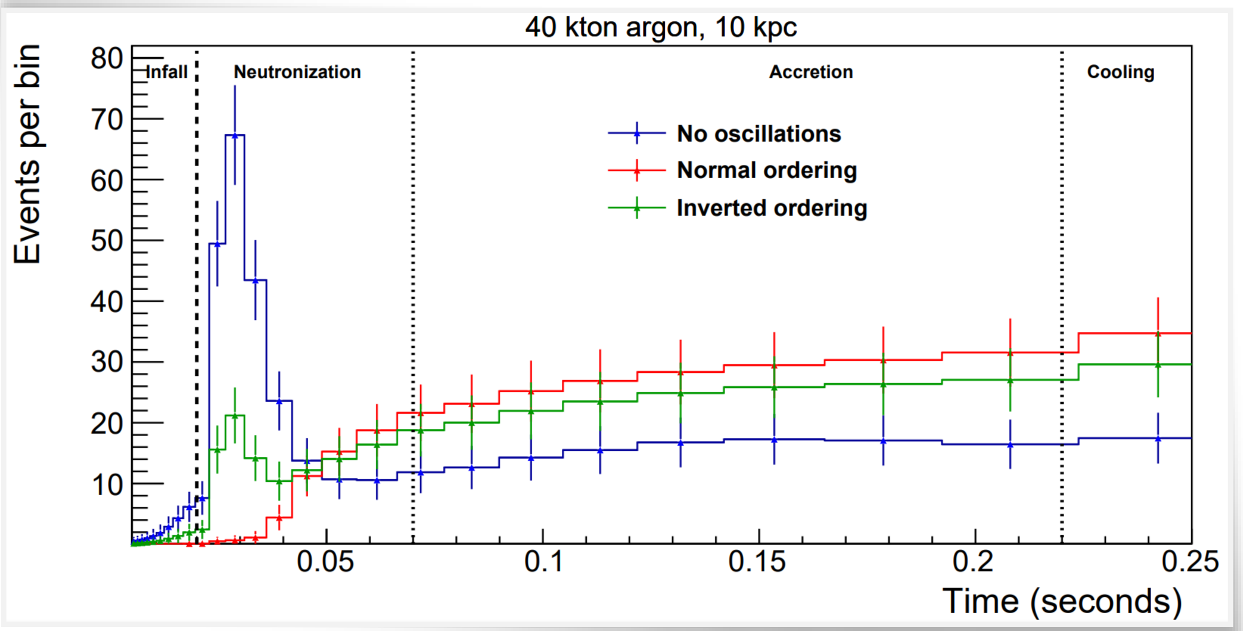


- Last year, a study [arXiv:2305.09732] used brightness oscillations to predict:
  - "After carbon is exhausted (likely in less than ~ 300 years) in the core, a core-collapse leading to a supernova explosion is expected in a few tens of years."
- Not everyone agree with the conclusions [10.3847/2515-5172/acdb7a].
- 99% of SN energy is carried by neutrinos of *O*(10MeV).
  - Betelgeuse ~500ly (~150 pc) away
  - More than 10^6 neutrinos expected in each detector module of DUNE.
- Only one supernova burst (SN1987A) has been recorded to date, from 51kpc away.
  - 25 neutrinos recorded by Kamikande, IMB, and Baksan.

SN1987A remnant captured by JWST

Supernova Physics

- Astrophysical phenomena:
  - Early warning for optical telescopes (neutrinos escape a few hours before the first photons).
  - Details of supernova models.
- Neutrino physics:
  - Neutrino oscillation parameters, with matter effects of high-density supernova matter.
  - Mass ordering leaves a strong imprint on the resulting flavour spectrum.



Expected number of neutrino interactions in DUNE from a 10kpc supernova burst. From [arXiv:1804.01877]

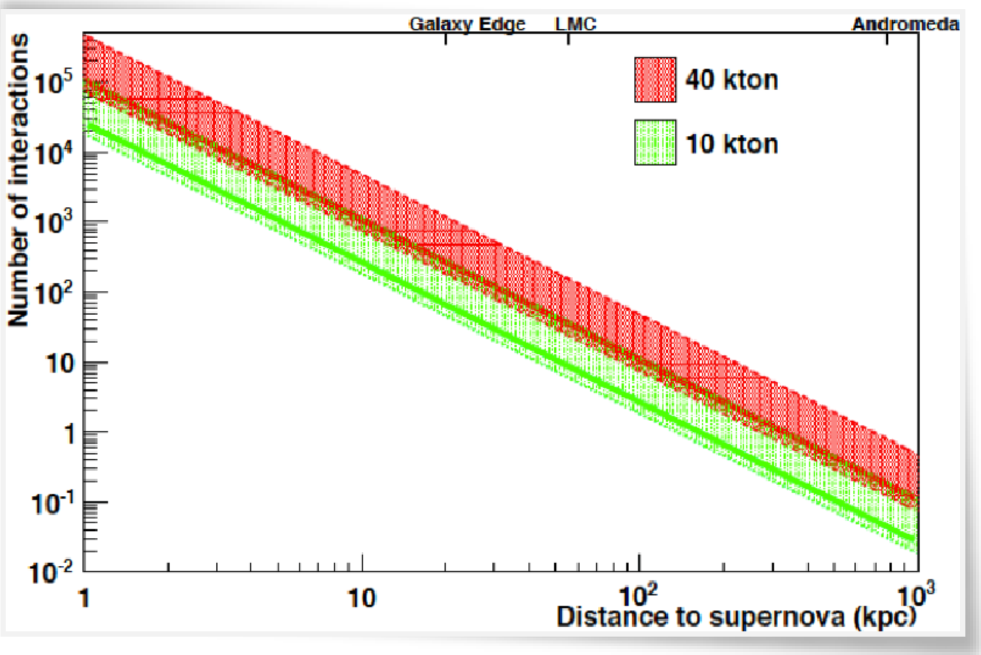
Betelgeuse or Bust?

- Even in the best case, at most 10% chance of Betelgeuse going supernova during lifetime of DUNE.

BUT

- Galactic supernova are expected 1-3 times per century.
- The edge of the Milky Way is ~25kpc away.
- About 60 neutrino interactions expected per module for a galactic supernova.

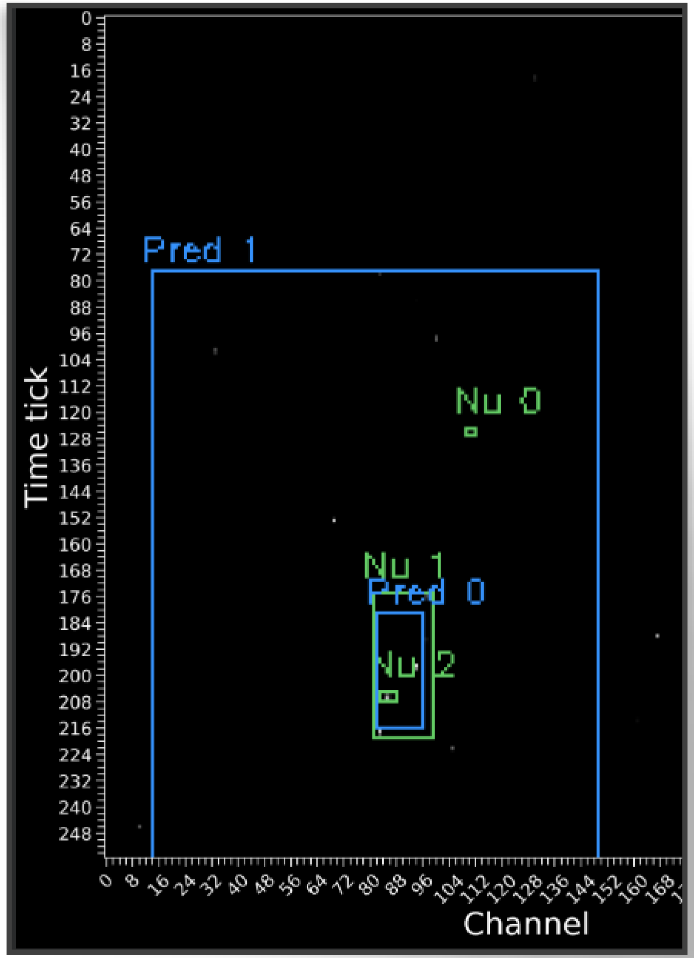
- How many neutrino interactions must be incident in a detector module to create a trigger?



Expected number of neutrino interactions in DUNE from a supernova burst. From arXiv:2011.06969

Low energy clustering

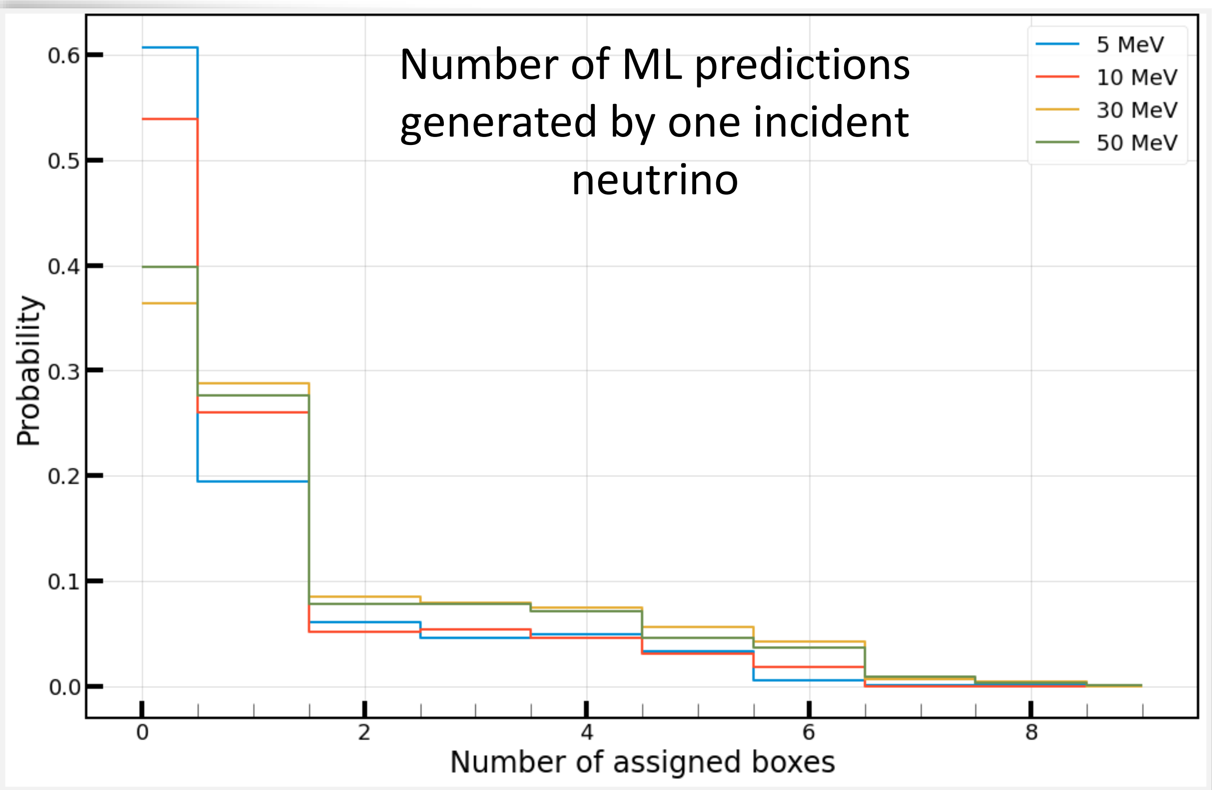
- We have performed an example study on the performance which may be obtained by machine learning methods.
- Network creates boxes where it predicts a neutrino is present.
- Many classical clustering algorithms produce “yes/no” type results (without multiplicity).
  - E.g. DBScan takes a set of points, an gives each point a cluster label
- We want a like-for-like method to compare ML results to classical clustering methods.
  - Produce trigger chance from clustering distribution and incident neutrino count.



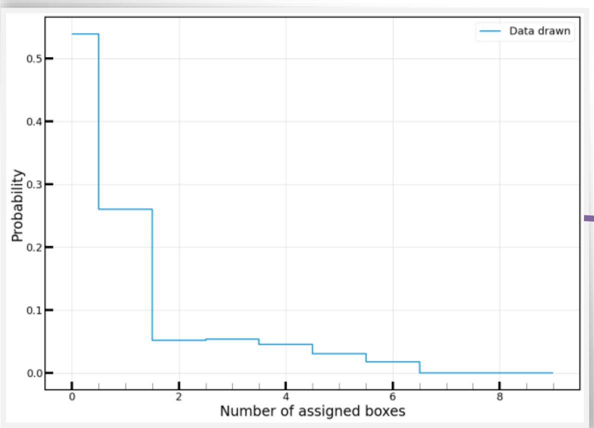
Example predictions produced by a neural network in blue, with true neutrino interactions in green.

Multiplicity

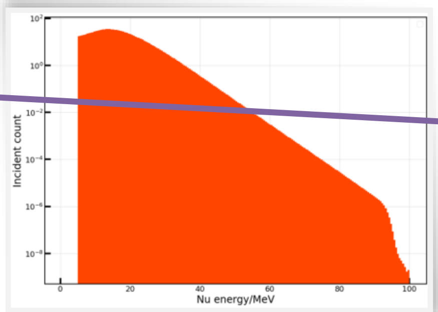
- Standard clustering algorithms normally produce binomial clustering efficiencies.
  - Each data point is assigned a cluster label.
- The neural network can produce multiple predictions overlapping the same region of space.
- Results in a multiplicity of predictions per true incident neutrino.
- Probability may be a function of neutrino energy.



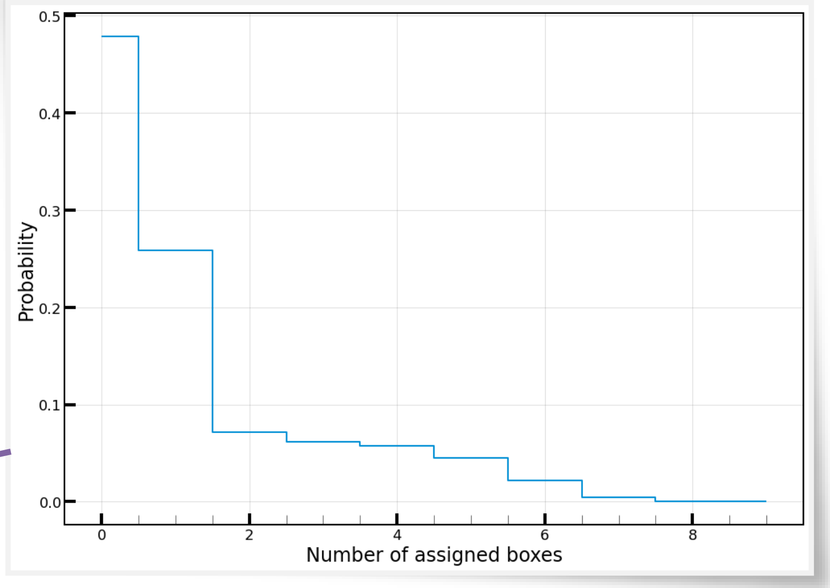
Supernova detection probability



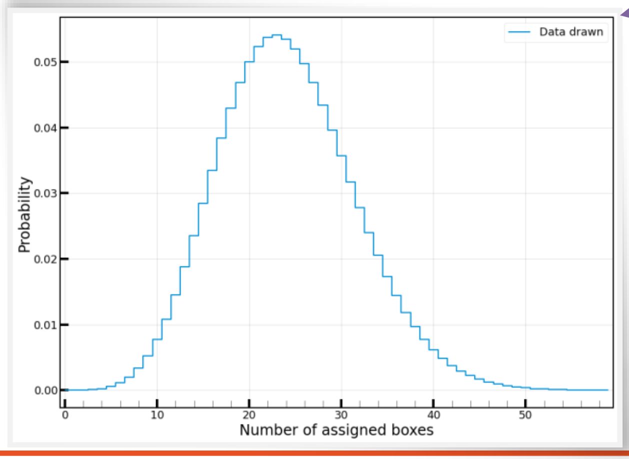
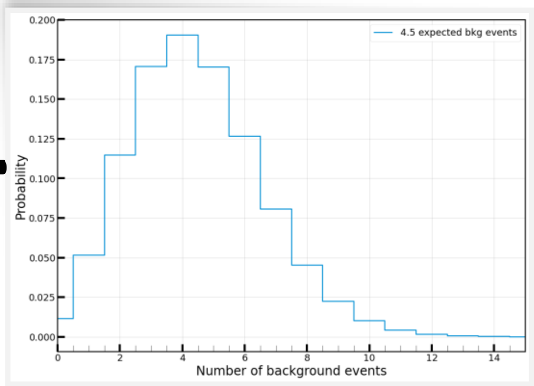
Detection multiplicity for a neutrino of a particular energy



Weight by energy distribution from a supernova sample (model dependent)



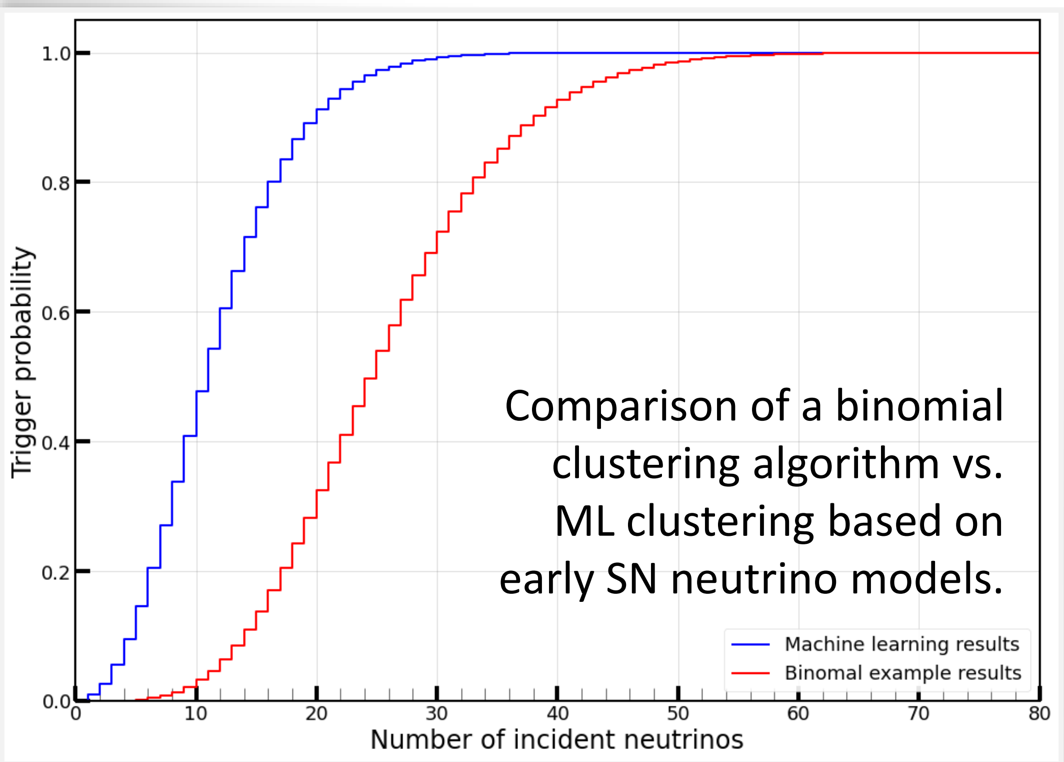
N neutrinos drawn at random from energy weighted detections

P( + ) ≥ threshold

Poisson count from random background events in 10s

Supernova detection probability

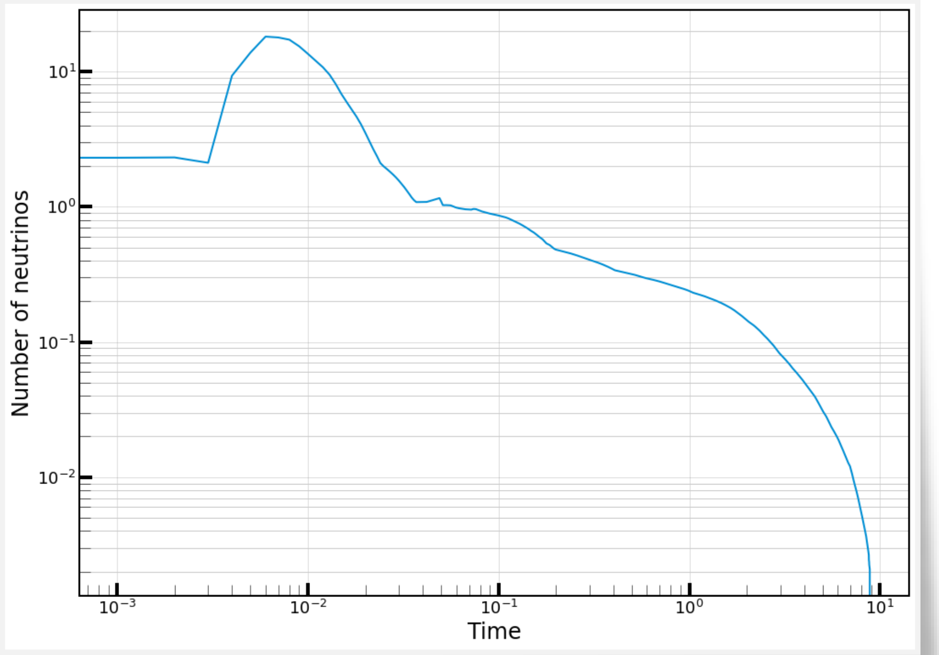
- Probability of creating a trigger in a window is a function of the number of neutrino detections in the window and the background count.
- Algorithms and parameters obtained from early low-background SN neutrino models in DUNE.
- Classical clustering:
  - 33% binomial efficiency
  - 0.14 Hz background rate
- Machine learning:
  - 1.2 average multiplicity
  - 0.45 Hz background rate



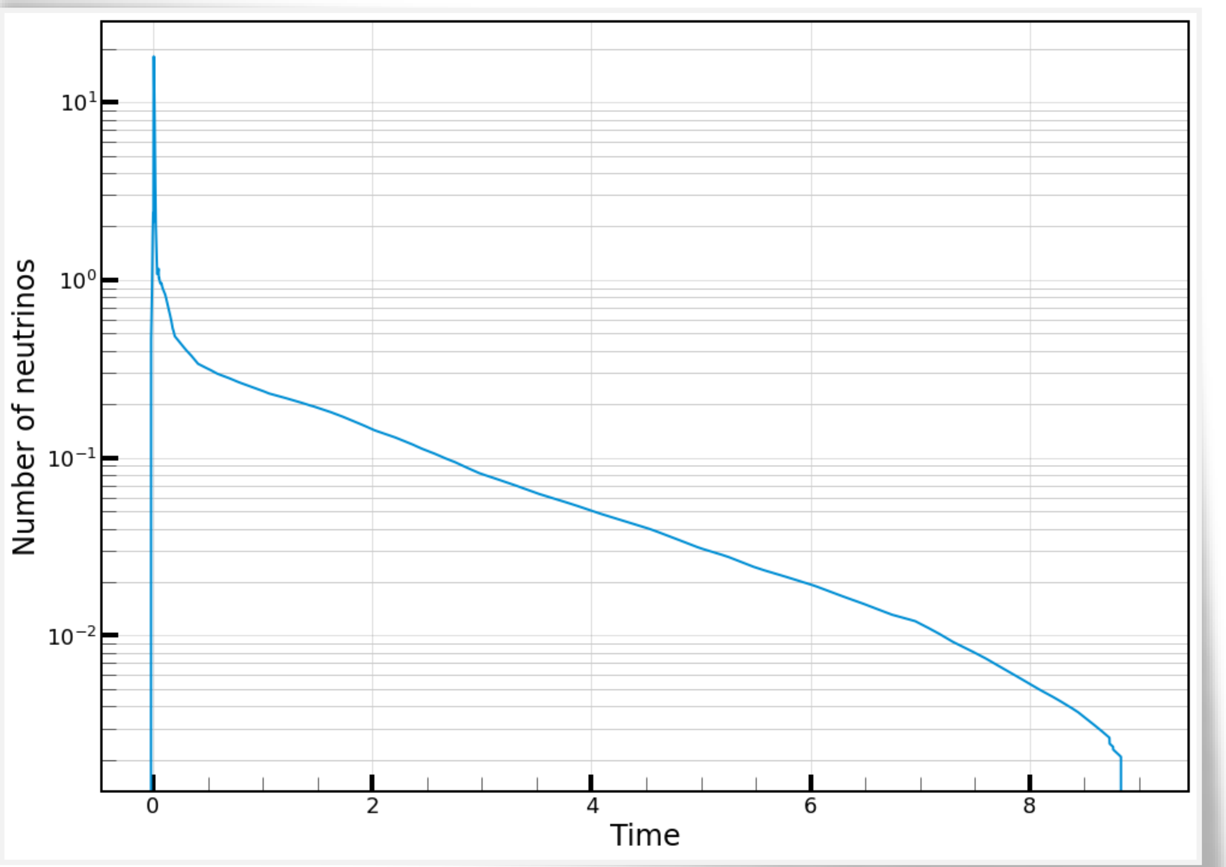
Comparison of a binomial clustering algorithm vs. ML clustering based on early SN neutrino models.

Unknown SN incidence time

- So far, we have assumed every incident neutrino falls exactly within one 10s detection window.
- In reality: neutrinos may be spread over multiple windows.



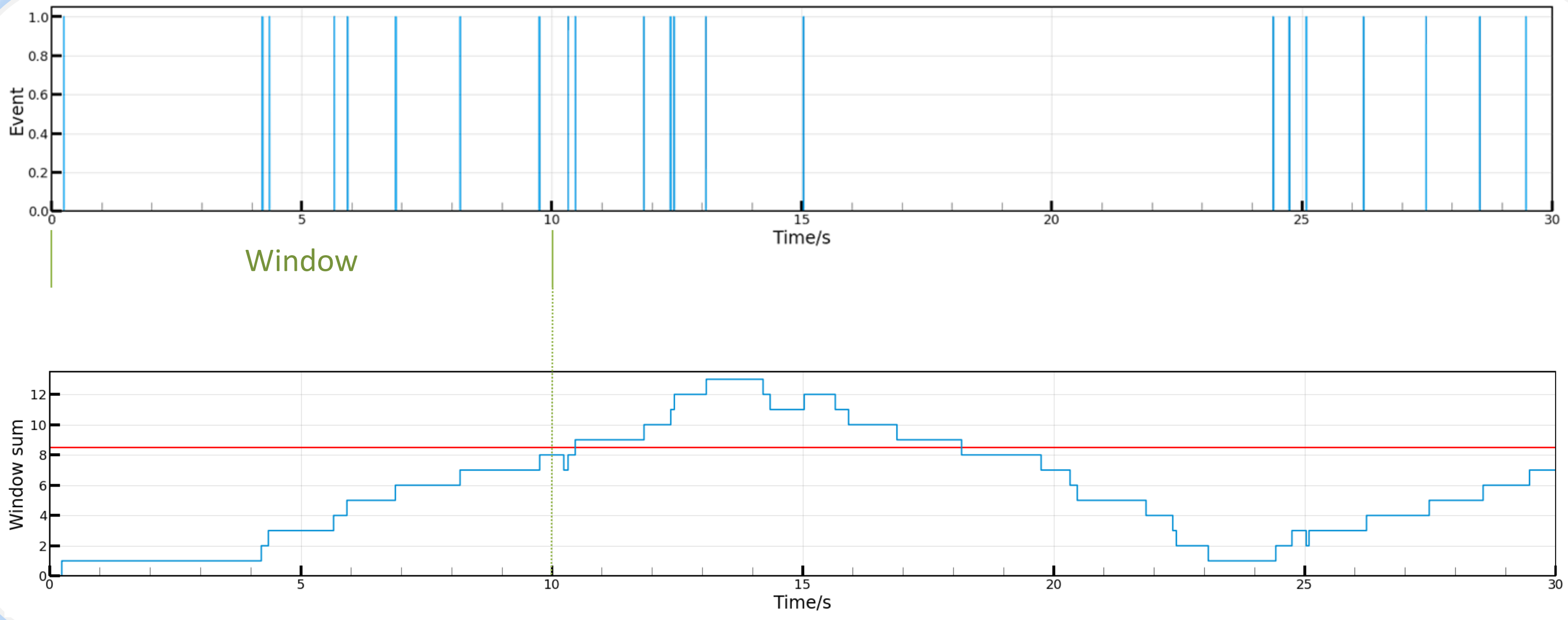
Example SN time spectrum (log-log scale)



Example SN sample time spectrum

Sliding window

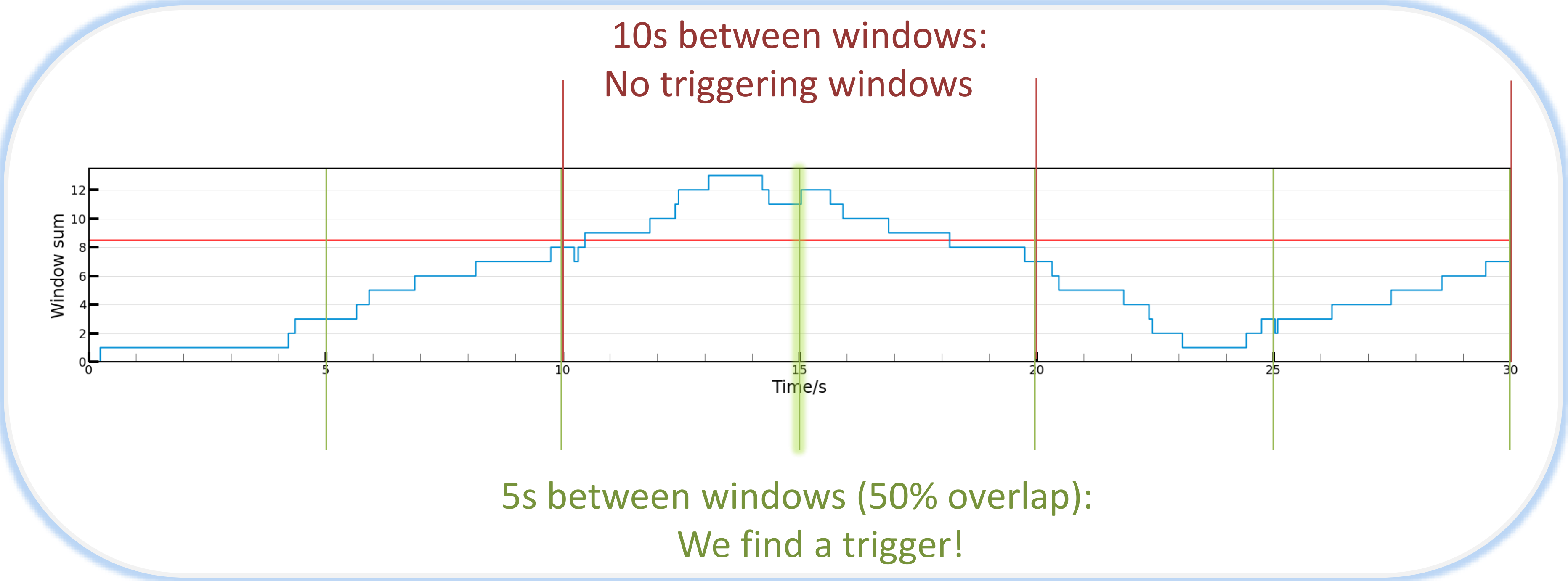
- Background events are modelled with a Poisson – the probability of an event arriving at time *t* is uniform



- We generate windows by counting events within some time frame (10s).

Sliding window

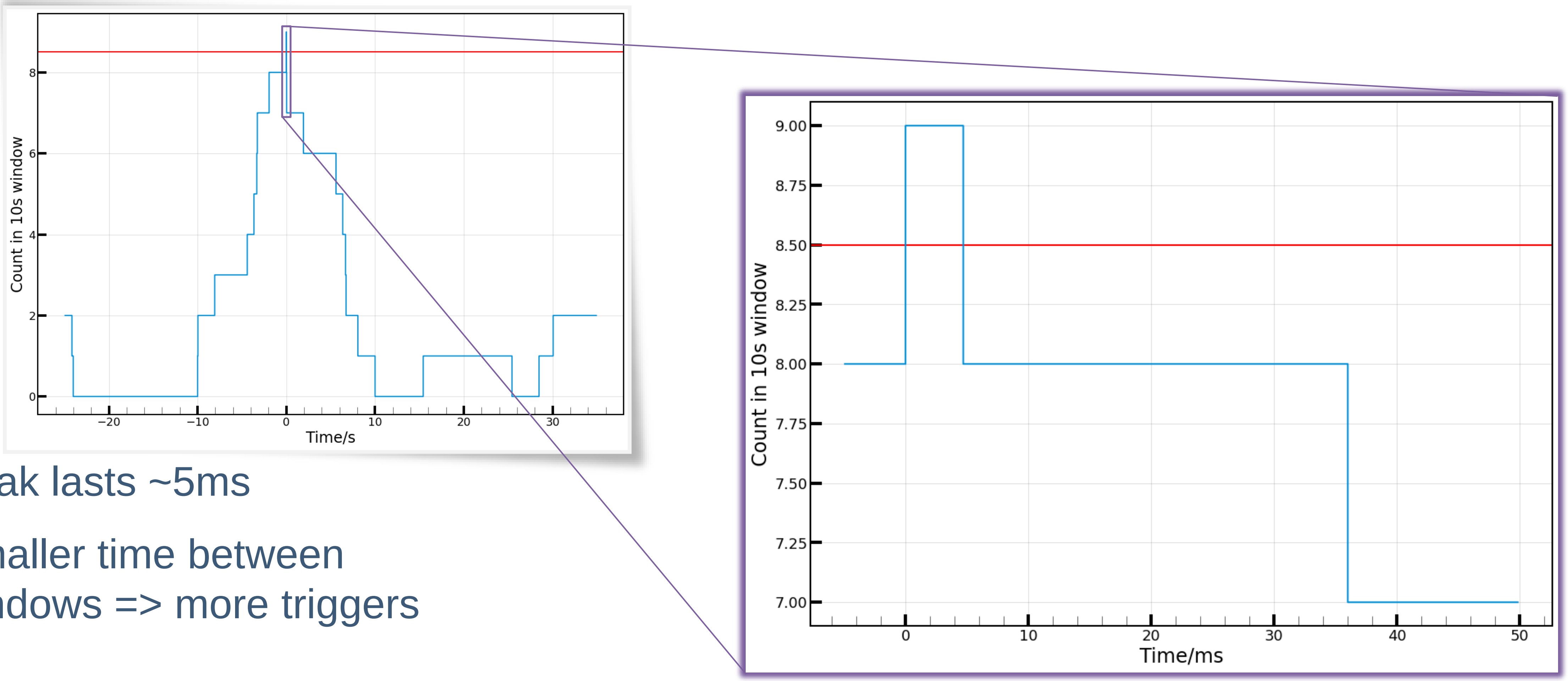
- Choosing to slide the window is equivalent to selecting a sample rate on the window sum plot.
- Define “sliding time” as the time between subsequent windows.



- Increased sliding also increases the number of triggers.

Sliding window – data driven

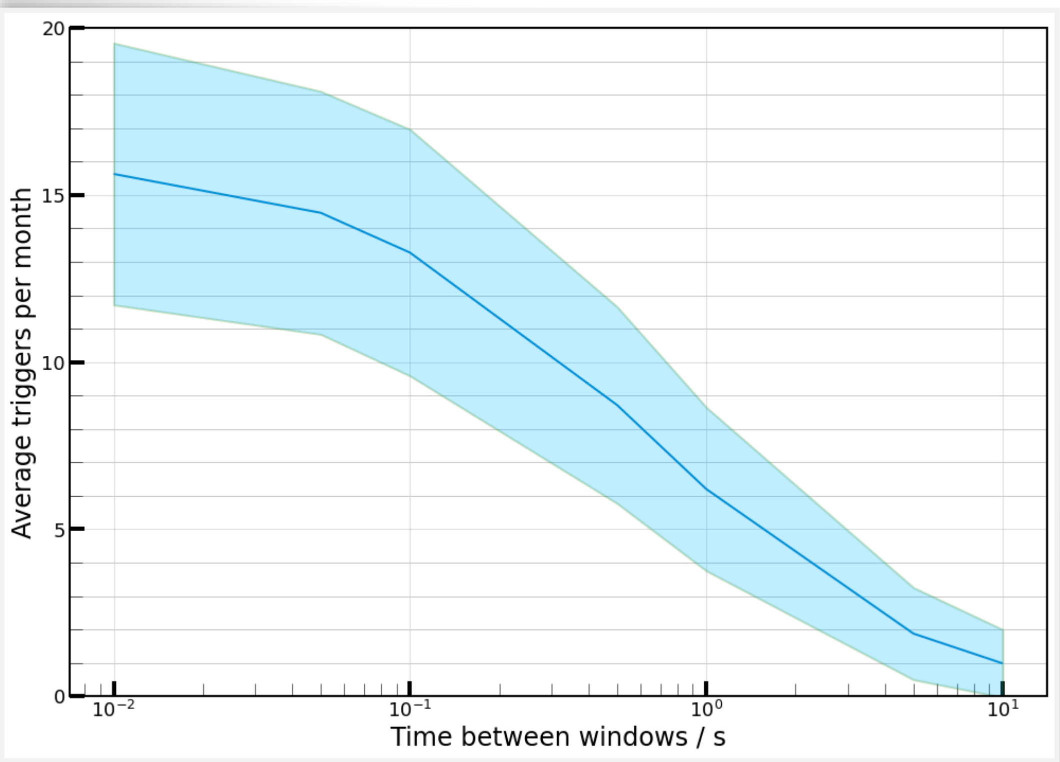
- Running with data driven parameters (0.1Hz fake rate, 9 count threshold)



- Peak lasts ~5ms
- Smaller time between windows => more triggers

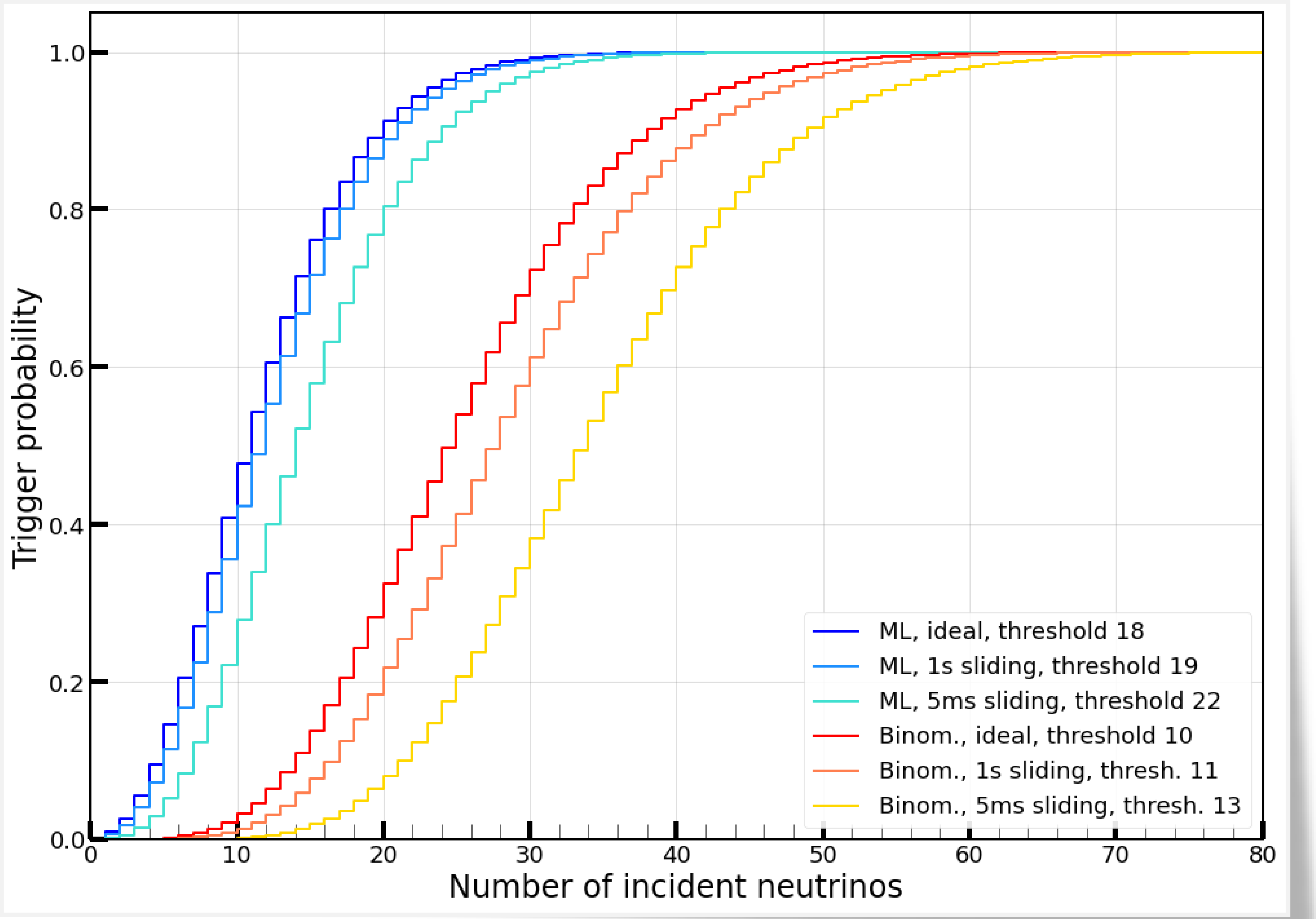
Sliding window – data driven

- Monte-Carlo simulation performed was used to characterise the trigger rate as a function of window sliding.
- A trigger is generated if the number of clusters found in a window exceeds some threshold.
- Reducing the time between window measurements with fixed threshold increases the number of triggers.
- To compensate – increase the threshold required to trigger.



Average number of triggers seen per month as a function of time interval between windows. Values calculated from 600 months of data.

Sliding window



Supernova triggering chance for various triggering options

Summary

- We can generate trigger chances from arbitrary clustering distributions.
  - Compare machine learning vs. classical methods
- Supernova detection probabilities need to include the effect of unknown arrival times.
- Sliding windows can catch supernova bursts at unknown times, with a small efficiency loss.
  - Sliding windows also ensure we capture a supernova burst promptly.
- Sliding windows require trigger threshold adjustment
- Machine learning methods have the potential to improve trigger chances, but can have peculiar “multiplicity” effects.