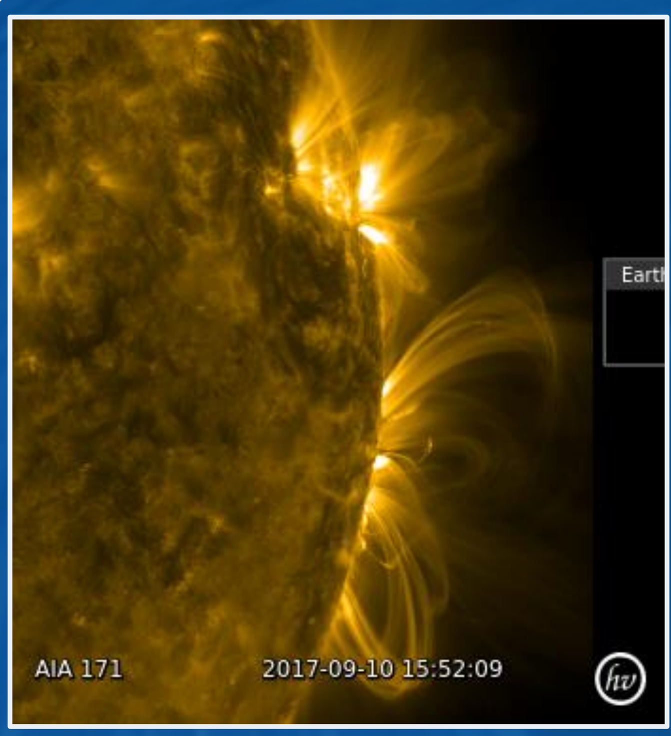
A GCN to search for solar-flare neutrinos with KM3NeT

Neutrino Production



Solar flares are highly energetic explosions that occur in the solar atmosphere.

They are multi-messenger transient sources, and they are known to be the site of particle acceleration.

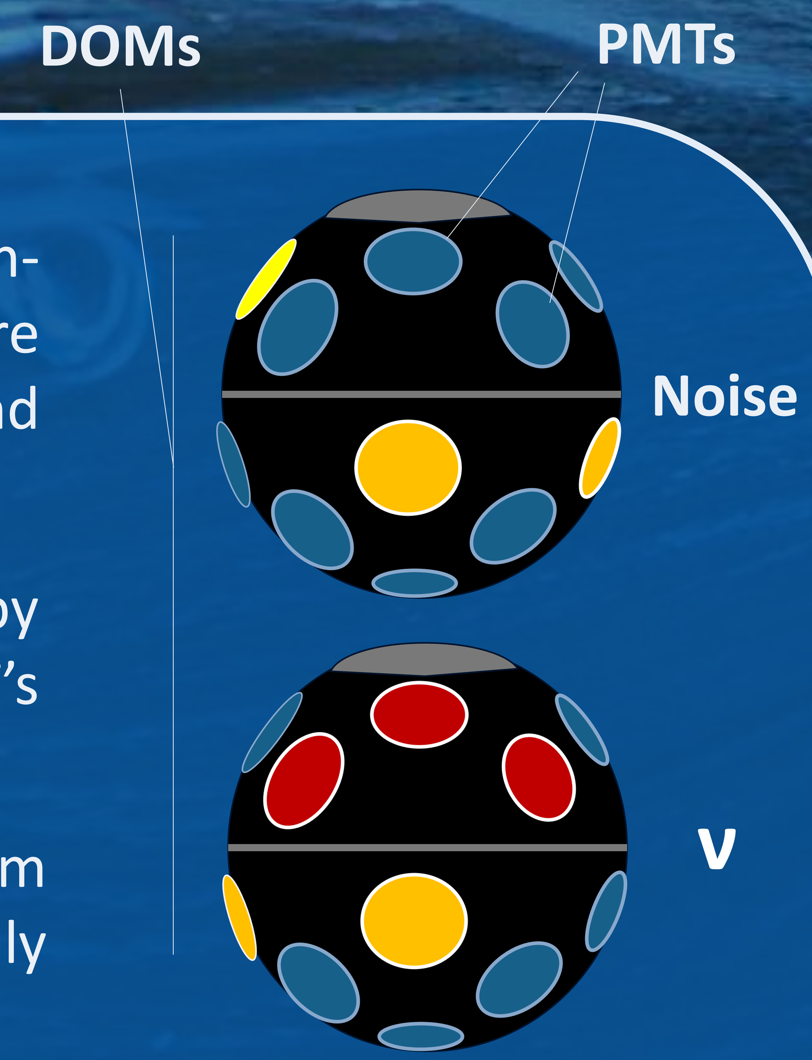
Although the physical processes involved are not fully understood.

We can still infer the neutrino production by looking in the γ-ray spectrum for pion decay emissions.

Solar-flare neutrinos are expected to range from

MeV to a few GeV.

Detection



KM3NeT is optimised to search for high-energy neutrinos, while GeV neutrinos are difficult to distinguish from atmospheric and environmental events.

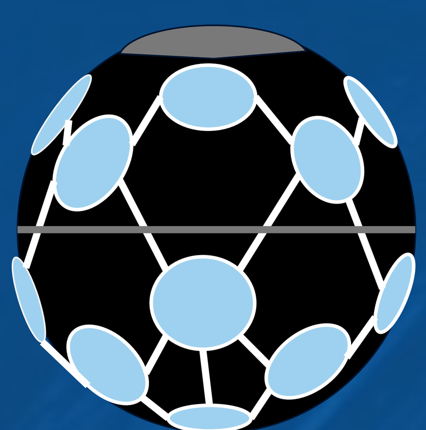
We aim to detect sub-GeV neutrinos by exploiting the multi-PMT design of KM3NeT's DOMs.

To discriminate low-energy neutrinos from background noise, we look for causally connected hits on the same DOM.

By comparing the number of low-energy events measured in coincidence with solar flares and in the absence of them, we will be able to probe the solar-flare neutrino flux, and to constrain production models.

Multi-PMT event selection would be easily adapted to the other next-gen neutrino telescopes, such as IceCube-Gen2.

Dataset



We build our dataset by grouping into events coincident hits (**Δt < 30 ns**), which occur on the same DOM.

We apply hard cuts on the hits (**ToT > 5 ns**) and on the events (**# of hits > 2**) to get rid of a large portion of the background.

We encode the events to graphs with 31 nodes, representing the PMTs on a DOM, edges are drawn based on PMT density.

We store the ToT of each hit as node attributes, while the standard deviation of charge distribution over time is used as a graph attribute.

We build the background events from data, and signal events from neutrino simulation, and we use them to train an ANN for graph classification.

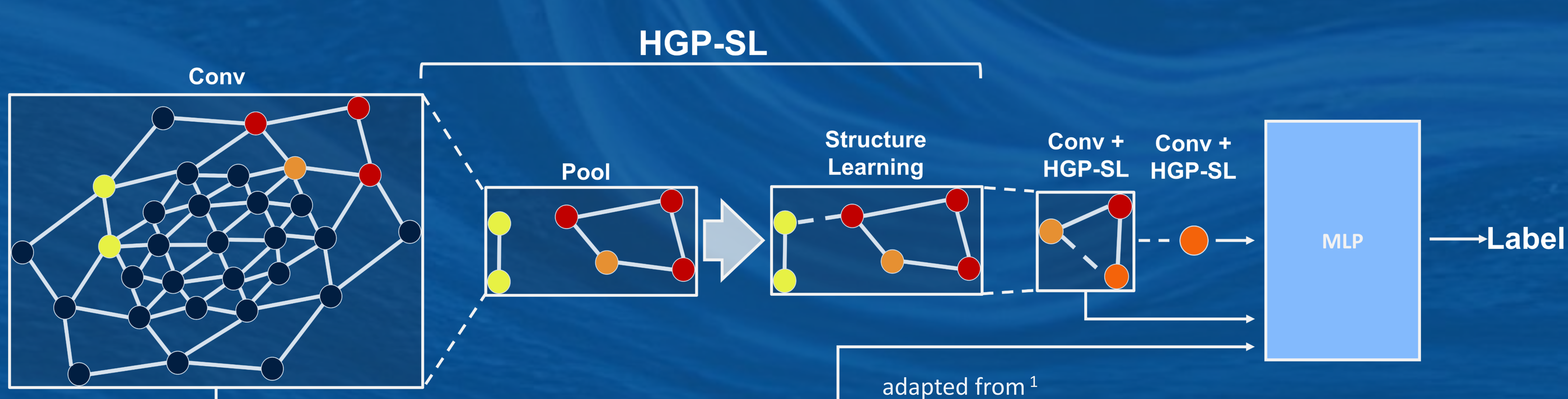
Classification

The classification algorithm is a graph convolutional network which implements the HGP-SL operator [1].

HGP-SL is used between the convolutional layers to perform pooling and structure learning.

The pooling operation is done by computing an information score, this allows us to select the subset of nodes that cannot be well reconstructed from their neighbours.

Structure learning is used to reproduce the original graph structure in the pooled representation. It calculates a similarity score between nodes, and it is trained to assign high similarity to directly connected nodes.



Disconnected nodes with high similarity are connected before the next convolutional layer.

The output of the convolutional layers is fed into an MLP to obtain the label prediction.

[1] "Hierarchical Graph Pooling with Structure Learning" Z. Zhang, J. Bu, M. Ester et al. (2019)

GraphNeT III
May 1 – May 4, 2023
Sandvig, Denmark

Jonathan Mauro
Jonathan.mauro@uclouvain.be
CP3, UCLouvain

UCLouvain