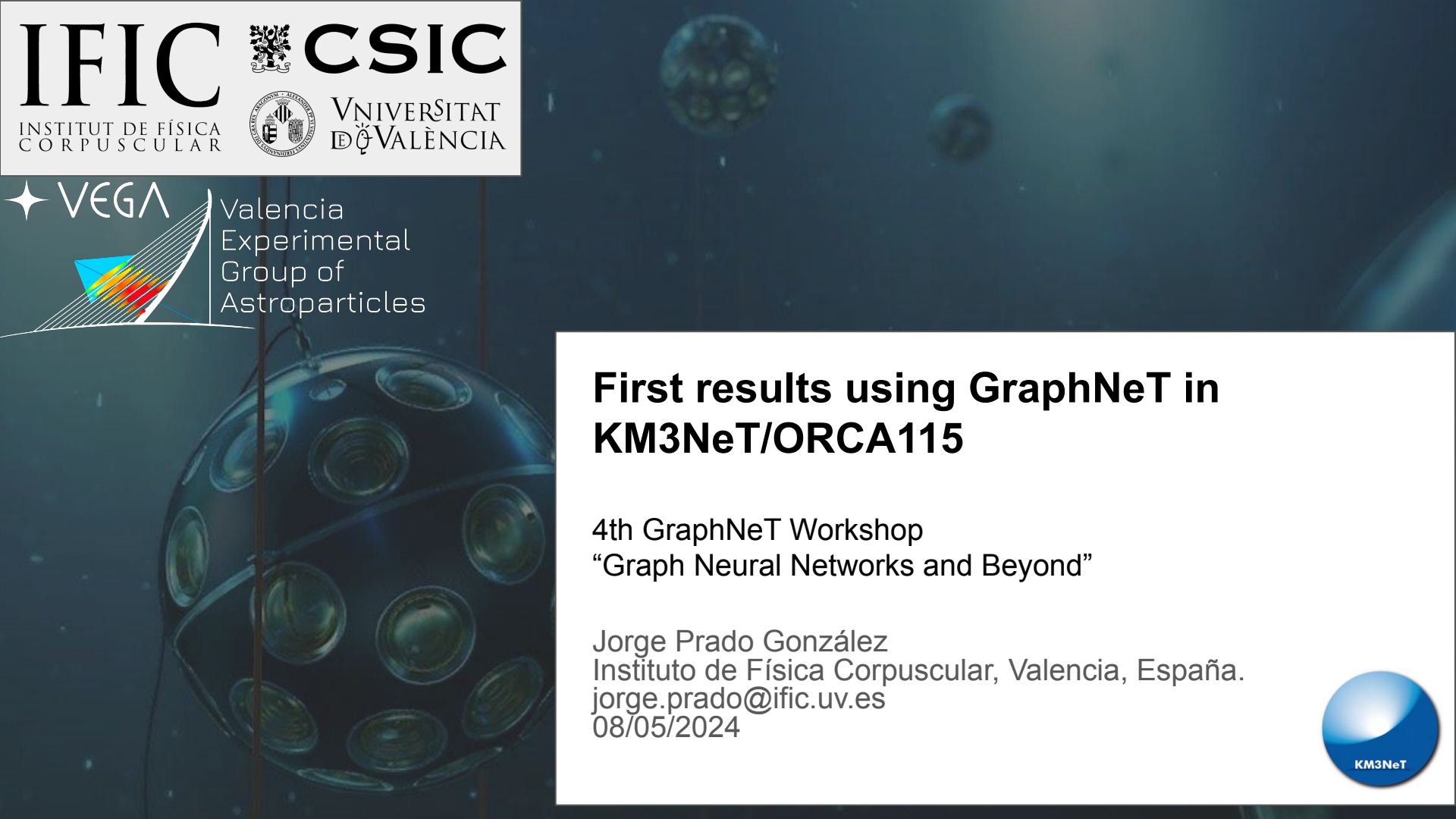


IFIC INSTITUT DE FÍSICA CORPUSCULAR
CSIC
VNIVERSITAT DE VALÈNCIA

VEGA Valencia Experimental Group of Astroparticles

First results using GraphNeT in KM3NeT/ORCA115

4th GraphNeT Workshop
“Graph Neural Networks and Beyond”

Jorge Prado González
Instituto de Física Corpuscular, Valencia, España.
jorge.prado@ific.uv.es
08/05/2024

KM3NeT

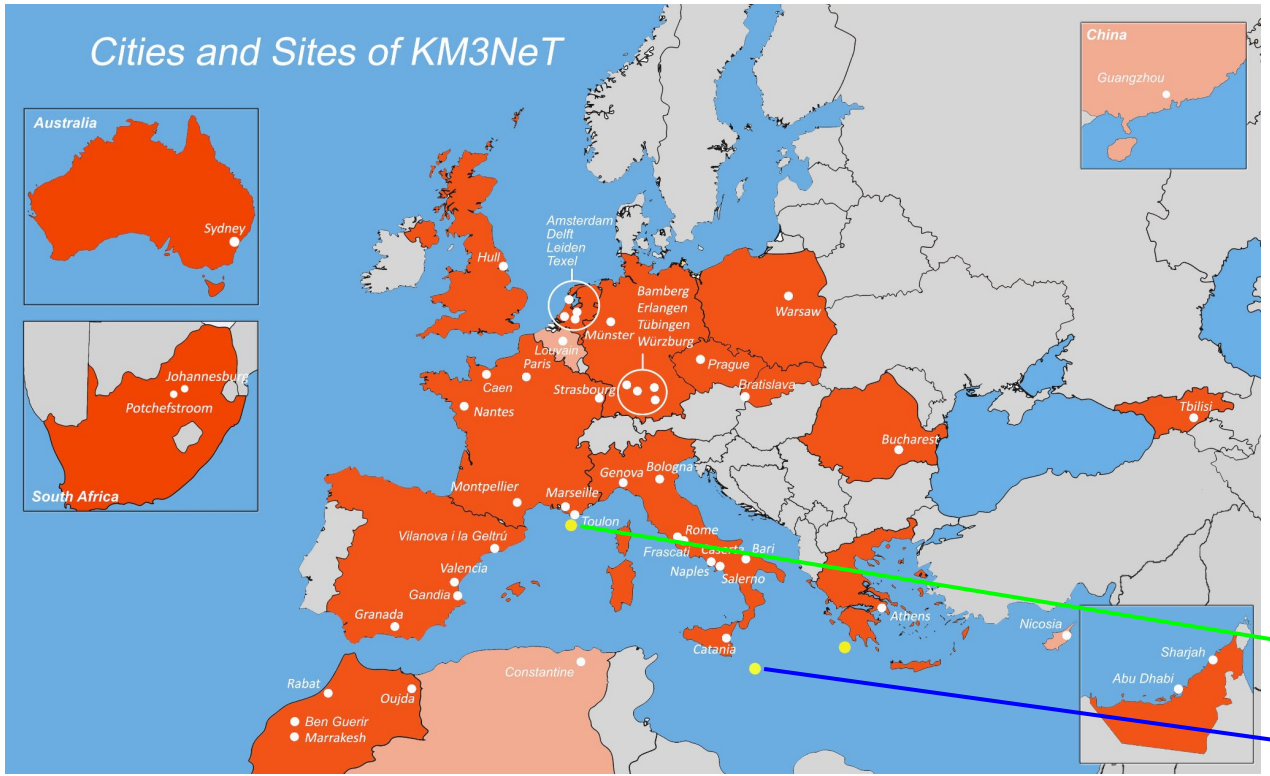
Index

- KM3NeT/ARCA and KM3NeT/ORCA. How do events look like in KM3NeT?
- Brief description of GraphNet KM3NeT extractors.
- Discussion of preliminary results in ORCA 115.

Index

- KM3NeT/ARCA and KM3NeT/ORCA. How do events look in KM3NeT?
- Brief description of GraphNet KM3NeT extractors.
- Discussion of preliminary results in ORCA 115.

KM3NeT



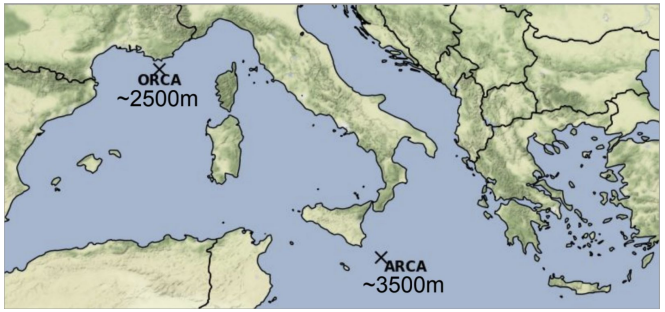
- International collaboration with ~250 members, 45 partner institutes over 14 countries.
- Two detectors in different sites: **KM3NeT/ORCA** and **KM3NeT/ARCA:**
  - ➢ Same technology
  - ➢ Same data processing
  - ➢ Same software and common dataformats.

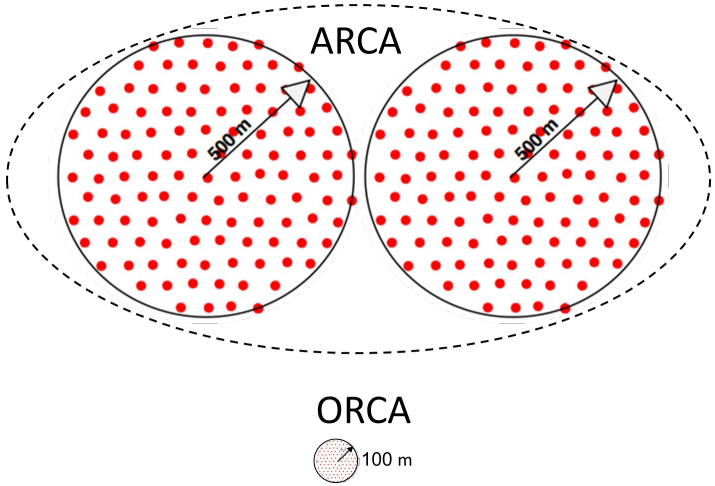
KM3NeT/ORCA

KM3NeT/ARCA

KM3NeT - ARCA and ORCA

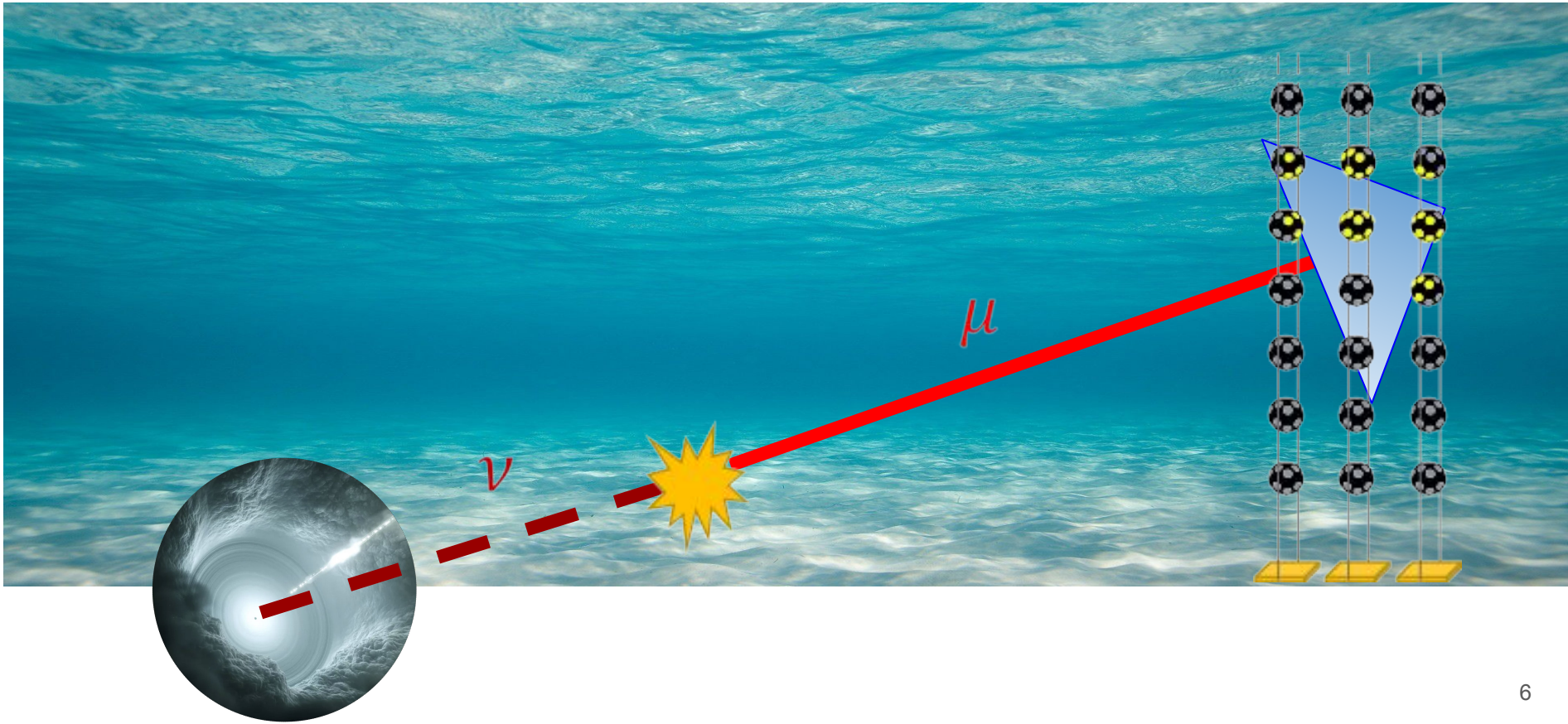
Two detectors. One Technology





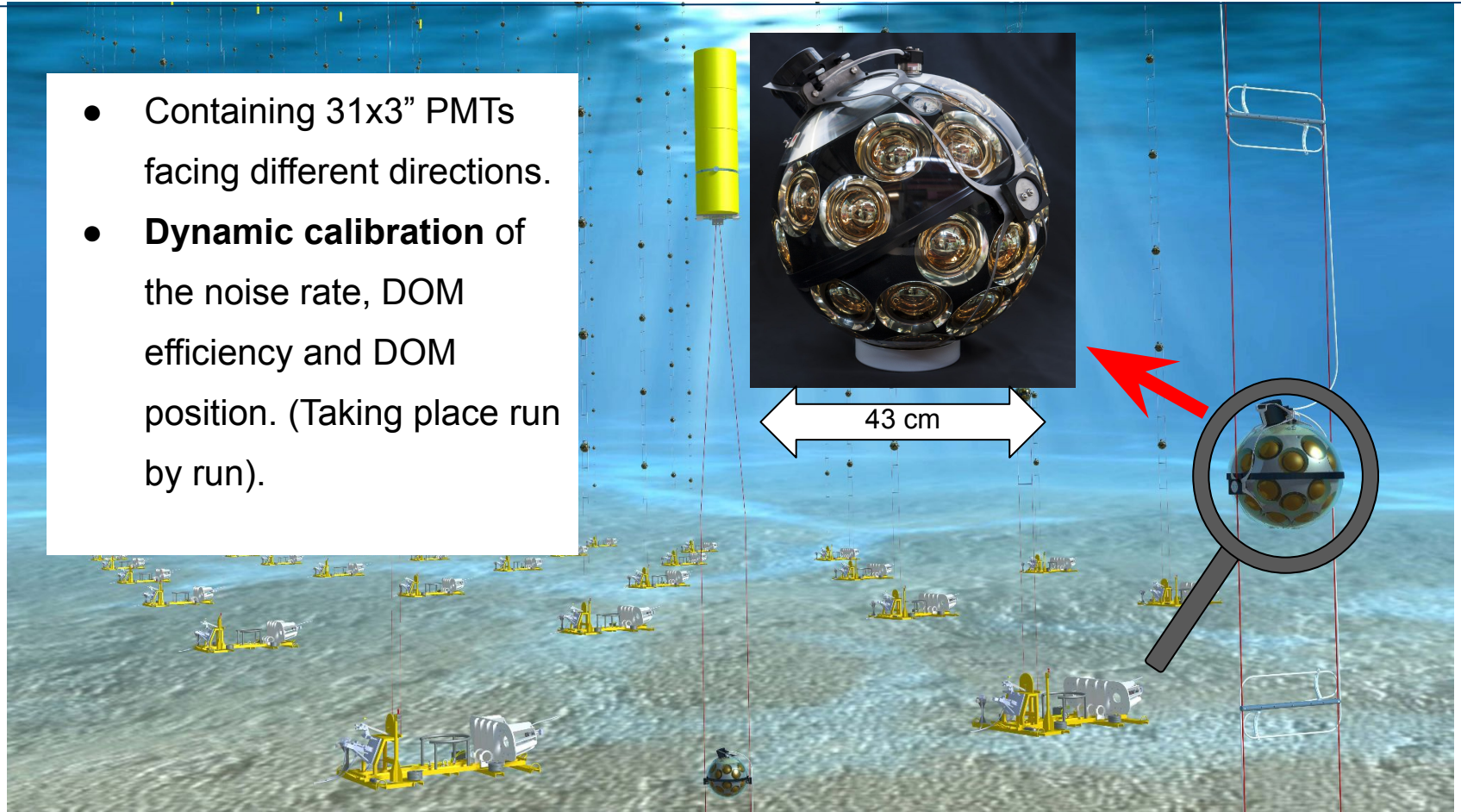
- **KM3NeT/ARCA:**
  - ➢ High Energies (sub-TeV to few PeV)
  - ➢ Astrophysical studies (mainly)
  - ➢ Full ARCA with 230 Detection Units (DU), 18 DOMs per DU, 128k PMTs in total. Currently 28 DUs deployed.
  - ➢ DOMs with ~36m vertical and DUs with ~90m horizontal separation.
- **KM3NeT/ORCA:**
  - ➢ Low energies (~few GeV)
  - ➢ Neutrino fundamental properties studies (mainly)
  - ➢ Full ORCA with 115 DUs, with 18 DOMs per DU, 64k PMTs in total. Currently 19 DUs deployed.
  - ➢ DOMs with ~9m vertical and DUs with ~20m horizontal separation.

KM3NeT - Detection principle

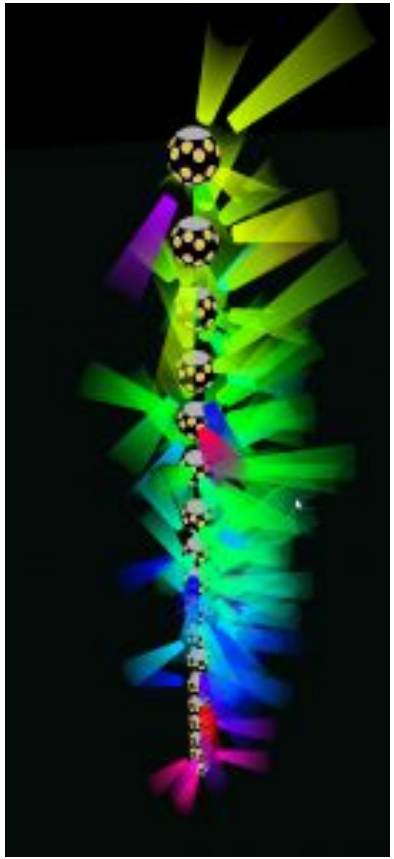


KM3NeT Digital Optical Modules (DOMs)

- Containing 31x3” PMTs facing different directions.
- **Dynamic calibration** of the noise rate, DOM efficiency and DOM position. (Taking place run by run).



Information in a pulse



- **X, Y and Z coordinates of the hit:** Position of the PMT that captured the light.
- **Orientation of the PMTs.**
- **Time of the hit.**
- **Time over Threshold (ToT):** The time in which the pulse remains over a 0.3 p.e. threshold.
- **Triggers:** Algorithms to quickly filter background from signal. Not intrinsic of the pulse but also depends on neighbouring pulses.

https://youtu.be/B7cbc7OHCbM?list=PLL9OR_-tW5qNsDSwmKUQtr3eRk4BypbGP

Index

- KM3NeT/ARCA and KM3NeT/ORCA. How do events look in KM3NeT?
- Brief description of GraphNet KM3NeT extractors.
- Discussion of preliminary results in ORCA 115.

New pipeline to feed GraphNeT

- GraphNeT needs **SQLite** or **Parquet** format databases.
- GraphNeT had a data-converter class to convert a data_file.format to data_file.db or data_file.parquet.
  - X Not working. It was written in a non-generic way only suitable for .i3 files (IceCube)
- New generalization of the code to create a generic DataConverter class.

  https://github.com/graphnet-team/graphnet/pull/659

New pipeline to feed GraphNeT

Extractors:

Extract information from a file, frame or other data structure.

TruthExtractor: to extract the true particle information if it is mc, PulseExtractor: extracts information from the different pulses (charge collected, time etc…)

Readers:

Class to collect and run all the extractors, collecting all the information into a dictionary of dataframes (or a list of dictionaries).

{“true”: pd.Dataframe(true), “pulse”: pd.Dataframe(pulse)}

Need to create a Unique ID per event and tag every variable.

Writers:

Generic and common to all experiments. Takes the output of the Reader and converts it into SQLite or Parquet databases. Ready to train!

New pipeline to feed GraphNeT

- DataConverter class easy to call with very few lines of code:

```
if __name__ == '__main__':
    index_column = 'your_index'
    input_dir = 'input_folder_with_eg_root_files'

    converter = DataConverter(file_reader = KM3NeTROOTReader(),
                              save_method=SQLiteWriter(),
                              extractors = [KM3NeTROOTTruthExtractor(name = "truth"),
                                            KM3NeTROOTTriggPulseExtractor(name = "trigg_pulse_map")],
                              outdir = "output_folder_with_sqlite_files",
                              num_workers = 1)
    converter(input_dir=input_dir)
    converter.merge_files()
```

New pipeline to feed GraphNeT

KM3NeTReader

```python
import km3io as ki


class KM3NeTROOTReader(GraphNeTFileReader):
    """Class for reading KM3NeTROOT files."""

    _accepted_file_extensions = [".root"]
    _accepted_extractors = [
        KM3NeTROOTTruthExtractor,
        KM3NeTROOTPulseExtractor,
        KM3NeTROOTTriggPulseExtractor,
    ]

    def __call__(
        self, file_path: Union[str]
    ) -> Dict[str, Union[Dict[Any, Any], Any]]:
        """Open and apply extractors to a single root file.

        Args:
            file_path: The path to the file to be read.

        Returns:
            data in a list of ordered dataframes with a unique ID.
        """
        file = ki.OfflineReader(file_path)
        if len(file.mc_trks[:, 0]) > 0:
            data = {}
            for extractor in self._extractors:
                data[extractor._extractor_name] = extractor(
                    file
                )  # extractor returns dataframe

            return data
```

KM3NeTExtractors

```python
primaries = file.mc_trks[:, 0]
unique_id = create_unique_id(
    np.array(primaries.pdgid),
    np.array(file.run_id),
    np.array(file.frame_index),
    np.array(file.trigger_counter),
)  # creates the unique_id

hits = file.hits
keys_to_extract = [
    "t",
    "pos_x",
    "pos_y",
    "pos_z",
    "dir_x",
    "dir_y",
    "dir_z",
    "tot",
    "trig",
]
```

```python
dict_truth = {
    "pdgid": np.array(primaries.pdgid),
    "vrx_x": np.array(primaries.pos_x),
    "vrx_y": np.array(primaries.pos_y),
    "vrx_z": np.array(primaries.pos_z),
    "zenith": zen_truth,
    "azimuth": az_truth,
    "part_dir_x": part_dir_x,
    "part_dir_y": part_dir_y,
    "part_dir_z": part_dir_z,
    "Energy": np.array(primaries.E),
    "Bj_x": np.array(file.w2list[:, 7]),
    "Bj_y": np.array(file.w2list[:, 8]),
    "i_chan": np.array(file.w2list[:, 9]),
    "is_cc_flag": np.array(file.w2list[:, 10] == 2),
    "n_hits": np.array(file.n_hits),
    "w2_gseagen_ps": np.array(file.w2list[:, 0]),
    "livetime": livetime * np.ones(len(primaries.pos_x)),
    "n_gen": n_gen * np.ones(len(primaries.pos_x)),
    "run_id": run_id,
    "frame_index": frame_index,
    "trigger_counter": trigger_counter,
    "event_no": np.array(unique_id).astype(int),
}
```

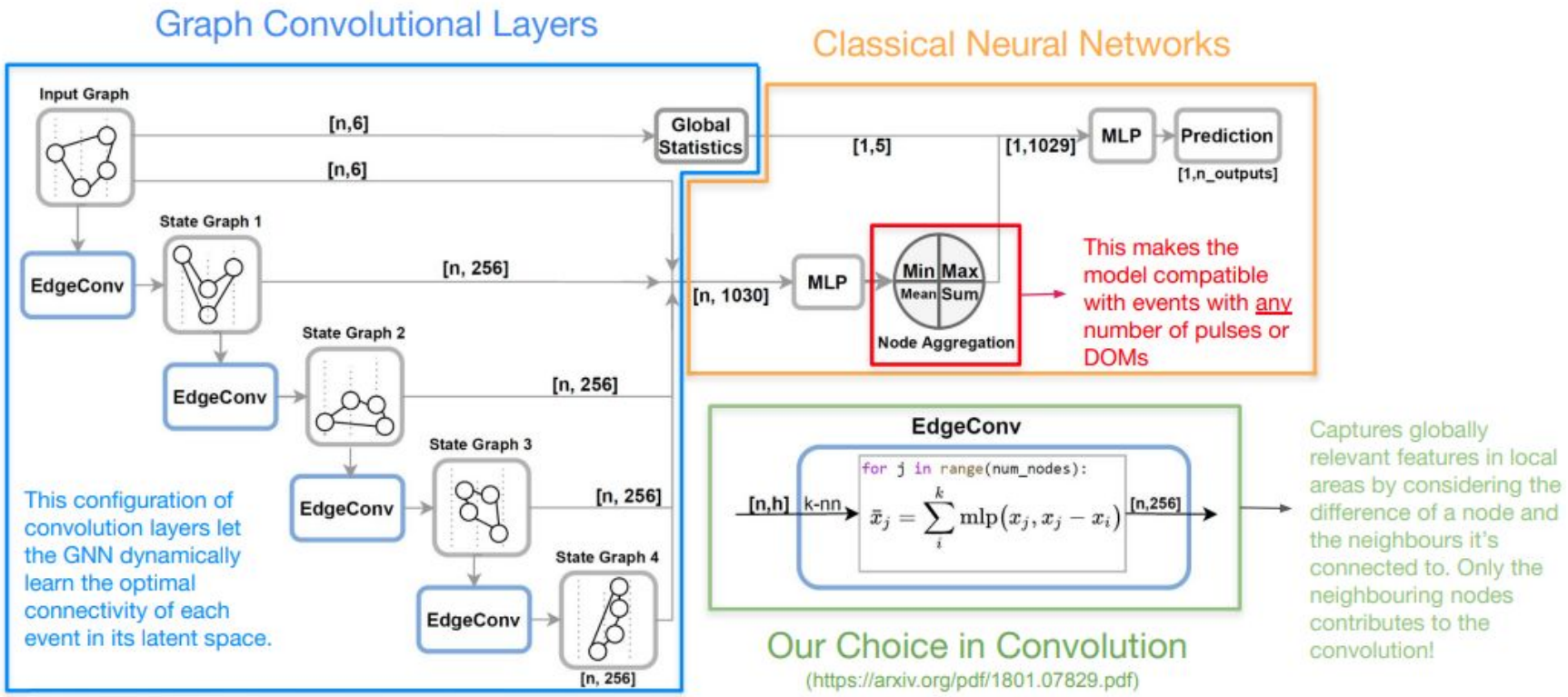
Index

- KM3NeT/ARCA and KM3NeT/ORCA. How do events look in KM3NeT?
- Brief description of GraphNet KM3NeT extractors.
- Discussion of preliminary results in ORCA 115.

GraphNeT in ORCA-115

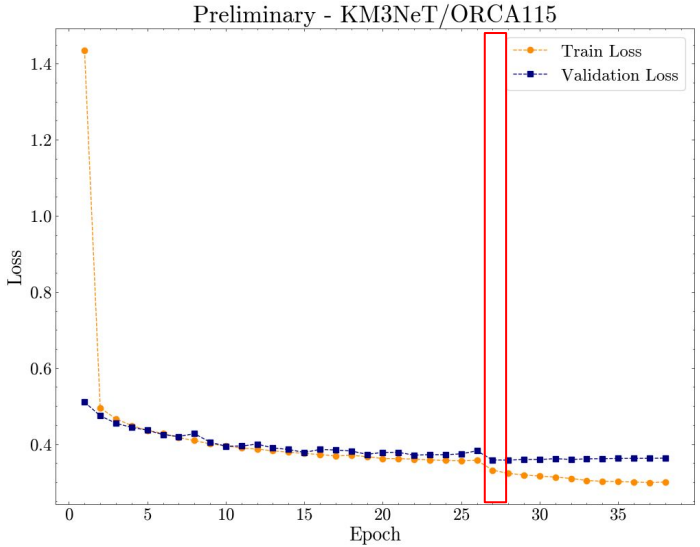
- Trained on the mc samples of ORCA-115
- Trained using **dynedge** model
- Trained for:
  - Track-cascade classification
  - Energy reconstruction
  - Direction reconstruction
  - Interaction Vertex position reconstruction
- Sample of:
  - ~400k ν_e+$\overline{\nu}_e$ 1-500 GeV
  - ~400k ν_μ+$\overline{\nu}_\mu$ 1-500 GeV

Dynedge



[https://www.nbi.dk/~petersen/Teaching/ML2022/Week5/IntroductionToGNNs_RasmusO.pdf]

Track shower classification



Node initialized with:

- pos_x, pos_y, pos_z, dir_x, dir_y, dir_z, t

Global pooling schemes:

- Min,max,mean, sum

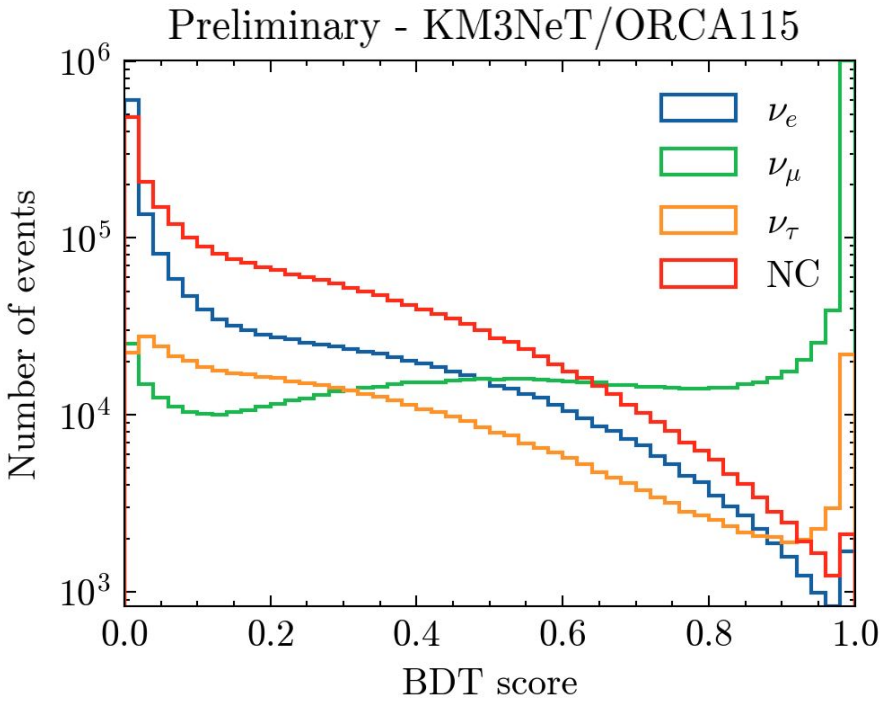
Common to all models:

- Saving the model after each epoch and then check which one to use.
- Using *ReduceLROnPlateau.*
- LR initialize in 1e-4 and ε set to 1e-6

	Epoch to min	N.Neigh	Loss function	Batch size	Optimizer	Trig
T&C	27	40	BinaryCross EntropyLoss	128	Adam	Triggered

Track shower Boosted Decision Tree (BDT)

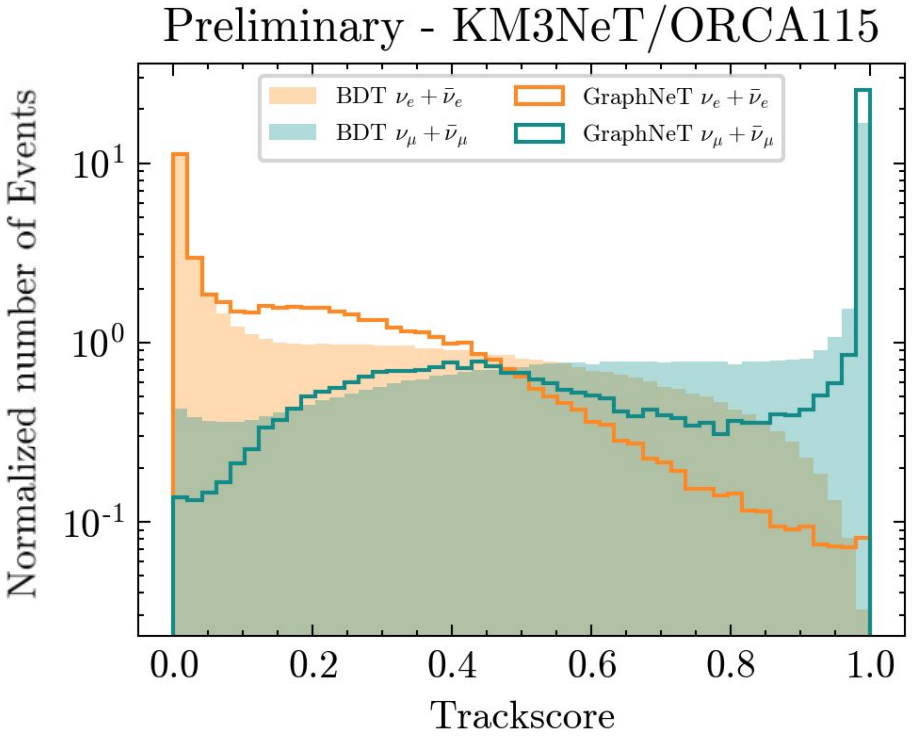
- Trained with 200k tracks (numu/anumu-CC) and 200k showers (nue/anue-CC).
- Model using the 25 features with highest impact in the classification.
- Hyperparameter optimization in unweighted training.
- Some pre-cuts: Only events reconstructed as upgoing, with more than 15 triggered hits and some further event quality reconstruction cuts.



[https://git.km3net.de/parapid/parampid]

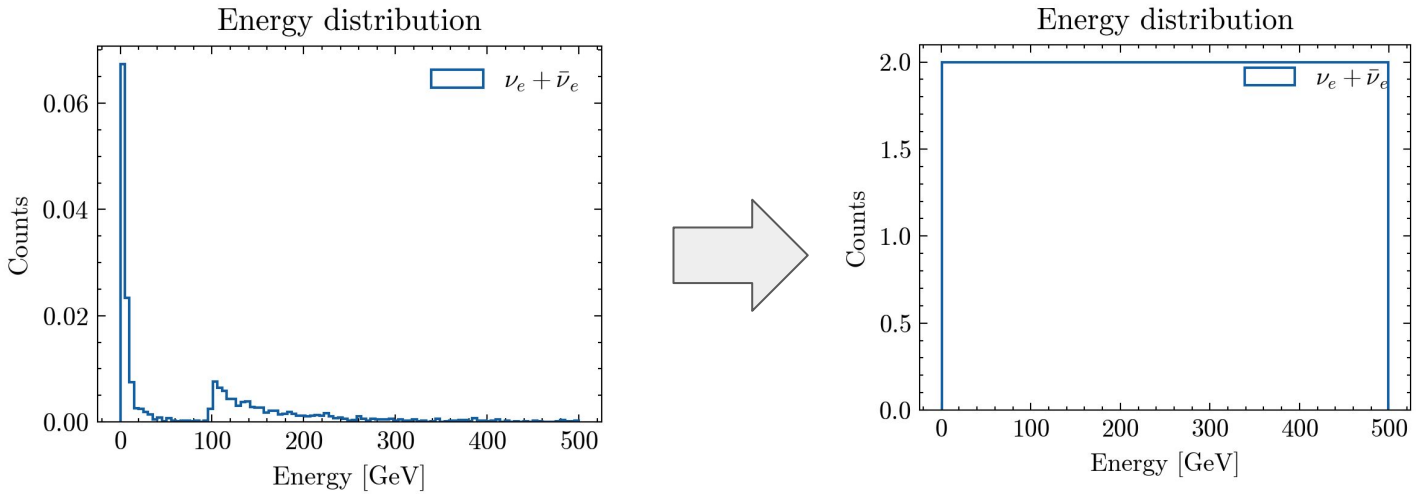
Track shower classification

- Similar performance.
- GraphNeT better in not classifying tracks as “very shower-like events”.
- **Not fair comparison.** BDT sample with precuts to reject bad quality reconstructed events while GraphNeT includes all triggered events.

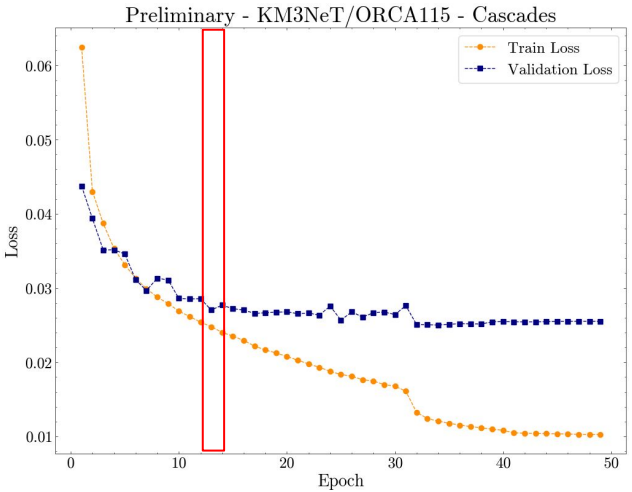


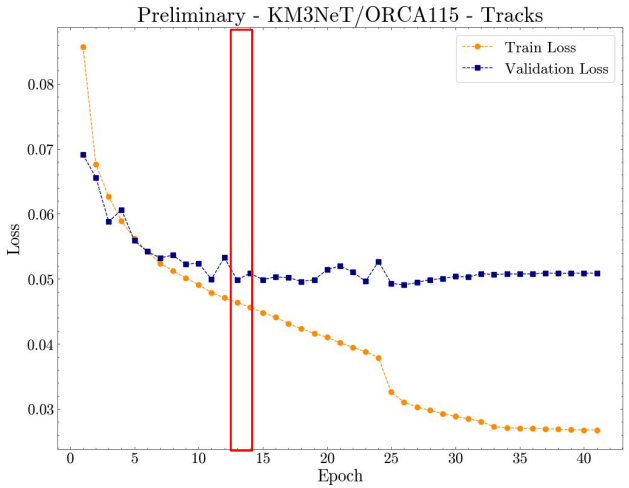
Energy reconstruction

- Training on a “reweighted-to-uniform” spectrum.
- Trained separately on tracks and cascades.
- Trained with the target “True neutrino energy”.



Energy reconstruction





Node initialized with:

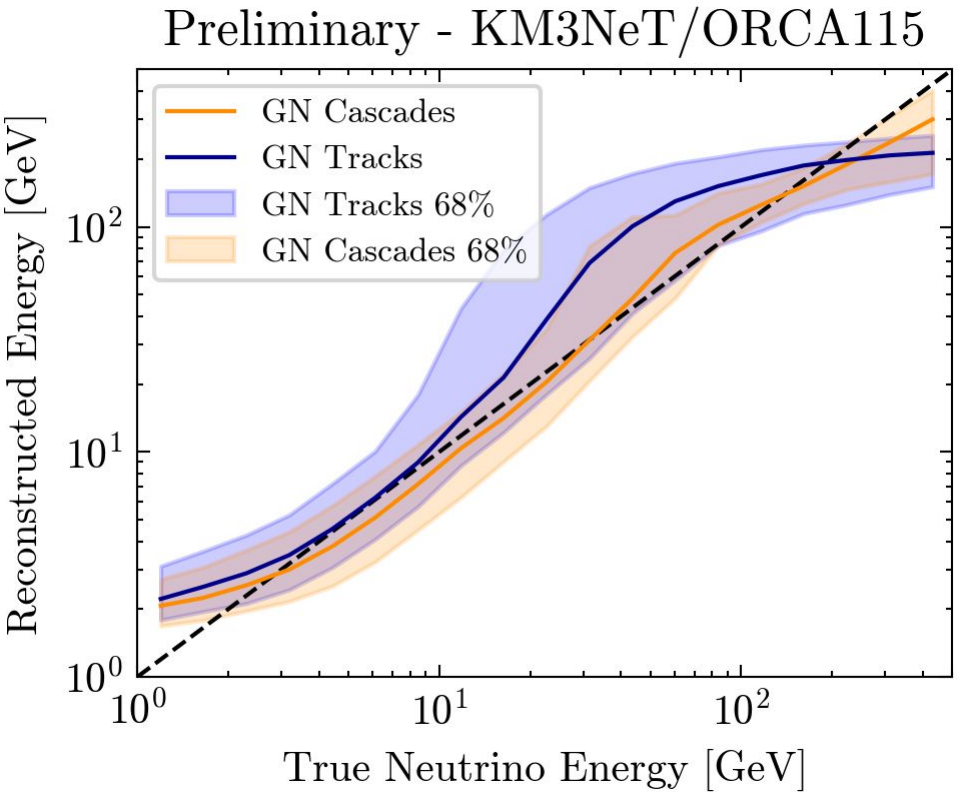
- pos_x, pos_y, pos_z, ToT

Global pooling schemes:

- Min,max,mean, sum

	Epochs To conv	N.Neigh	Loss function2	Batch size	Optimizer	Trig
Ereco tracks	14	16	LogCoshLoss	128	Adam	Triggered
Ereco showers	13	16	LogCoshLoss	128	Adam	Triggered

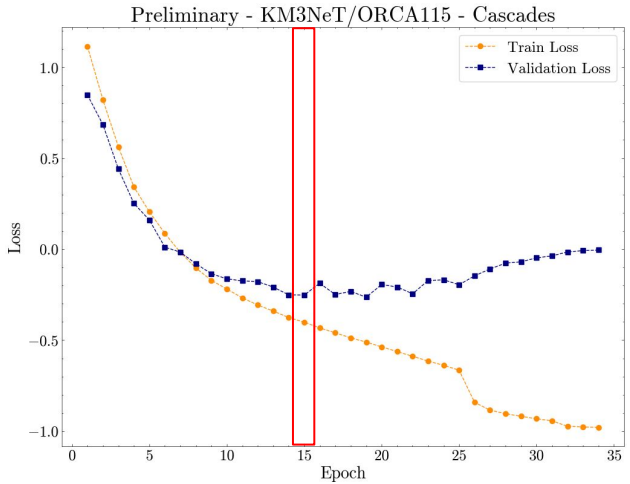
Energy reconstruction

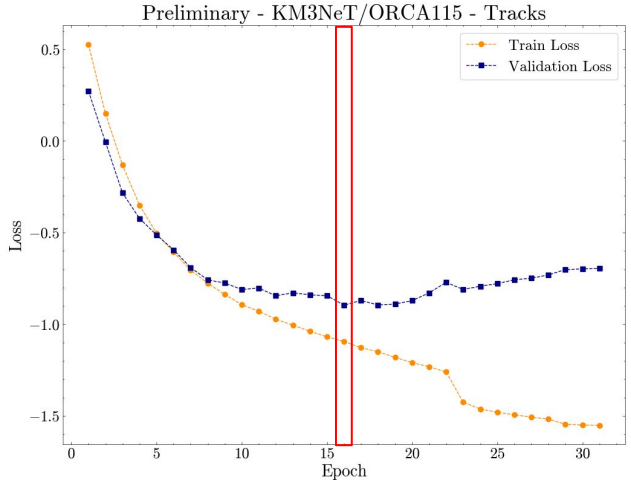


- Performance comparable to the standard likelihood methods.
- Room for improvement, but cascade results very promising.
- Different approach rather than targeting the true neutrino energy?

  Visible energy as a candidate.

Direction reconstruction





Node initialized with:

- pos_x, pos_y, pos_z, dir_x, dir_y, dir_z, t

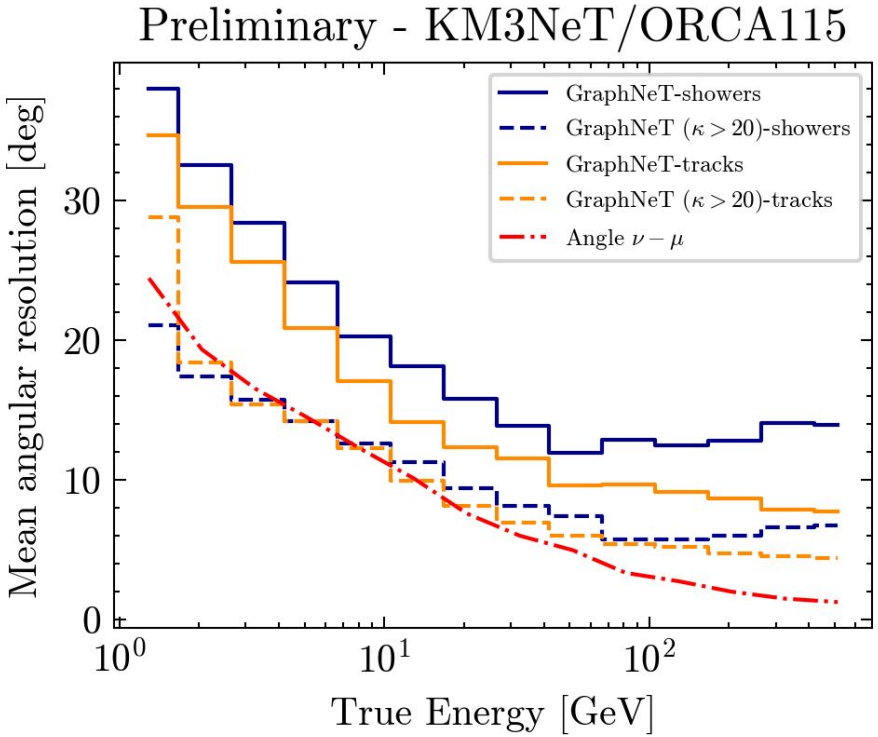
Global pooling schemes:

- Min,max,mean, sum

	Epochs To min	N.Neigh	Loss function	Batch size	Optimizer	Trig
DirReco tracks	16	16	VonMisesFisher3DLoss	128	Adam	Triggered
DirReco showers	15	16	VonMisesFisher3DLoss	128	Adam	Triggered

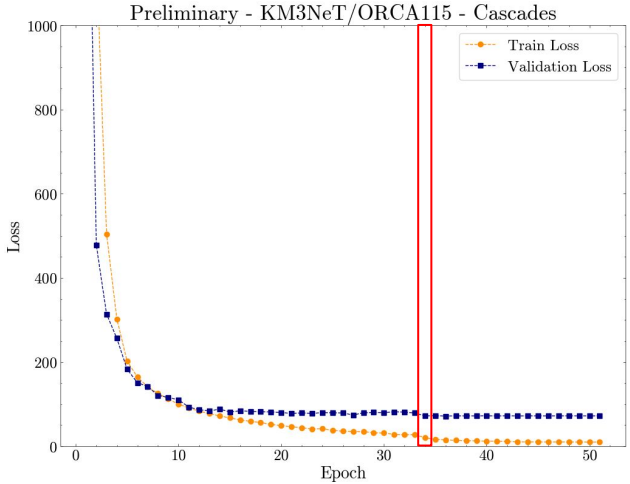
Angular resolution

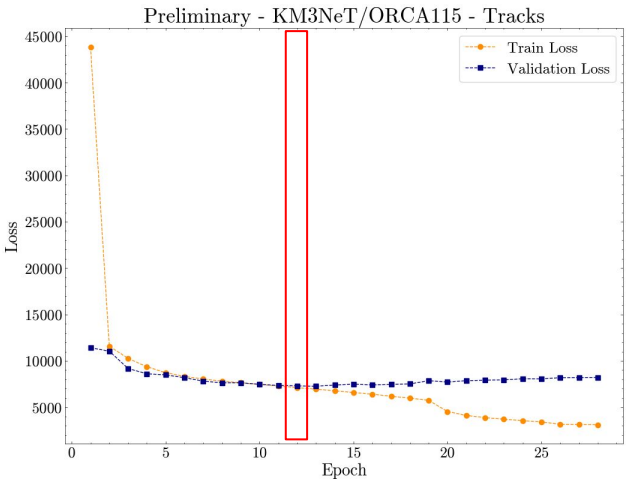
- Performance comparable with the standard likelihood methods.
- Studies of whether Von Mises-Fisher kappa can measure how difficult to reconstruct an event is.
- Kappa cut (K>20) leaves 50% of the ν_e and 70% of the ν_μ.
- "Angle ν-μ" understood as the angle between neutrino and muon in ν_μ-CC events.



$$Ang.\,Resolution = arcos(x_1.x_2 + y_1.y_2 + z_1.z_2)$$

Vertex position reconstruction





Node initialized with:

- pos_x, pos_y, pos_z, dir_x, dir_y, dir_z, t

Global pooling schemes:

- Min,max,mean, sum

	Epochs To conv	N.Neigh	Loss function	Batch size	Optimizer	Trig
DirReco tracks	12	16	MSELoss3D*	128	Adam	Triggered
DirReco showers	34	16	MSELoss3D*	128	Adam	Triggered

*Not among the the Loss functions available in GraphNet.

Vertex position reconstruction

RMSE [m]	
GraphNeT tracks	88.7
GraphNeT showers	7.1
GraphNeT contained*	31.0

RMSE (Root Mean Squared Error)

*First fast approach: Contained within a cube with faces touching min,max dom-position in x, y and z

- Preliminary results.
- MSELoss3D not implemented in GraphNeT.
- Seems not to be possible to reconstruct events with vertex far away from the detector. Maybe different approach like reconstruction of the entry point in the detector or train only on contained tracks.

Final remarks

- **First attempt to create KM3NeT/GraphNeT datasets** and make some predictions.
- **Room from improvement**, specially in Energy and Vertex position reconstructions.
- **Promising results** already comparable with KM3NeT classic likelihood methods for reconstruction.

  and BDT models for classification, although comparisons are still in need of more work to set up proper benchmarks.
- Software ready to produce datasets of a “continuously” changing detector.

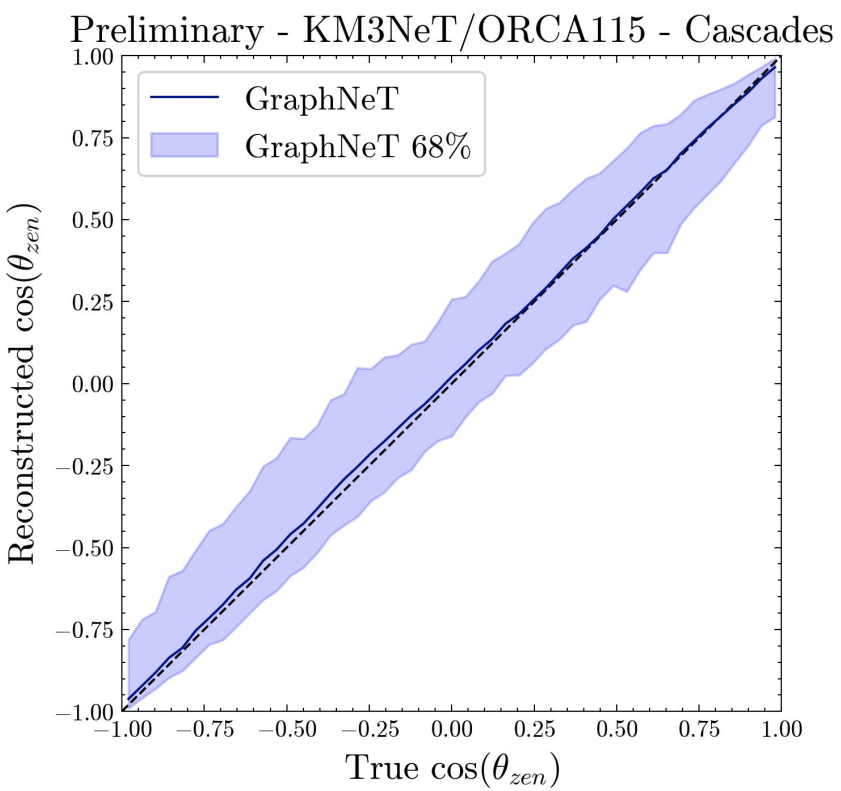
Thank you!

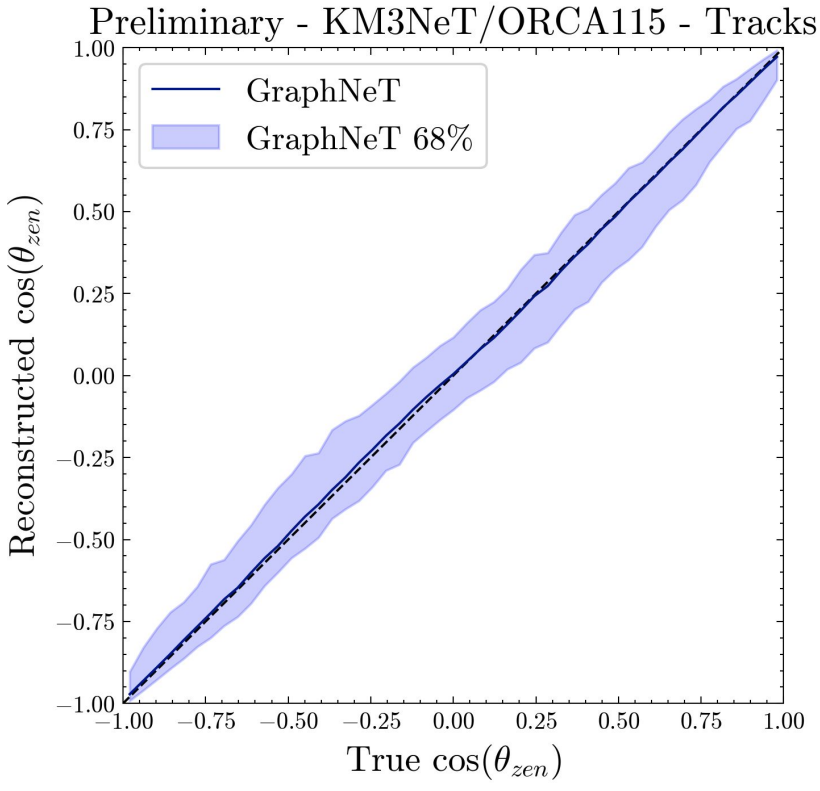
Backup

MSELoss3D

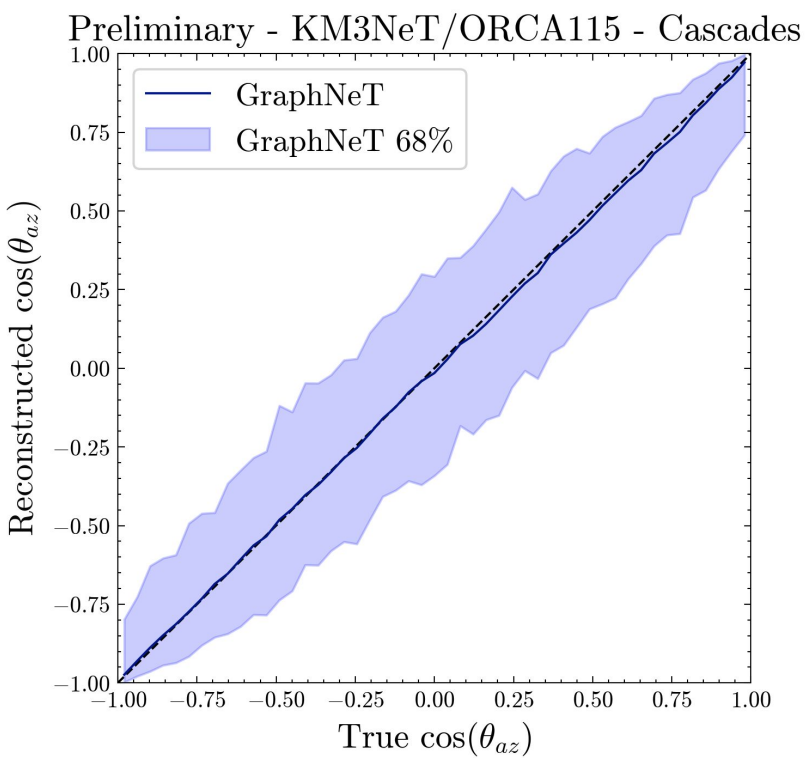
```python
class MSELoss3D(LossFunction):
    """Mean squared error loss for 3D points."""

    def _forward(self, prediction: Tensor, target: Tensor) -> Tensor:
        """Implement loss calculation."""
        # Check(s)
        target = target.reshape(-1, 3)

        assert prediction.dim() == 2
        assert prediction.size() == target.size()

        elements = torch.mean((prediction - target) ** 2, dim=-1)
        return elements
```

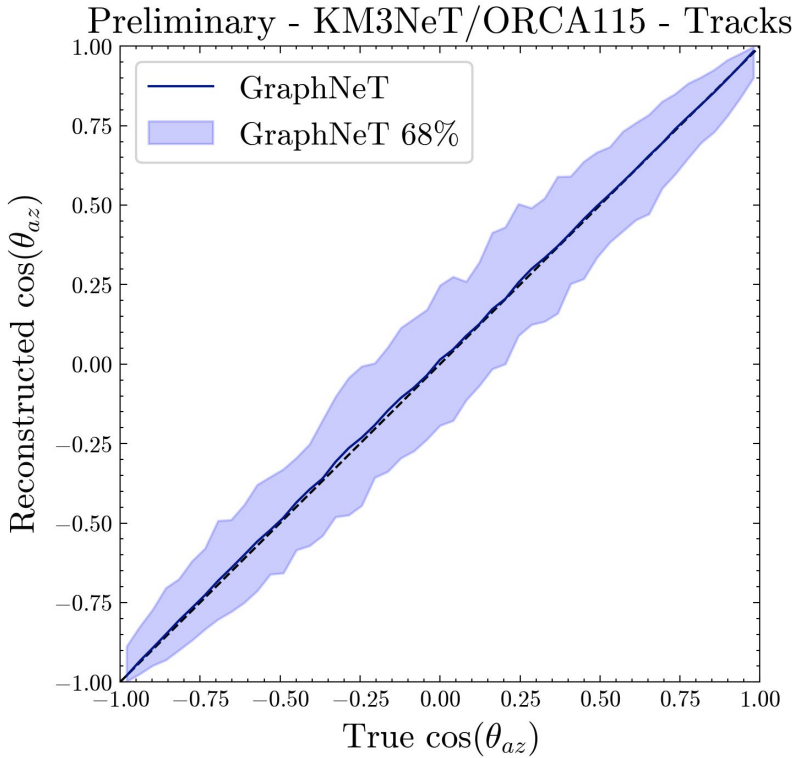
Direction true vs reco



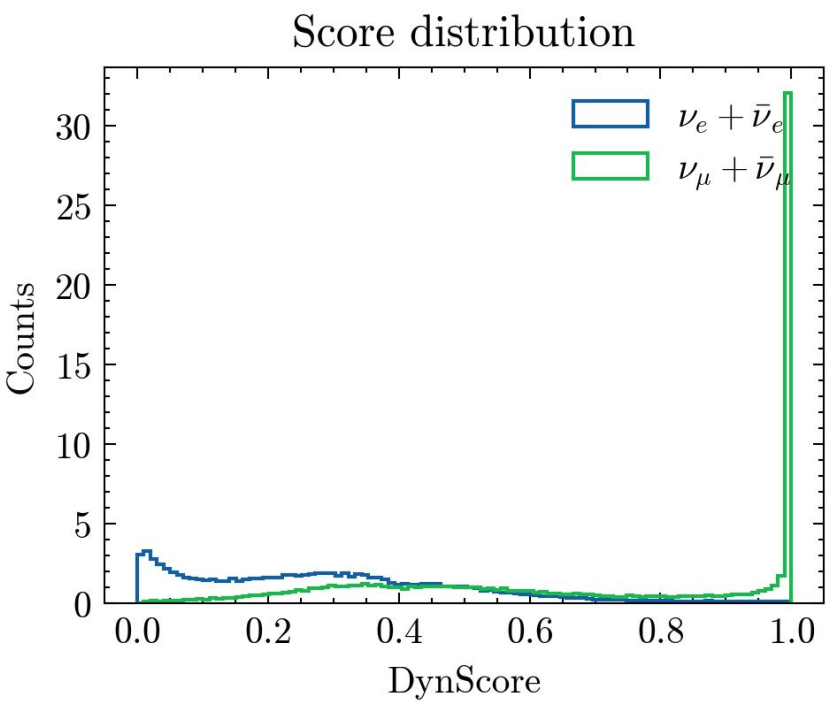


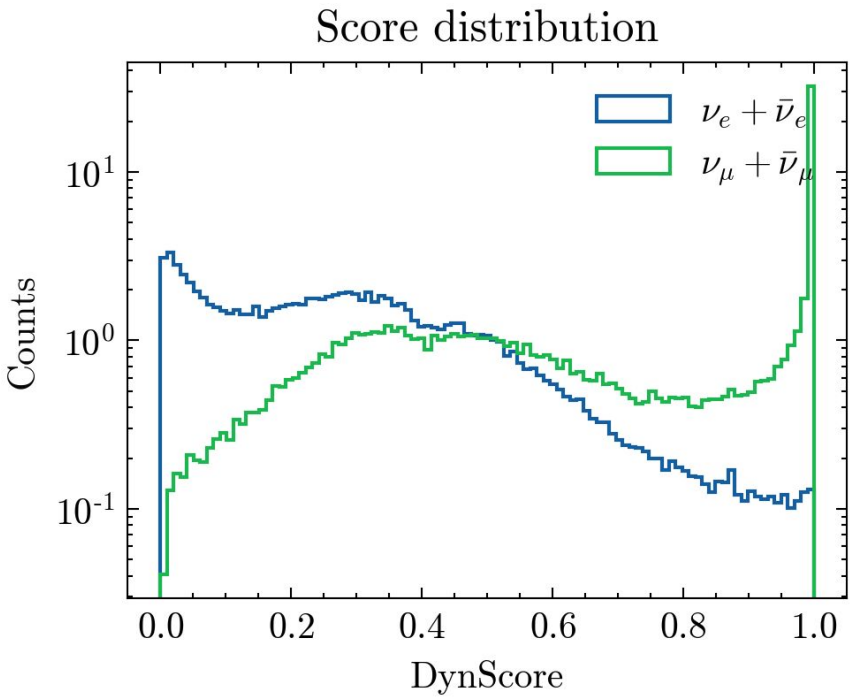
Direction true vs reco



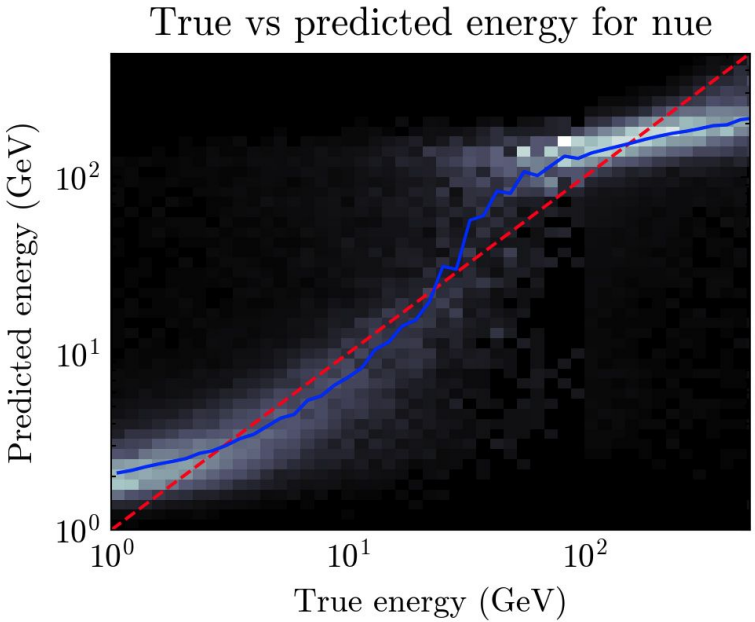


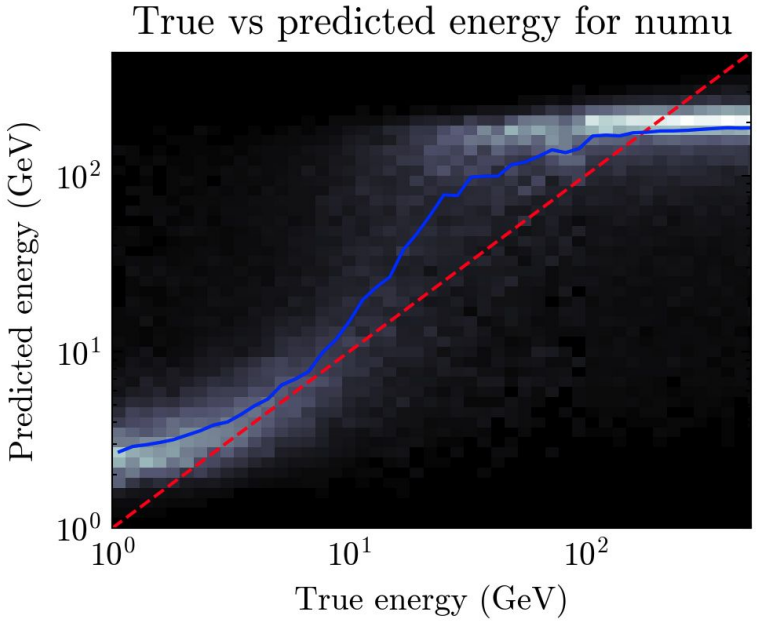
Extra Plots: Ugly ORCA6v7.1 results



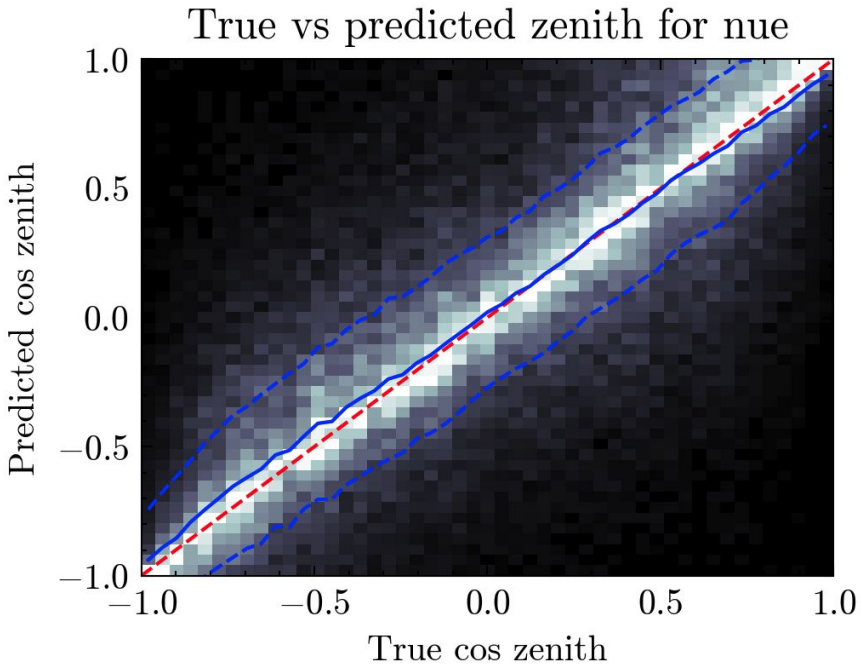


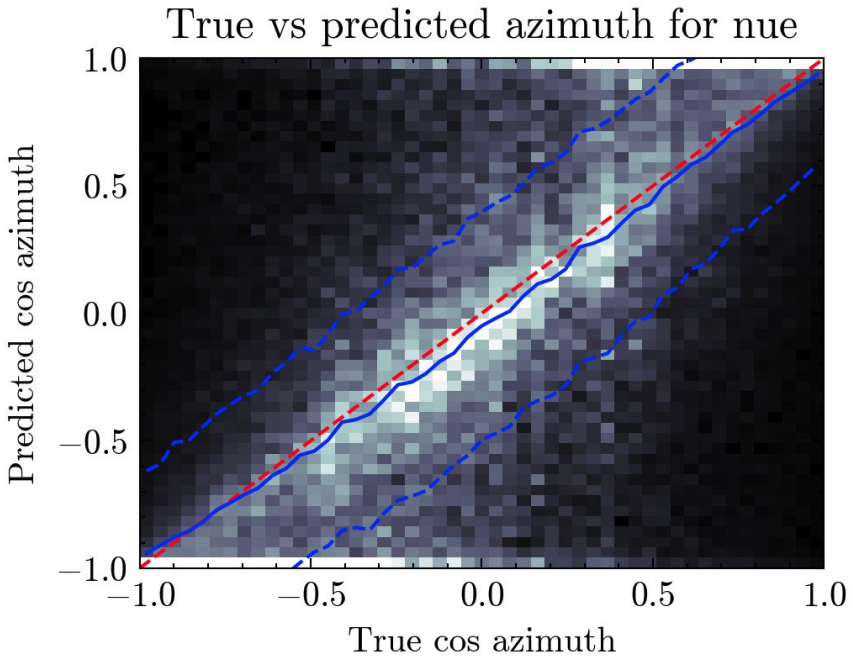
Extra Plots: Ugly ORCA6v7.1 results





Extra Plots: Ugly ORCA6v7.1 results





Extra Plots: Ugly ORCA6v7.1 results