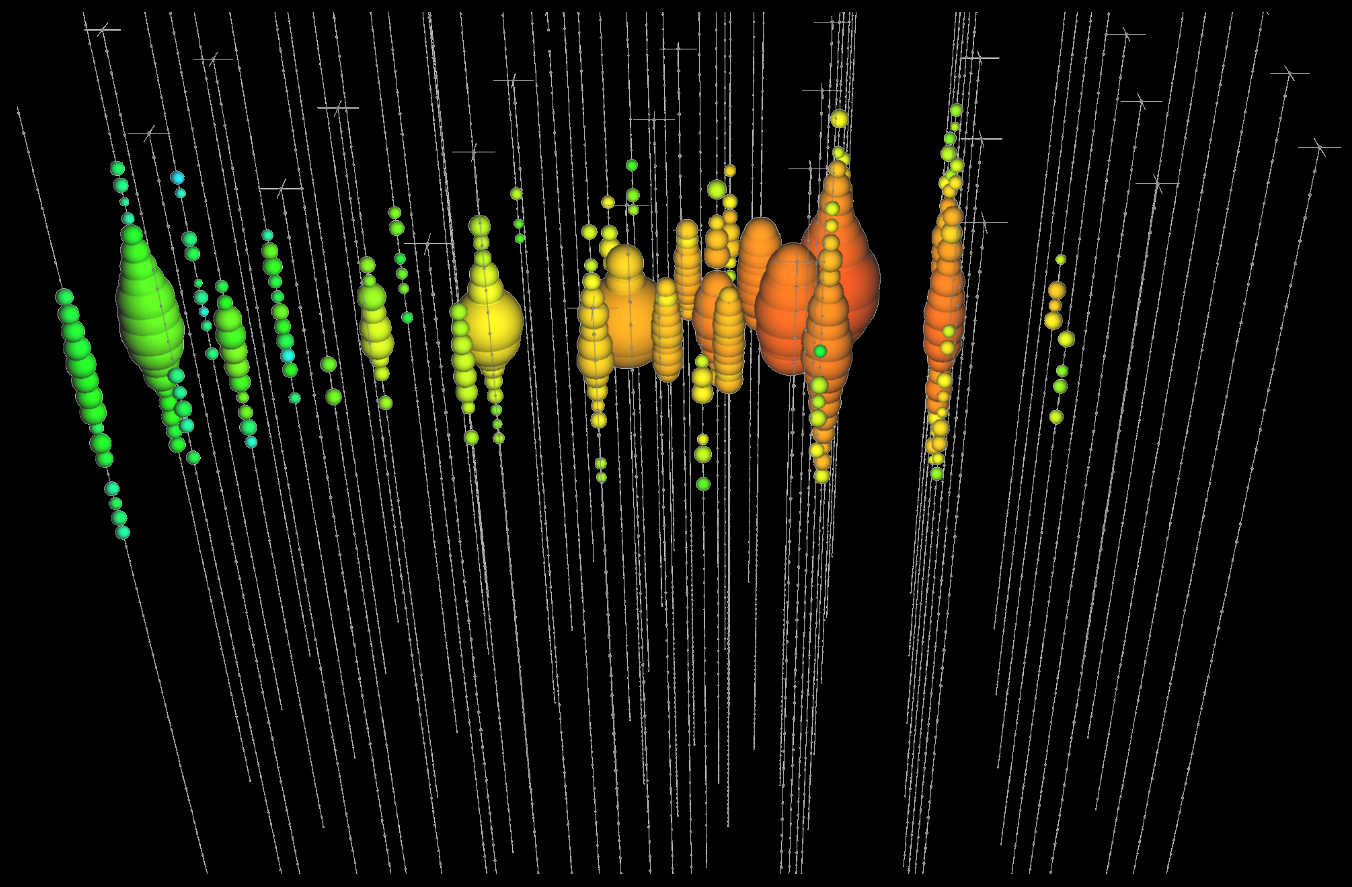
Machine Learning in IceCube

Landscape of emerging developments

SCAP 2021, Claudio Kopper - Heavily borrowed (with permission) from M. Huennefeld's NPML presentation

Event Topologies

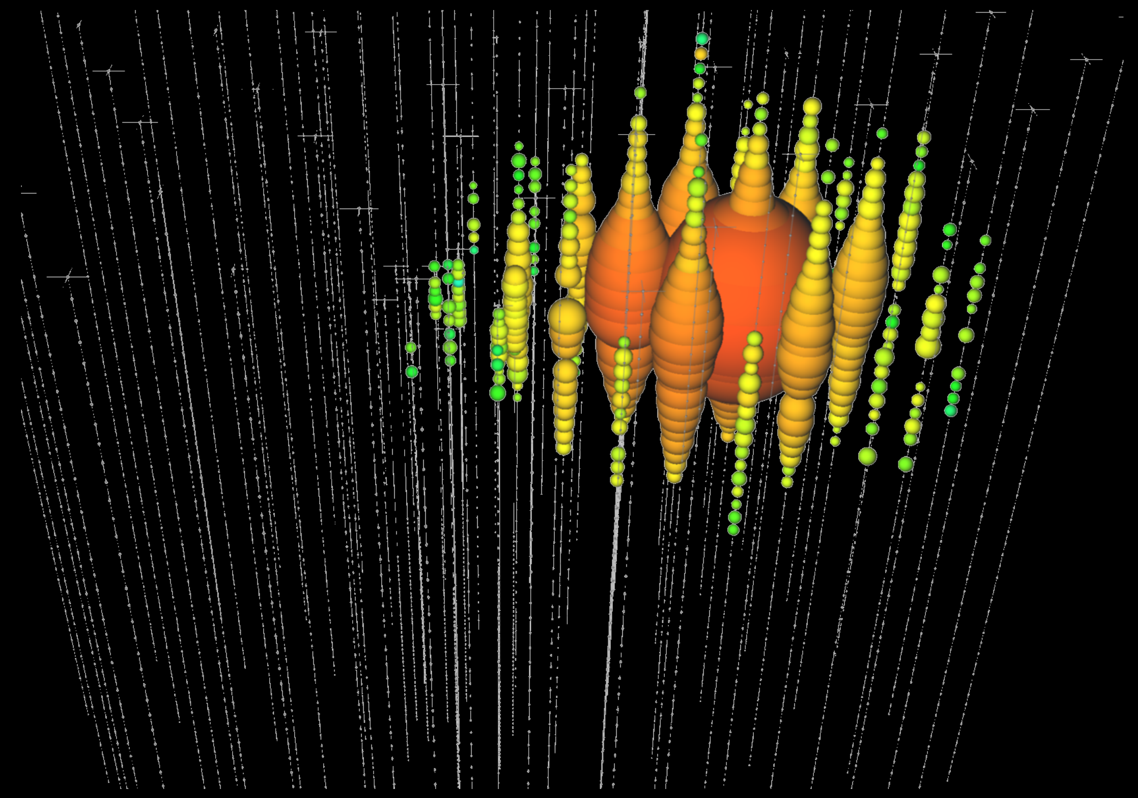
CC ν_μ



$$\nu_\mu + N \rightarrow \mu + X$$

Track

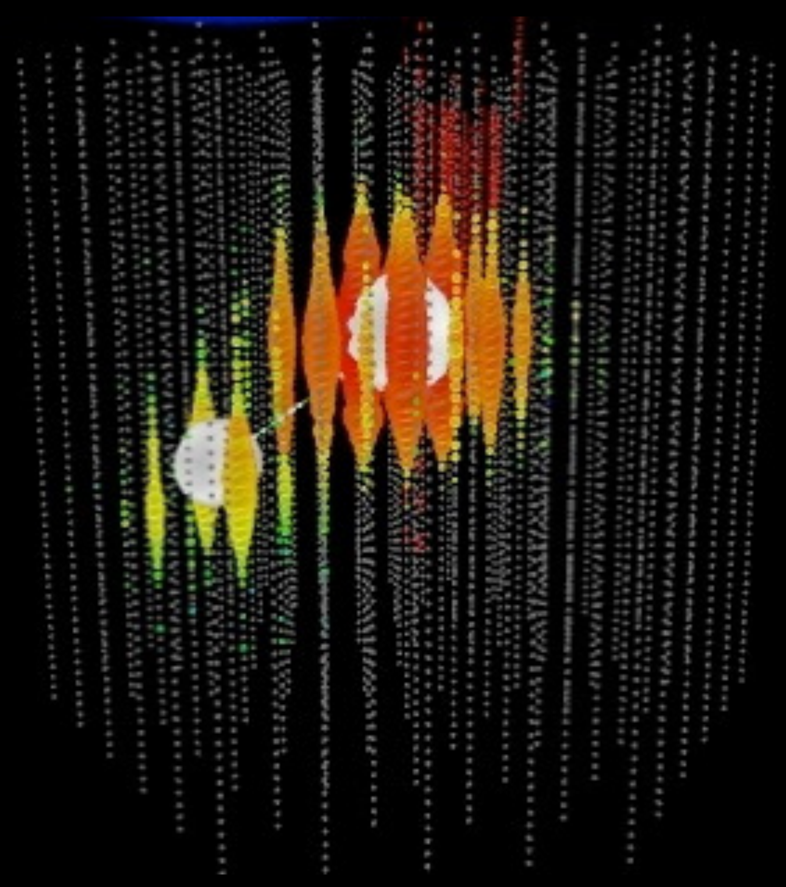
CC ν_e / NC ν_*



$$\nu_e + N \rightarrow \mathrm{e} + X$$
$$\nu_* + N \rightarrow \nu_* + X$$

Cascade

CC ν_τ

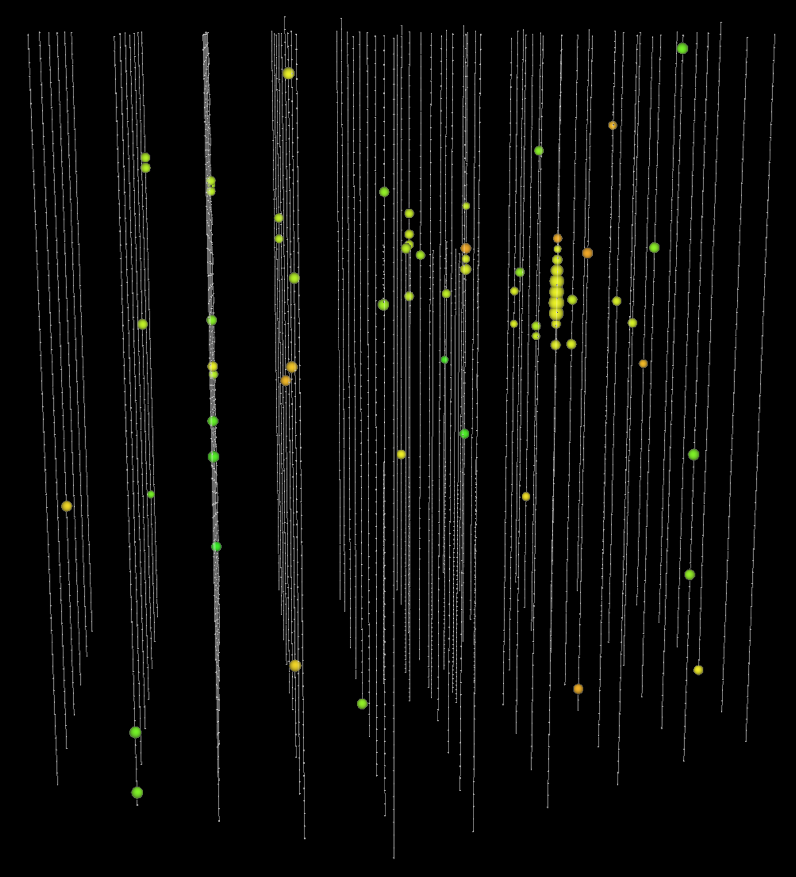


$$\nu_\tau + N \rightarrow \tau + X$$

Cascade / Track / Double-Cascade

Event Topologies

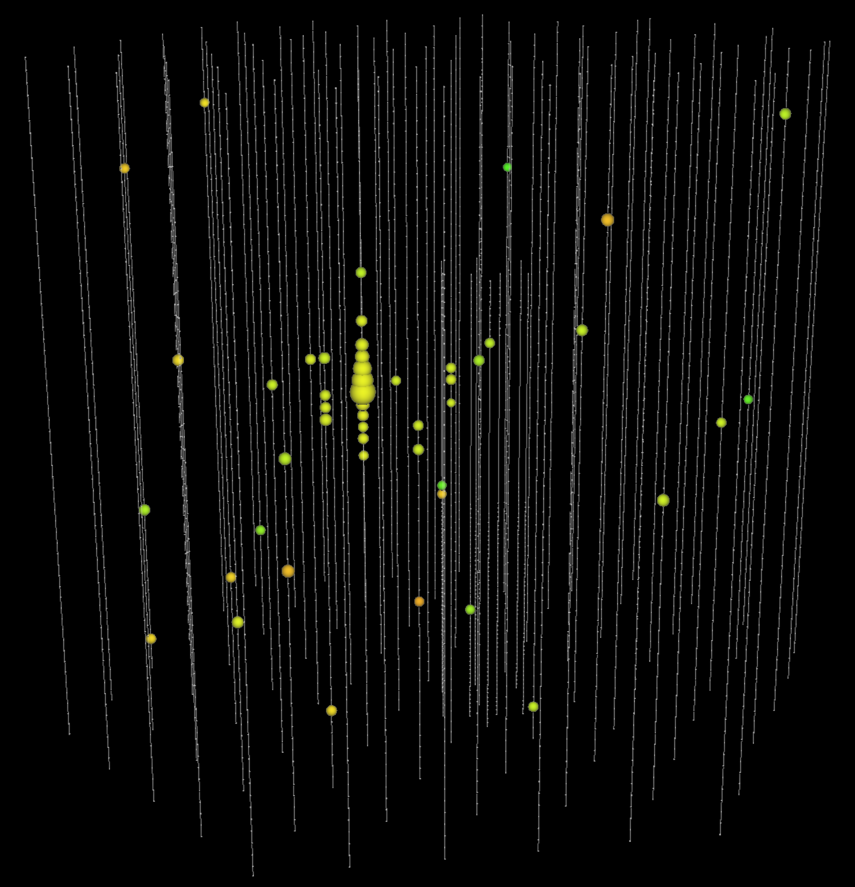
CC ν_μ



$$\nu_\mu + N \rightarrow \mu + X$$

Track

CC ν_e / NC ν_*

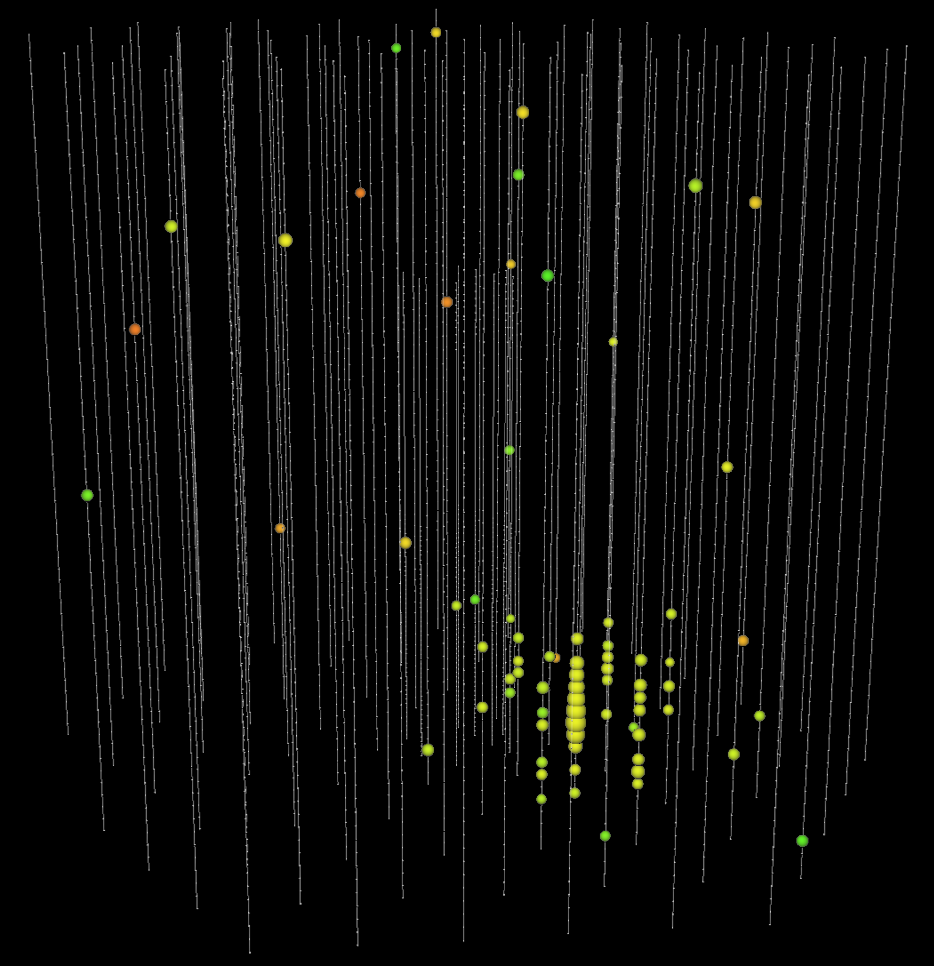


$$\nu_e + N \rightarrow \mathrm{e} + X$$

$$\nu_* + N \rightarrow \nu_* + X$$

Cascade

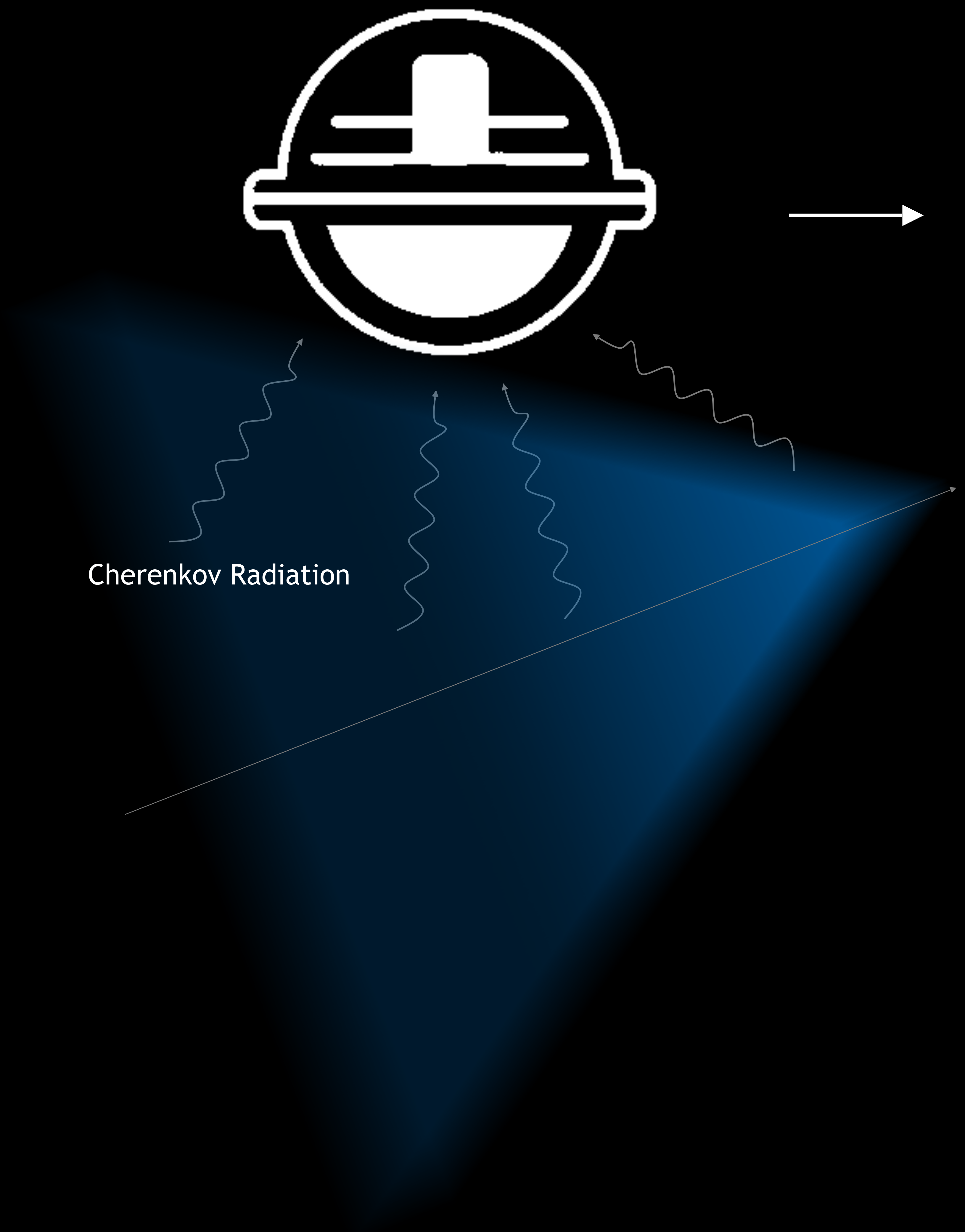
CC ν_τ

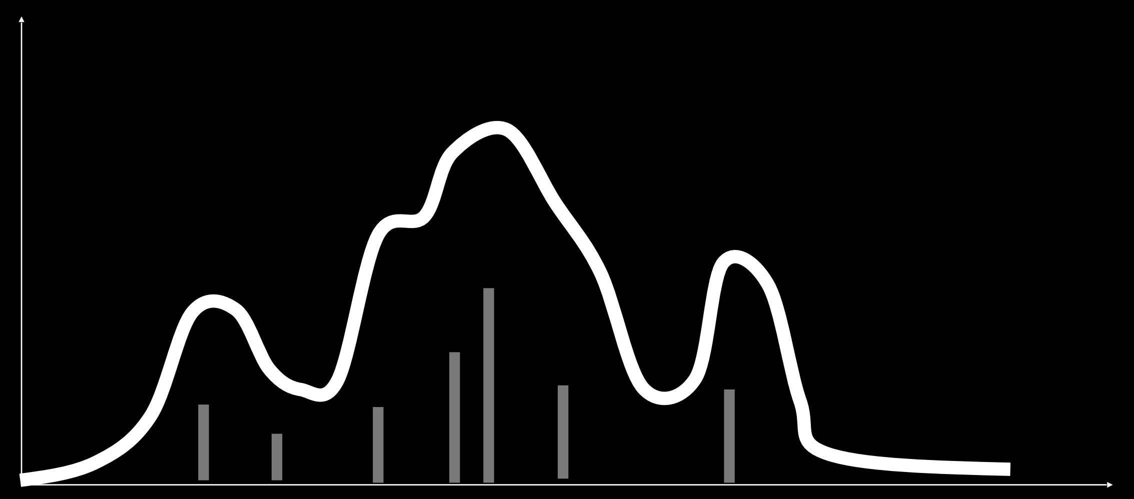


$$\nu_\tau + N \rightarrow \tau + X$$

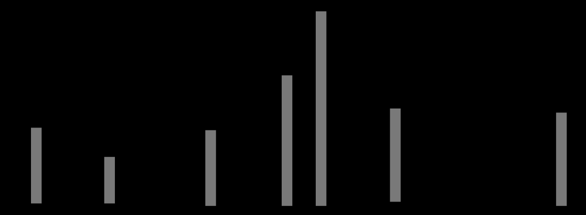
Cascade / Track / Double-Cascade

Event Reconstruction

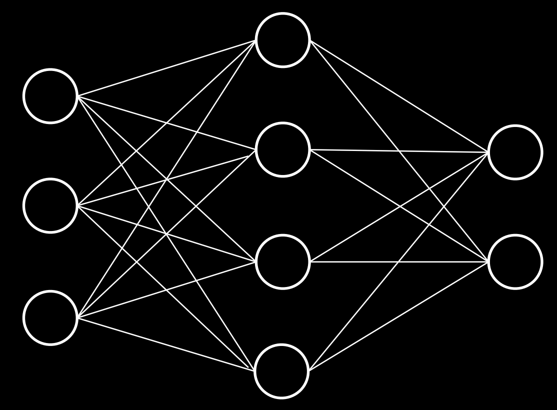




Pulse Series: (t_i, q_i)

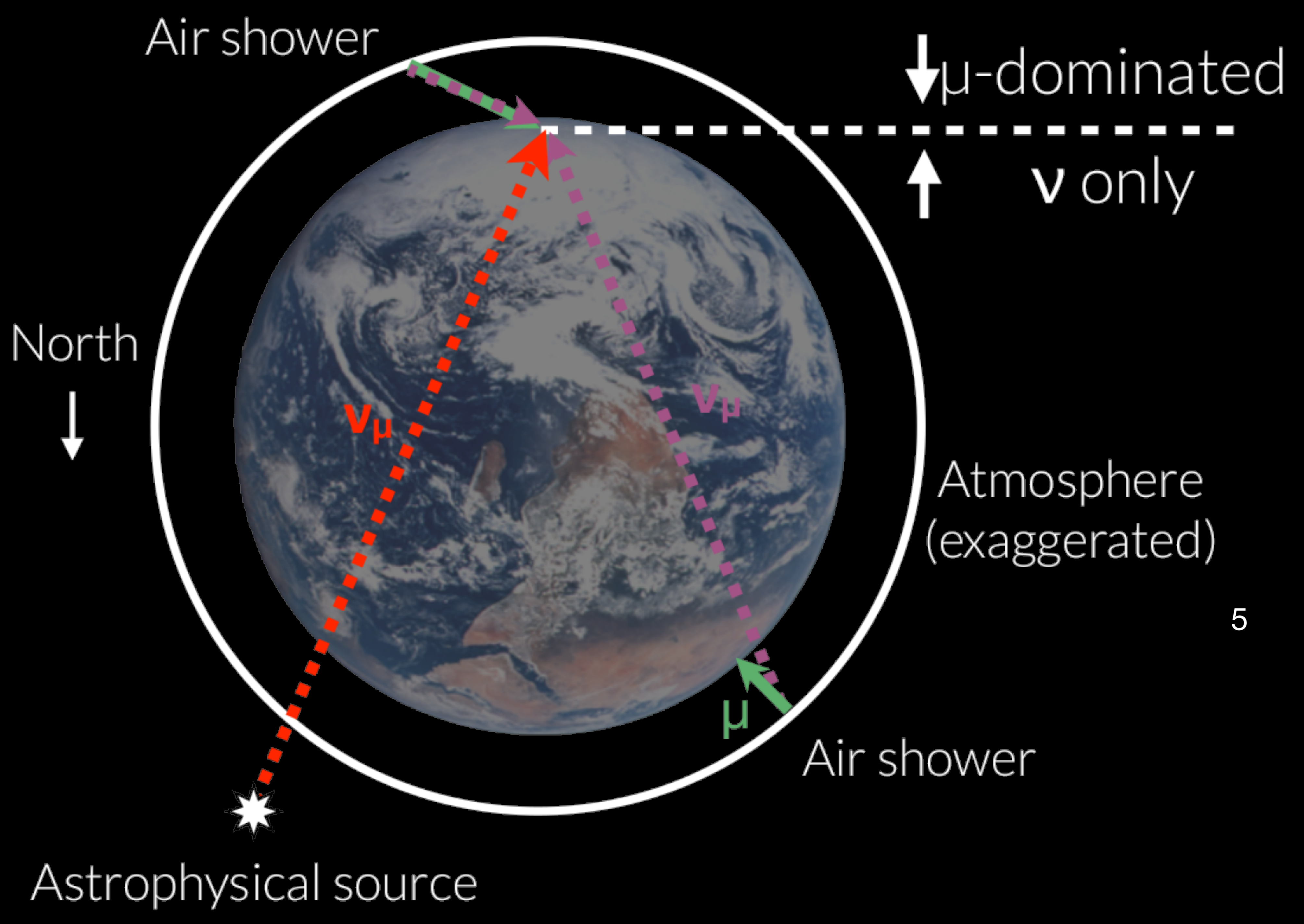


$$\mathscr{L}\left(\vec{x}\,\middle|\,\vec{\theta}\right) = \prod_i p(x_i \mid \vec{\theta})$$



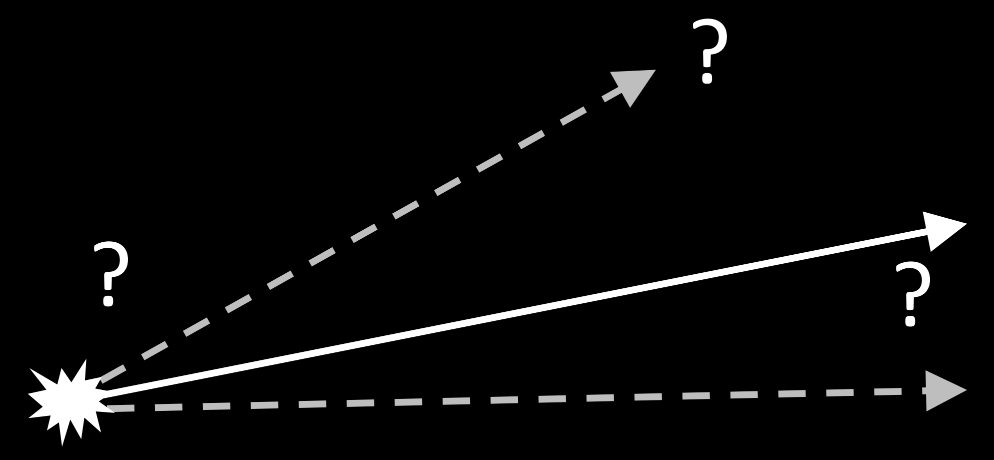
Main Reconstruction Tasks

Event Selection and Classification



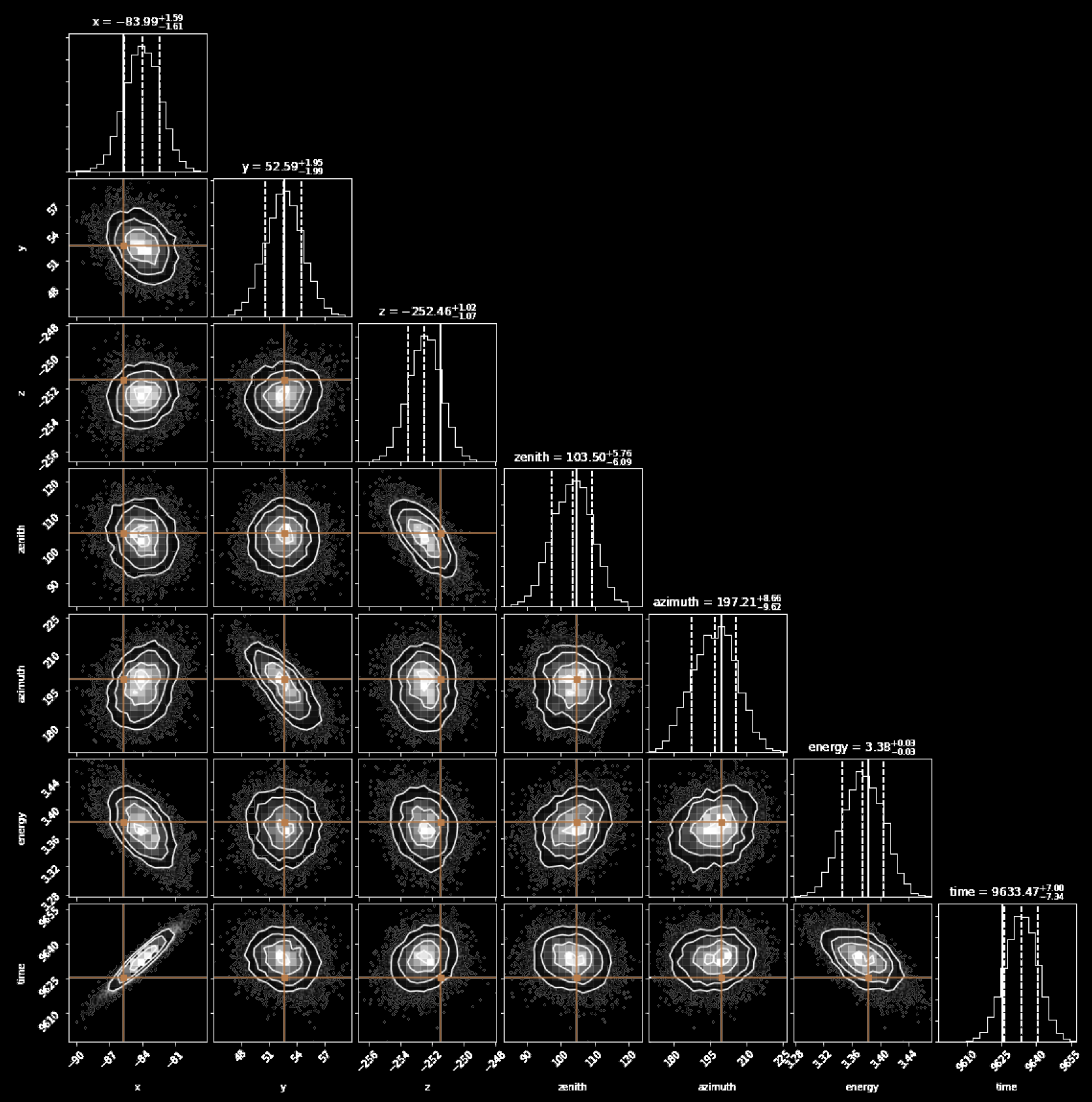
Rates:
Atmospheric Muons: ~10^3 Hz
Atmospheric Neutrinos: ~10^{-3} Hz
Astrophysical Neutrinos: ~10^{-7} Hz

Estimation of Event Parameters



$\theta, \phi, E, \vec{x}$ ...

Uncertainty Estimation



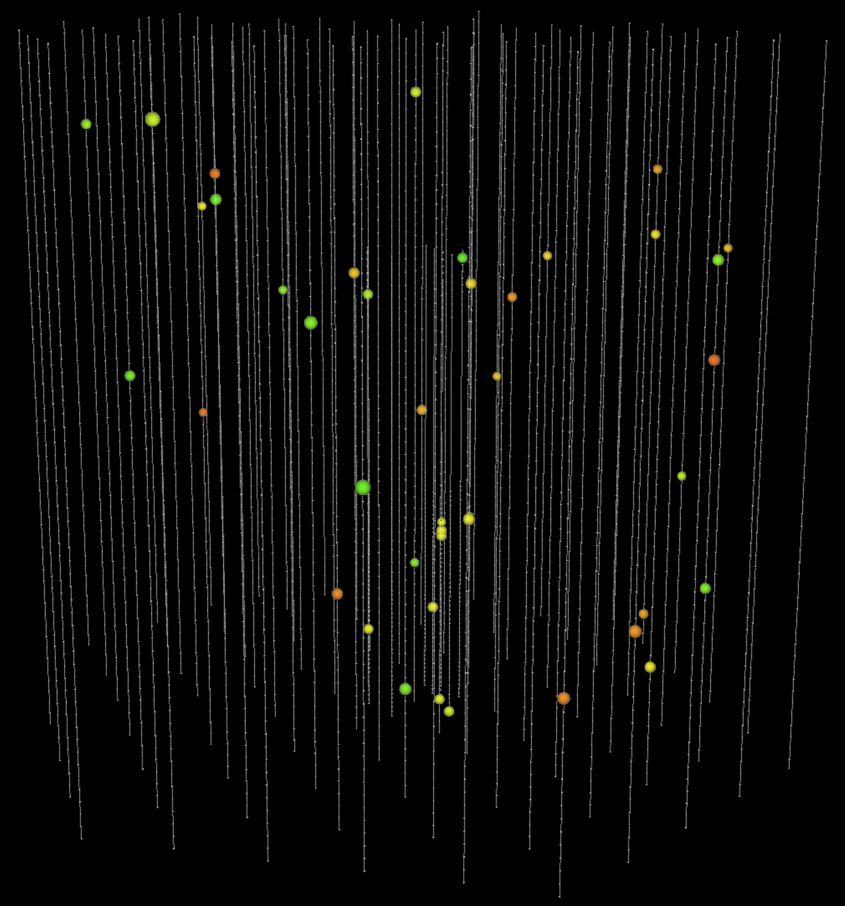
Challenges and Potential for ML

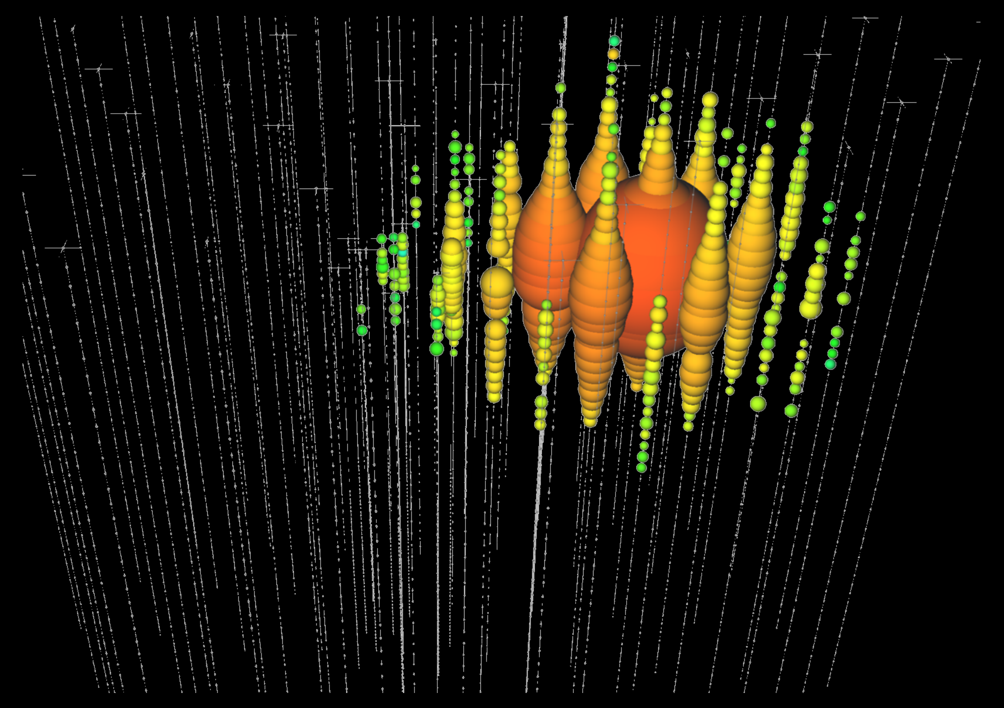
Challenges for traditional reconstruction methods

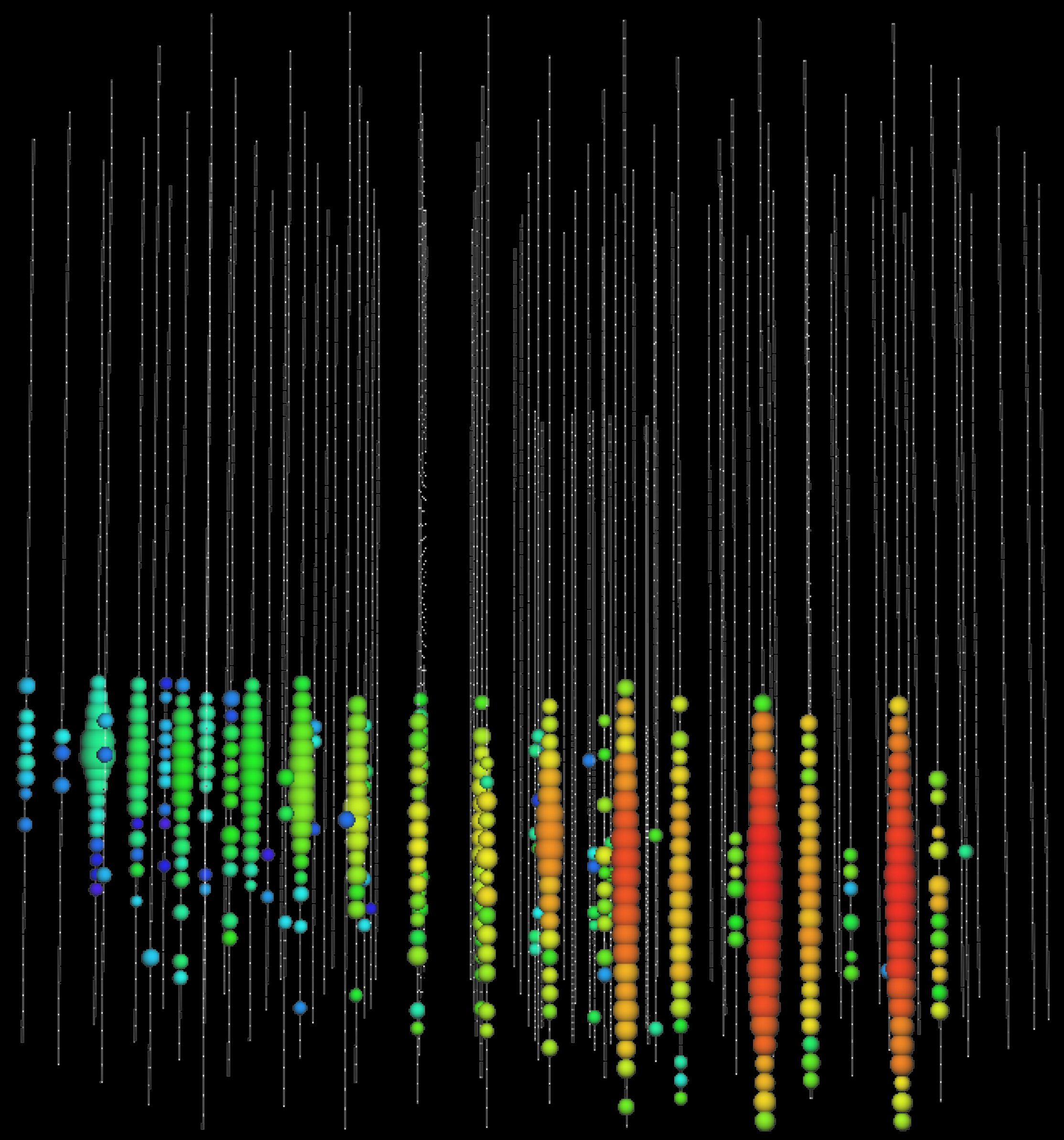
- High-dimensionality and complexity of data
- Defining problem/likelihood can be difficult
- Computation of likelihood can be intractable
- Energy spans over many orders of magnitude
- Time constraints on event reconstruction

Potential for machine and deep learning

- Can handle raw and complex data
- Problems can easily be defined: setting up likelihood function is not necessary
- Inference is typically extremely fast: will enhance real-time alerts and follow-ups





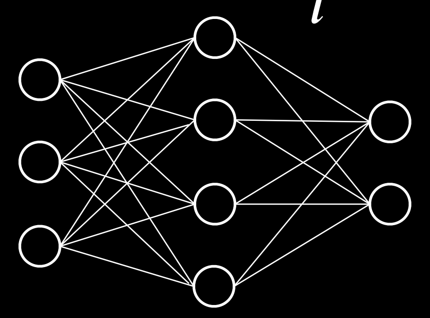


ML Applications

“Classical” ML

- High-level and low-dimensional input data:

$$\mathscr{L}\left(\vec{x}\,\middle|\,\vec{\theta}\right) = \prod_i p(x_i \,|\, \vec{\theta}\,)$$

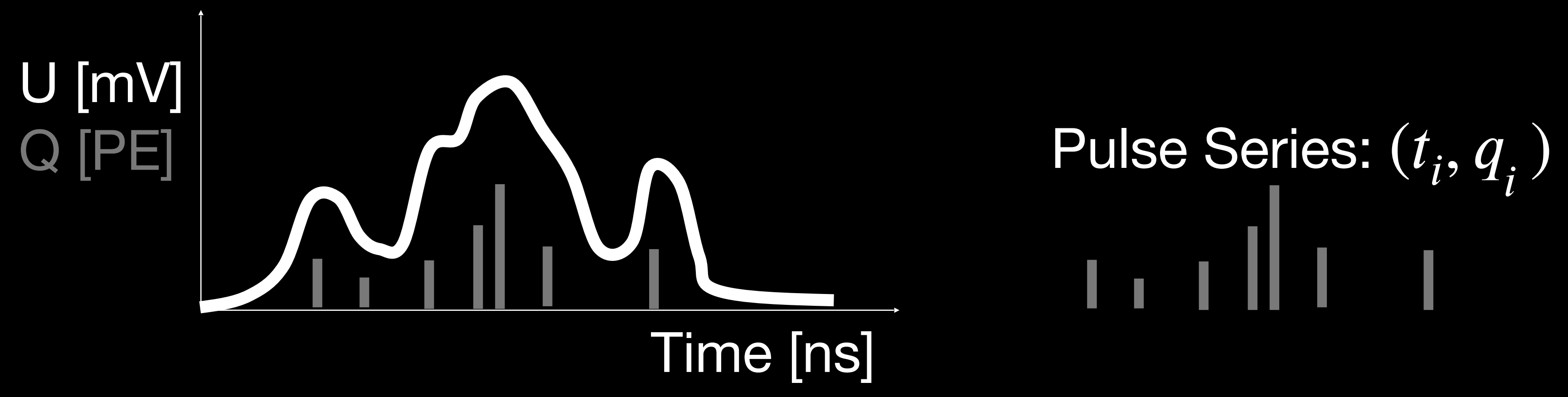


- Input features rely on previous reconstructions or summary statistics of pulses

- Mainly used in classification tasks, but also in development of analysis methodology and in regression tasks

Deep Learning

- Raw and high-dimensional input data:



- Input data typically does not rely on previous reconstructions

- Mainly used in classification and regression tasks

- Often applied very early on in the processing chain due to fast inference time

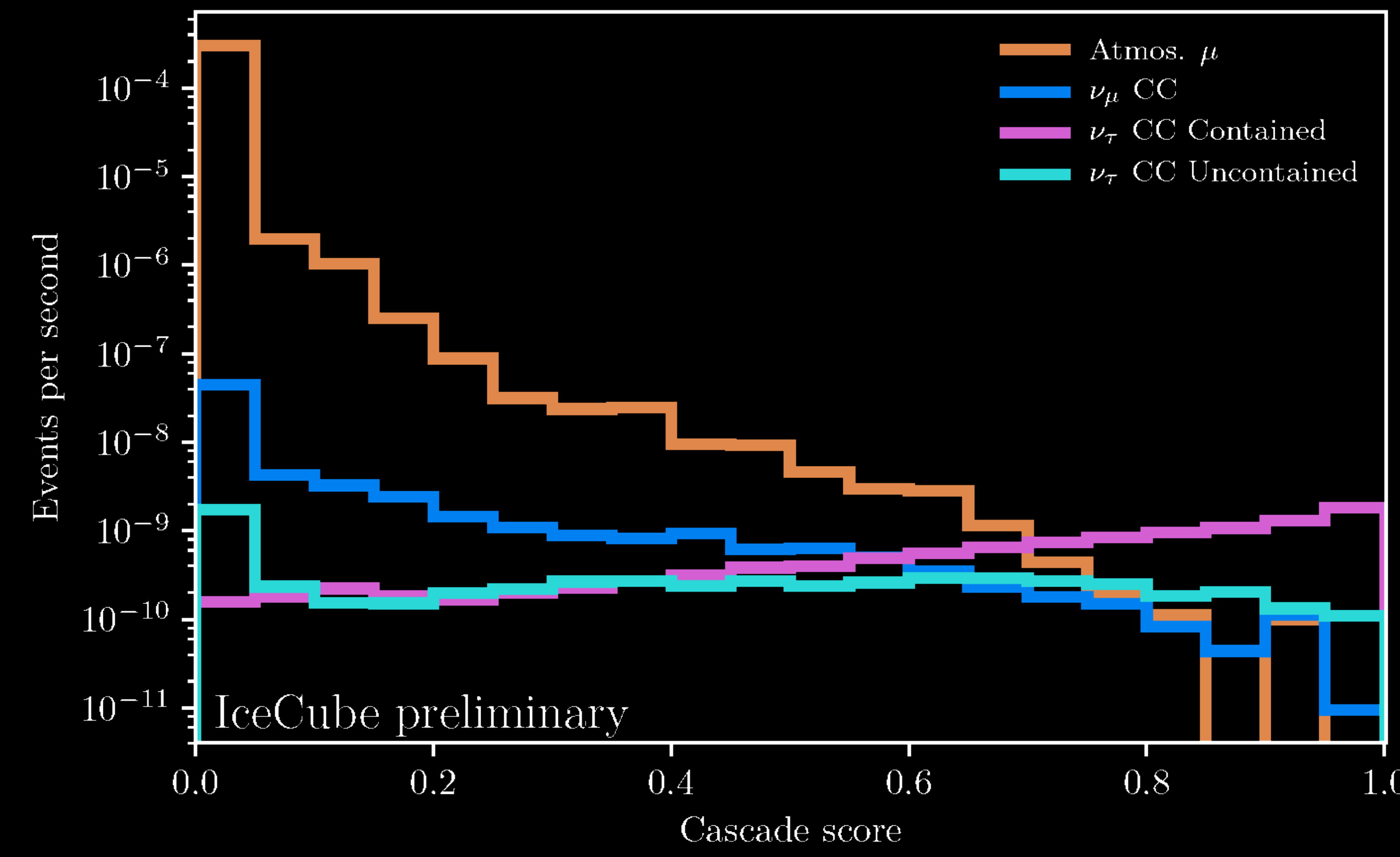
Examples

(**Note:** almost **all** new reconstruction development is done using Machine Learning)

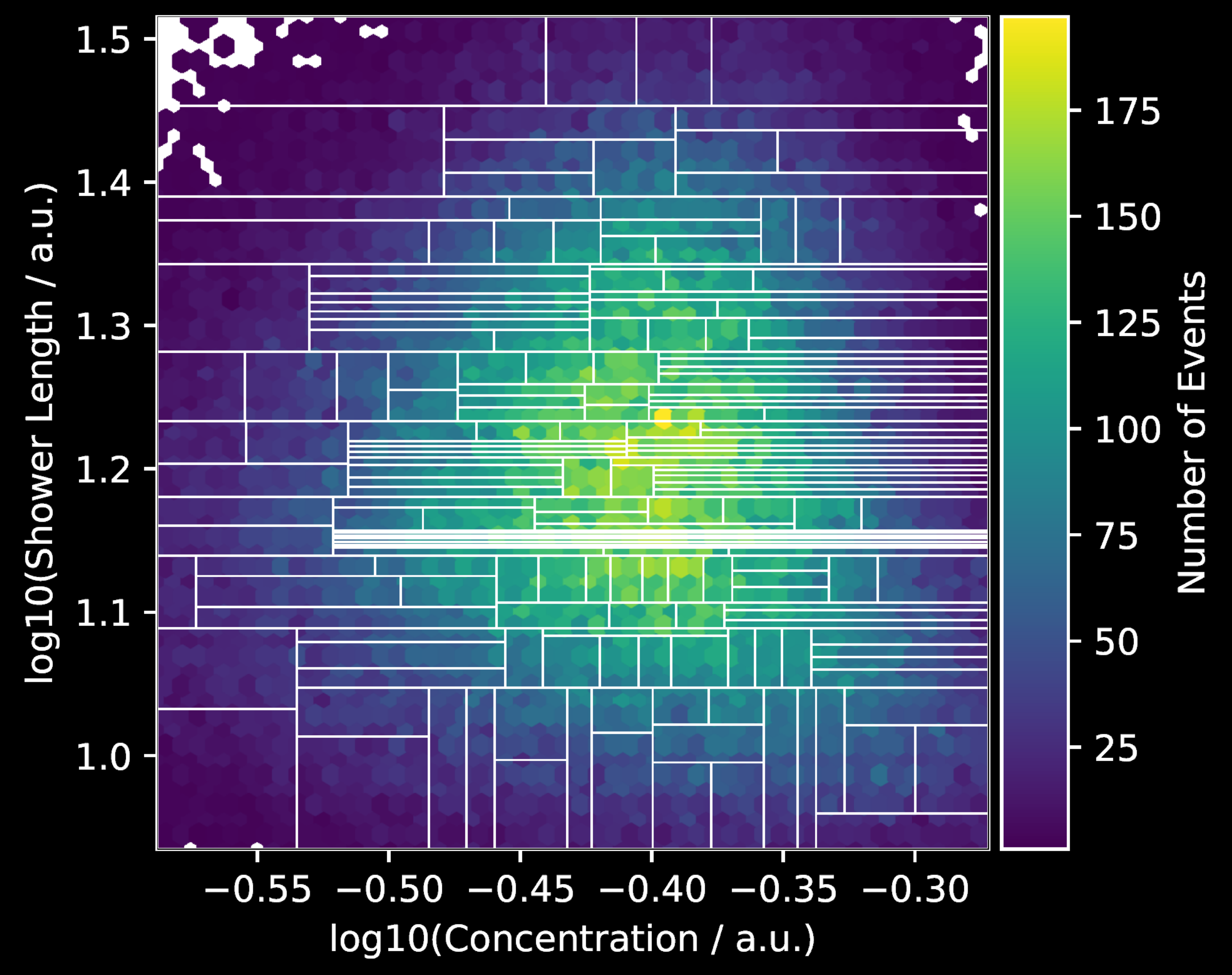
“Classical” Machine Learning

- **Classification (and some regression) tasks:**
  - Background suppression, topology classification, energy and uncertainty estimation
  - Tree-based learners, shallow NNs, …
  - Widely adopted and core component of most IceCube analyses
- **Analysis methodology:**
  - Description of PDFs used in analyses (via KDEs for example)
  - Decision tree binning (Use decision tree to bin high-dimensional parameter space)
  - Iterative unfolding methods with an ML classifier as core component

M. Meier



M. Börner



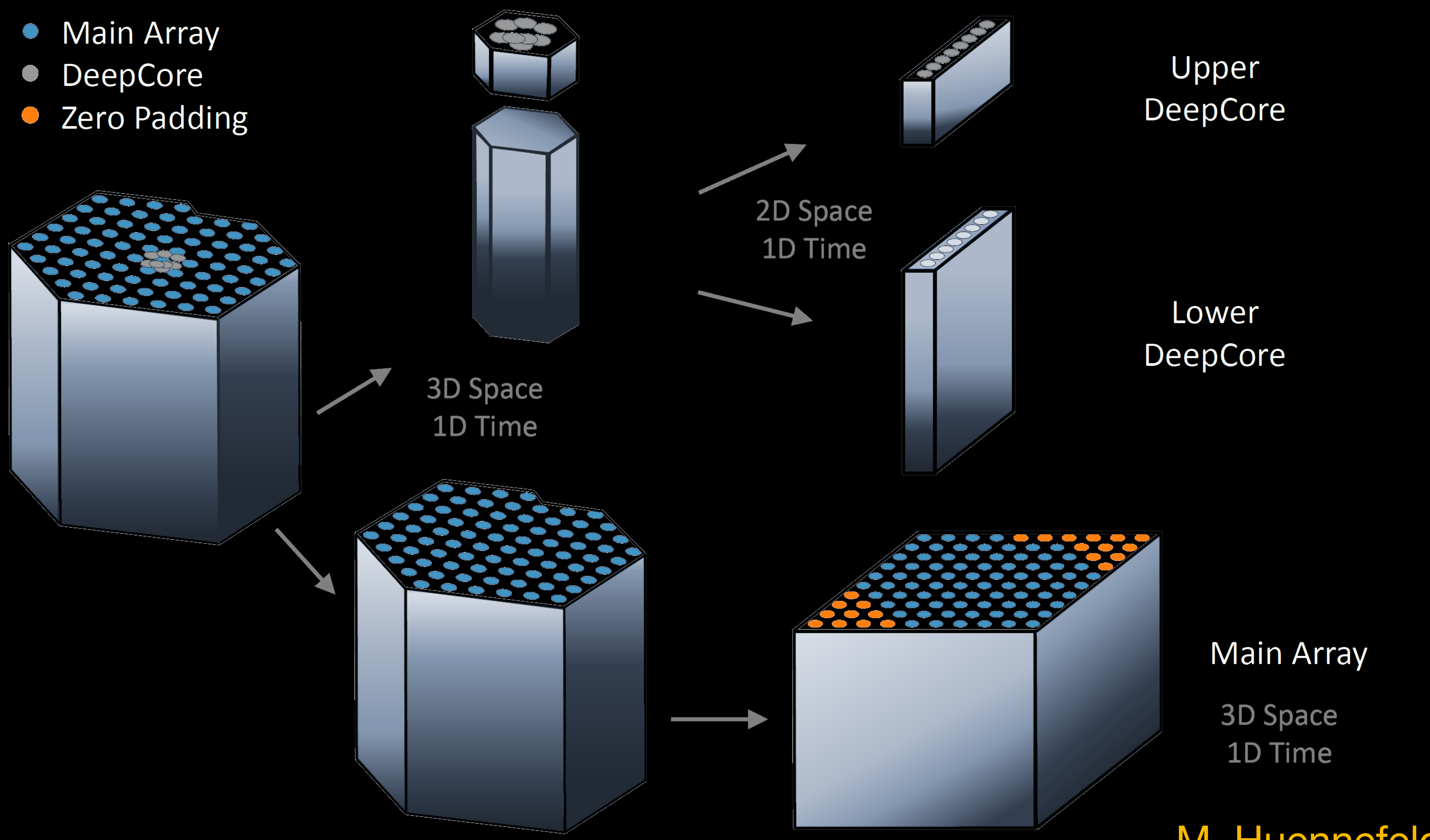
Deep Learning in IceCube

- **What data representation to use?**
  - Tradeoff between curse of dimensionality and information loss
  - Does representation reflect symmetries in data?
- **What type of NN architecture to use?**
  - Is the architecture suited to the data?
  - Can the architecture exploit symmetries in data?
- **How to exploit domain knowledge?**
  - Neutrino interactions are invariant under translation in space and time as well as rotation in space
  - Dust impurities, physics laws, …

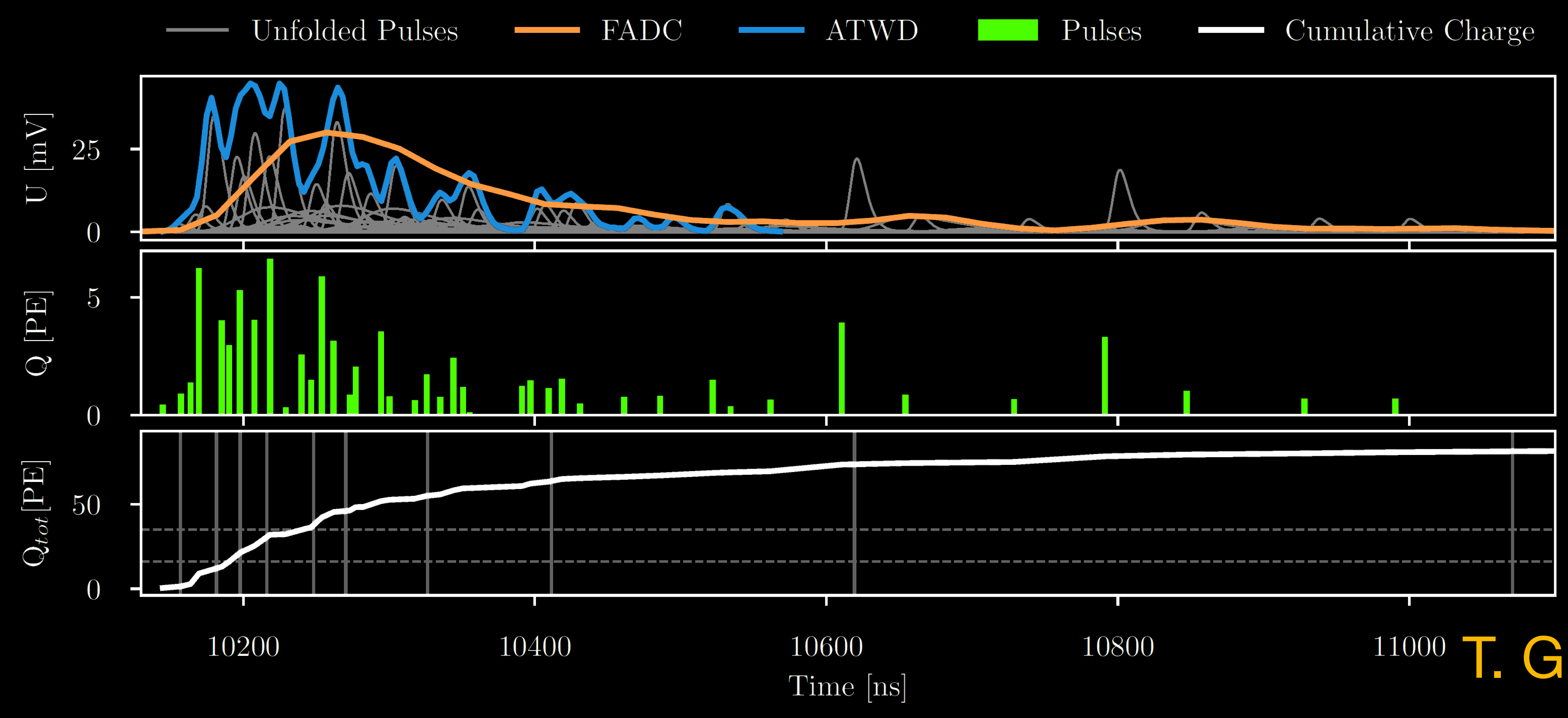
- **Goal:**
  - Find NN architecture suitable for data format that is capable of exploiting symmetries and domain knowledge
- **Architectures Investigated:**
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Graph Neural Network (GNN)
  - Hybrid Maximum-Likelihood Estimation (MLE) / Deep Learning (DL) approaches

Convolutional Neural Networks (CNN)

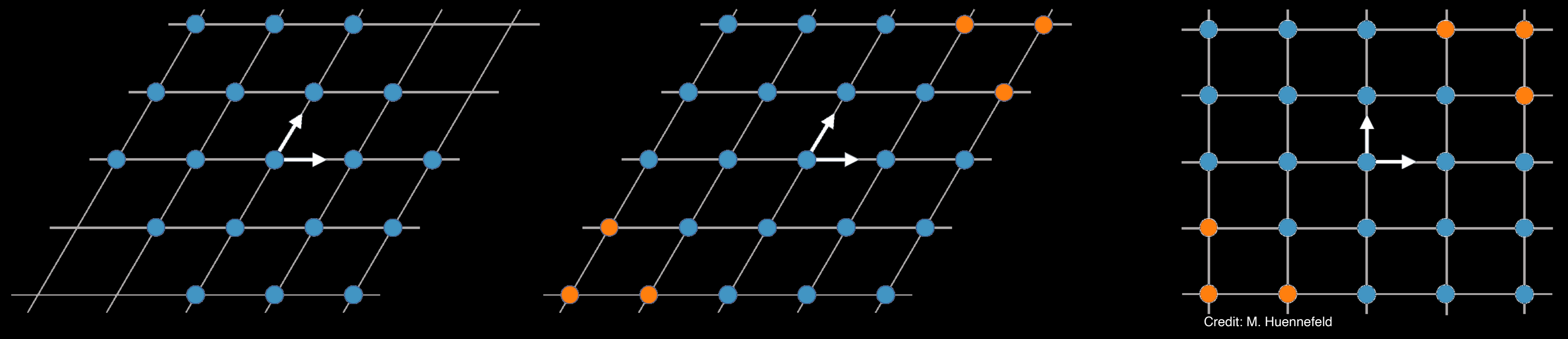
- Require constant and uniform input:
  - Use summary statistics on pulses
- IceCube's geometry needs special handling:
  - Low-energy reconstructions reduce input to DeepCore strings
  - High-energy typically split up detector parts or only use main array



M. Huennefeld



T. Glauch



Hexagonal Convolution Kernels

Convolutional Neural Networks (CNN)

- Speed up reconstruction by orders of magnitude
- Can improve accuracy in comparison to traditional methods
- Uncertainty estimation via Gaussian likelihood as loss
- Many different approaches for various tasks
  - CNN **event reconstruction**
  - CNN event **topology classification**
  - CNNs focused on **low-energy events**

- **Pros/Cons:**

  ⊕ **Exploit (approximate) translational invariance in data**

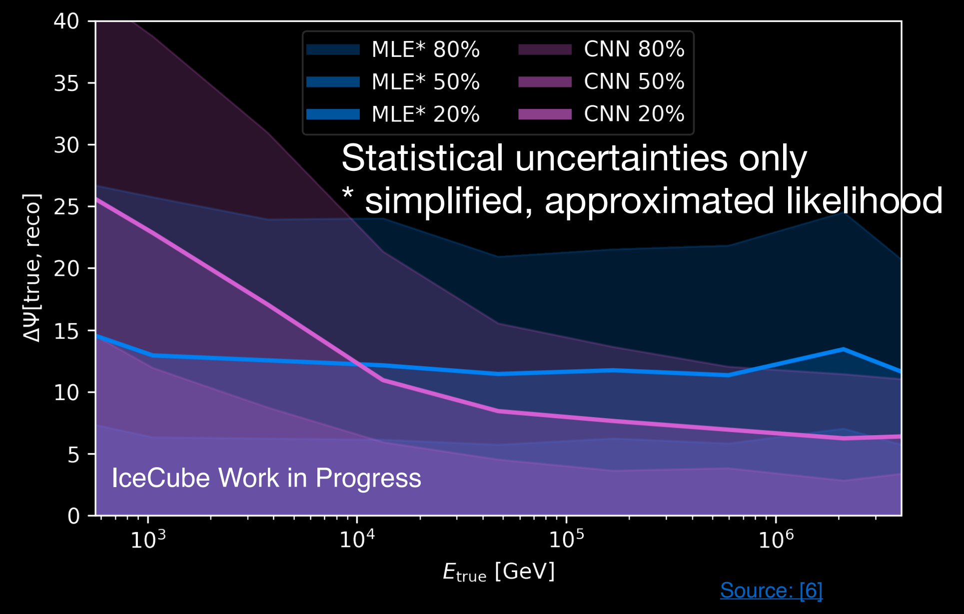
  ⊖ **CNN assumes symmetric grid**

  ⊖ **Cannot naturally account for inhomogeneities in medium**

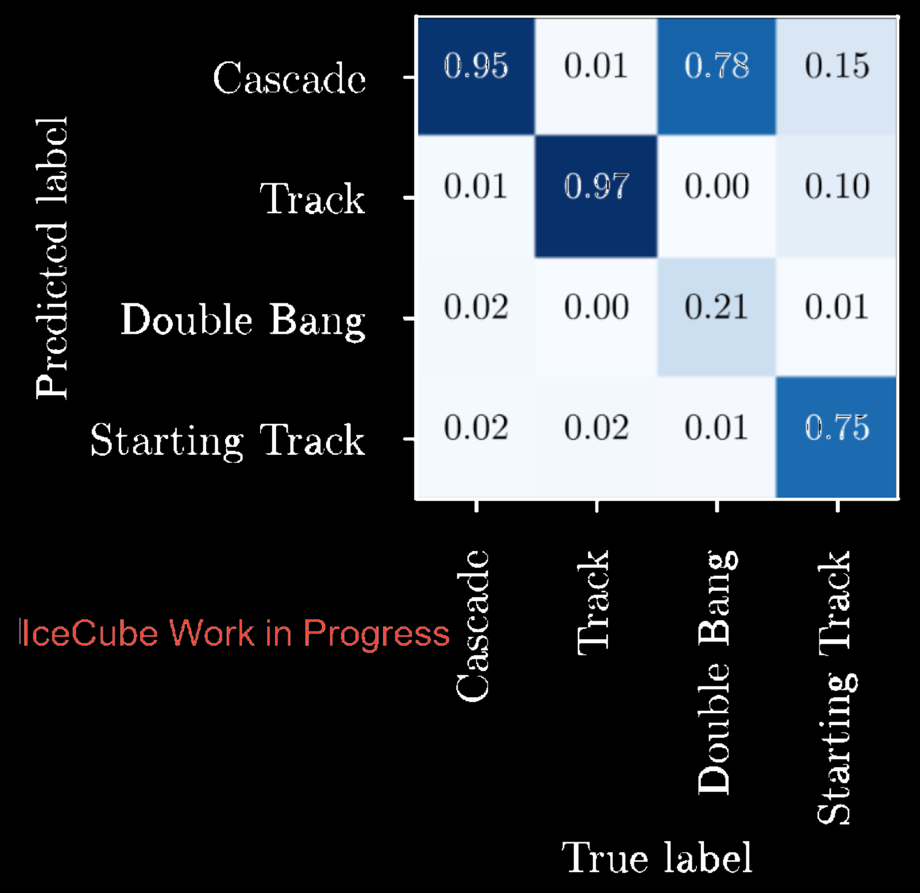
  ⊖ **Loss of information due to summary statistics**

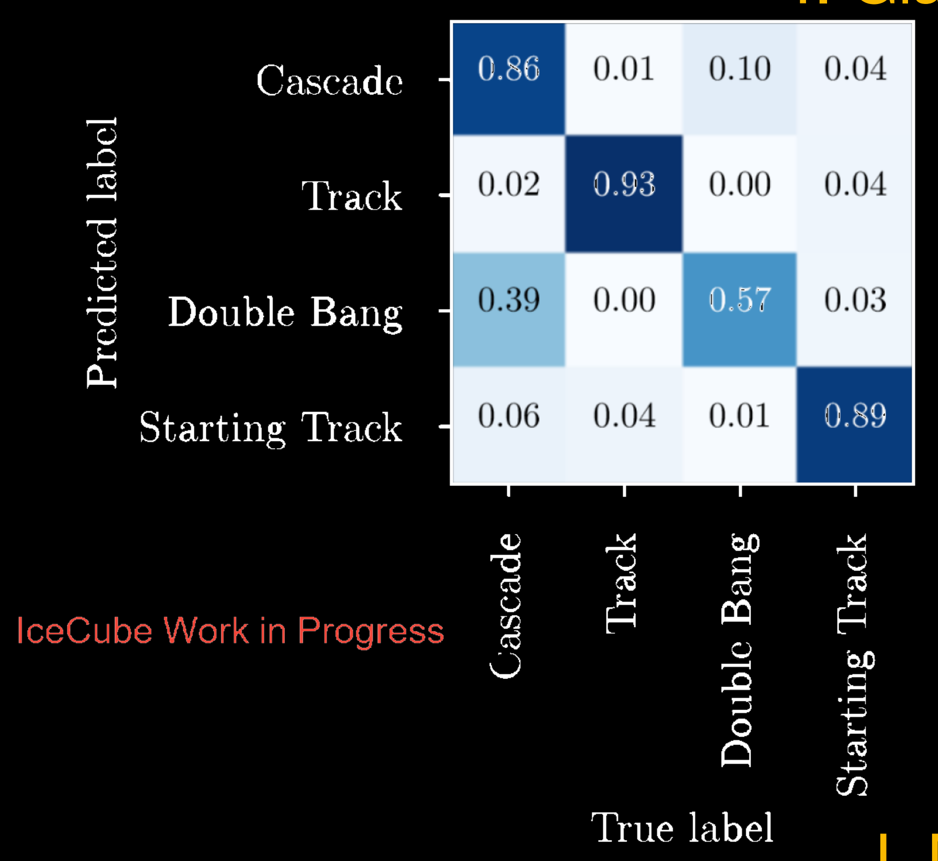
  ⊖ **Inclusion of additional domain knowledge is difficult**

M. Huennefeld

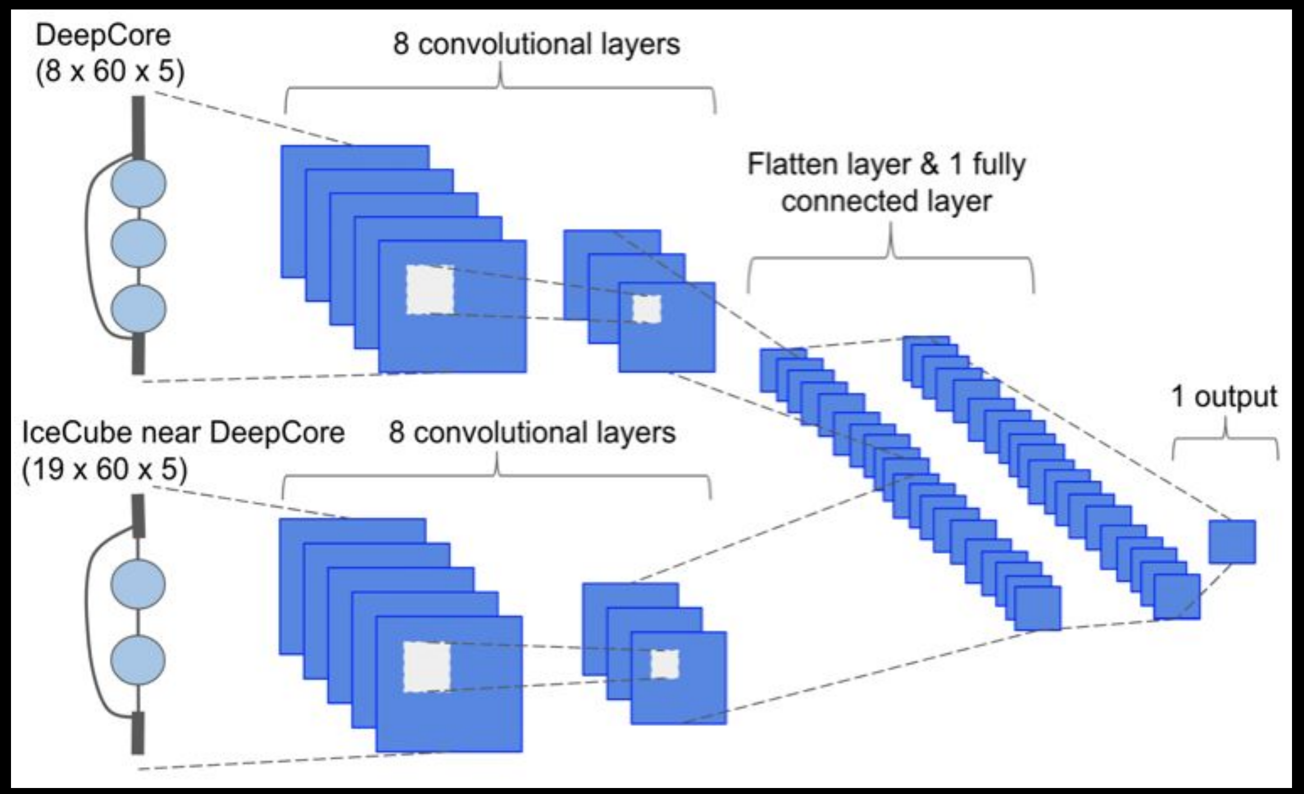


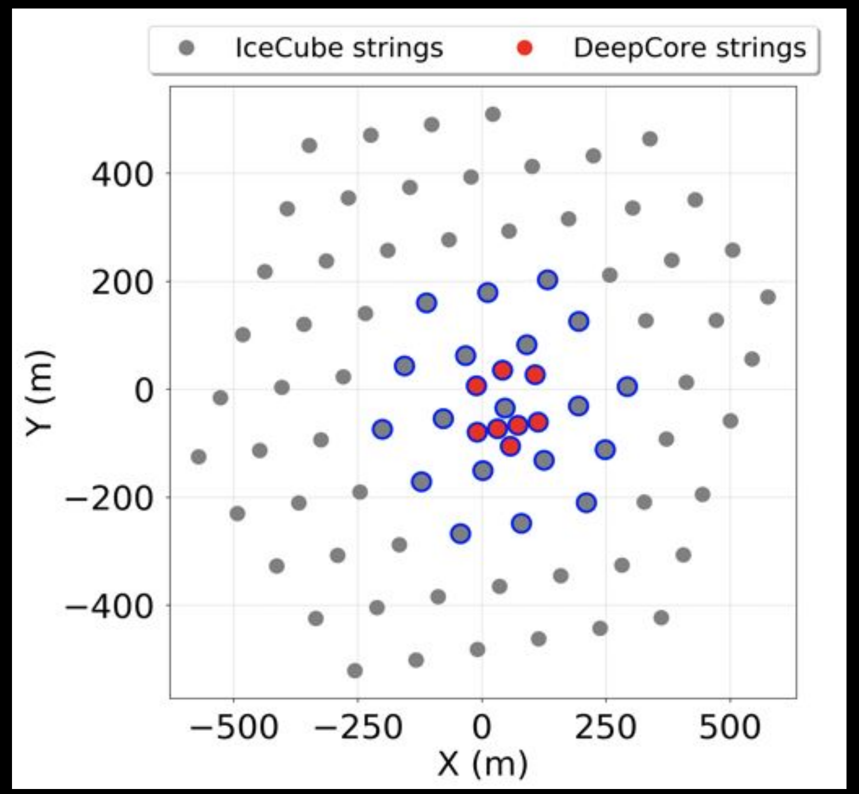
T. Glauch





J. Micallef





Recurrent Neural Networks (RNN)

- RNN can handle arbitrary input length, 2 main approaches:
  - RNN over pulses $(\vec{x}_i,\ q_i,\ t_i)$
  - RNN over event "snapshots" in time
    (CNN over spatial dimension in each time step)
- **Pros/Cons:**

⊕ **Time domain and sequential data is naturally handled**

⊖ **Inclusion of additional domain knowledge is difficult**

RNN over pulses $(\vec{x}_i,\ q_i,\ t_i)$:

⊕ **Can handle inhomogeneities in detector grid and detector medium**

⊖ **(Approximate) translational invariance in data not used**

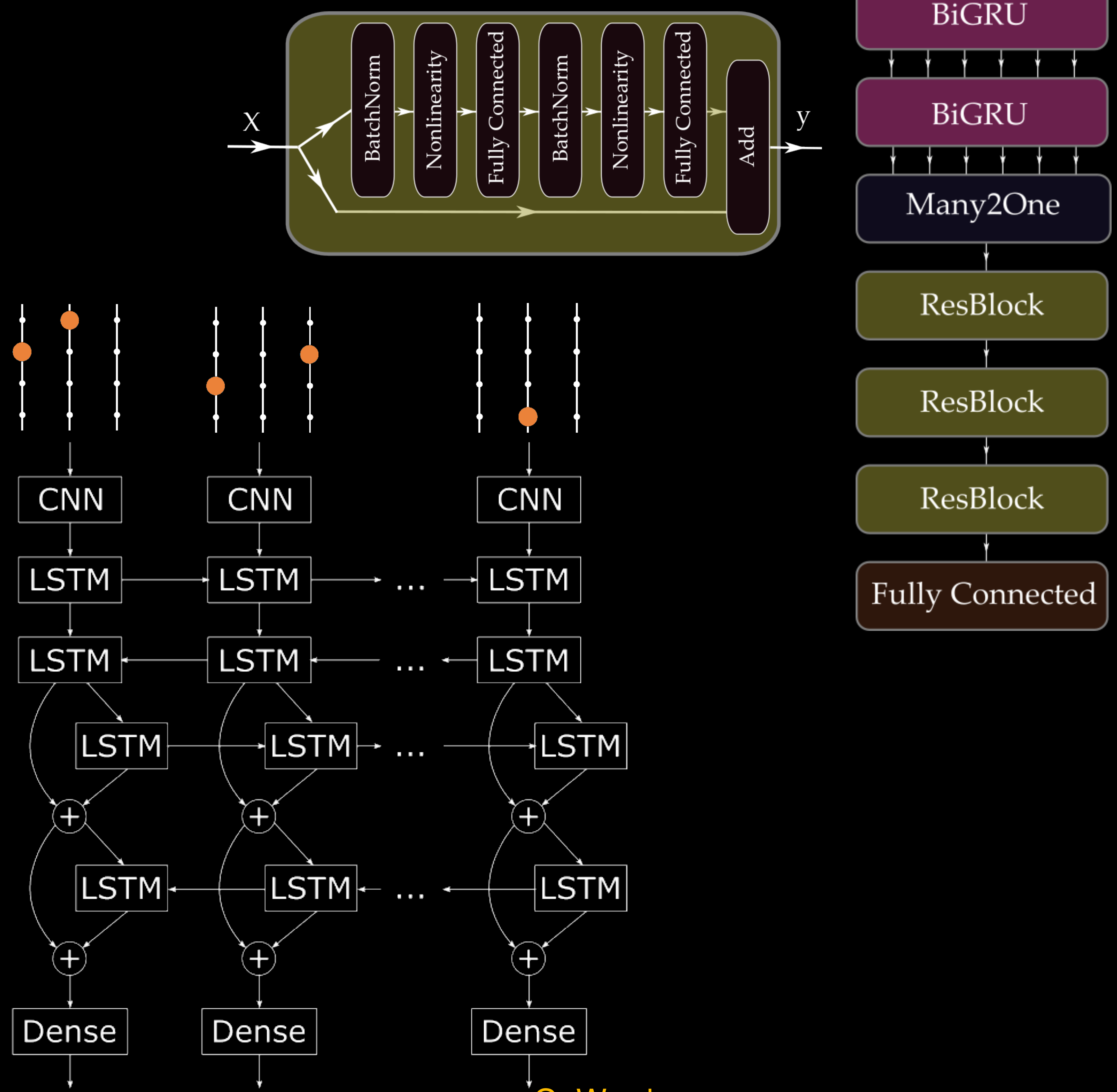
⊖ **Required run-time scales poorly with increased energy**

RNN over event "snapshots" in time (RNN + CNN):

⊕ **Exploits (approximate) translational invariance in data**

⊖ **Cannot naturally account for inhomogeneities in detector medium**

⊖ **Loss of information due to binning**



G. Wrede

Graph Neural Network (GNN)

- Graph convolutions and static graphs
  - DOM-based GNN / Pulse-based GNN
- **Pros/Cons:**

  ⊕ **GNN can handle arbitrary detector grid**

  ⊕ **Graph convolutions enable use of translational symmetries**

  ⊖ **Inhomogeneities in detector medium are not naturally handled**

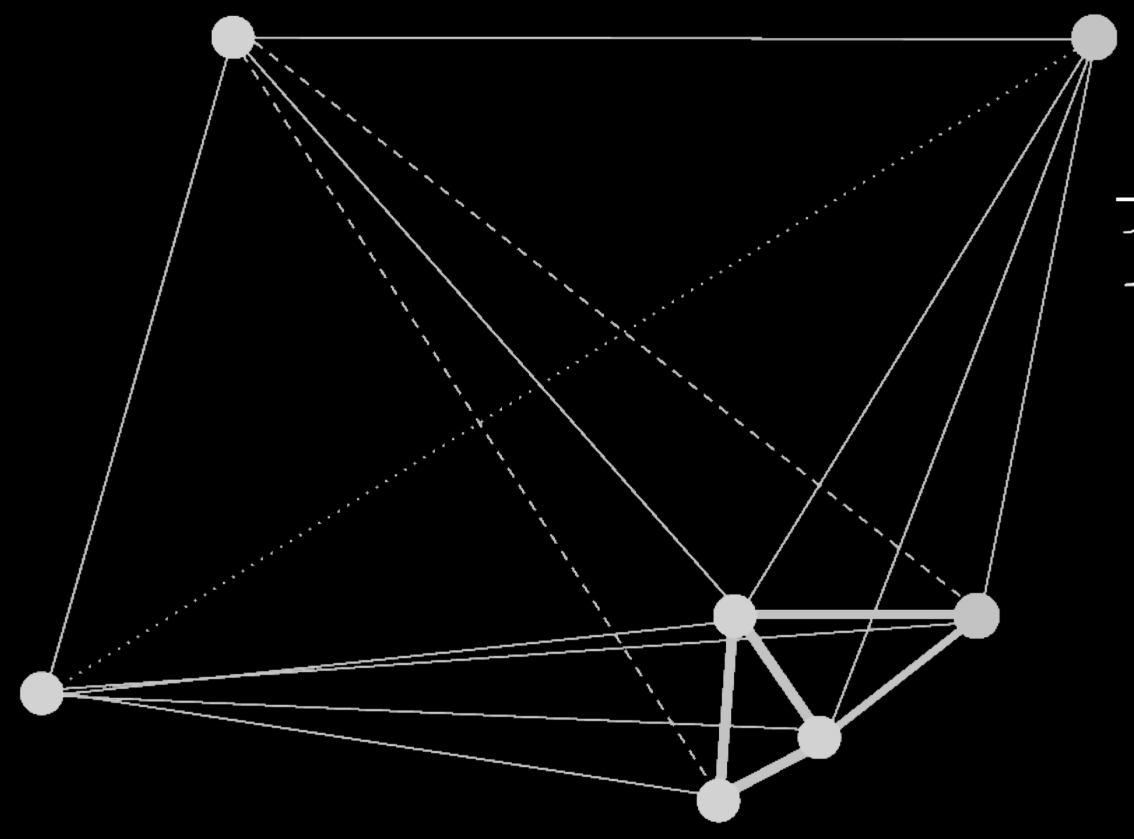
  ⊖ **Inclusion of additional domain knowledge is difficult**

  DOM-based:

  ⊖ **Does not scale well with increasing number of pulses**

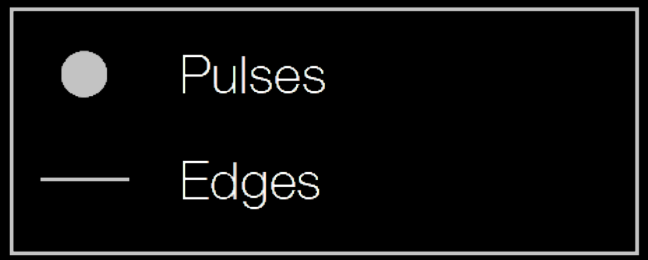
  Pulse-based:

  ⊖ **Loss of information due to summary statistics per DOM**

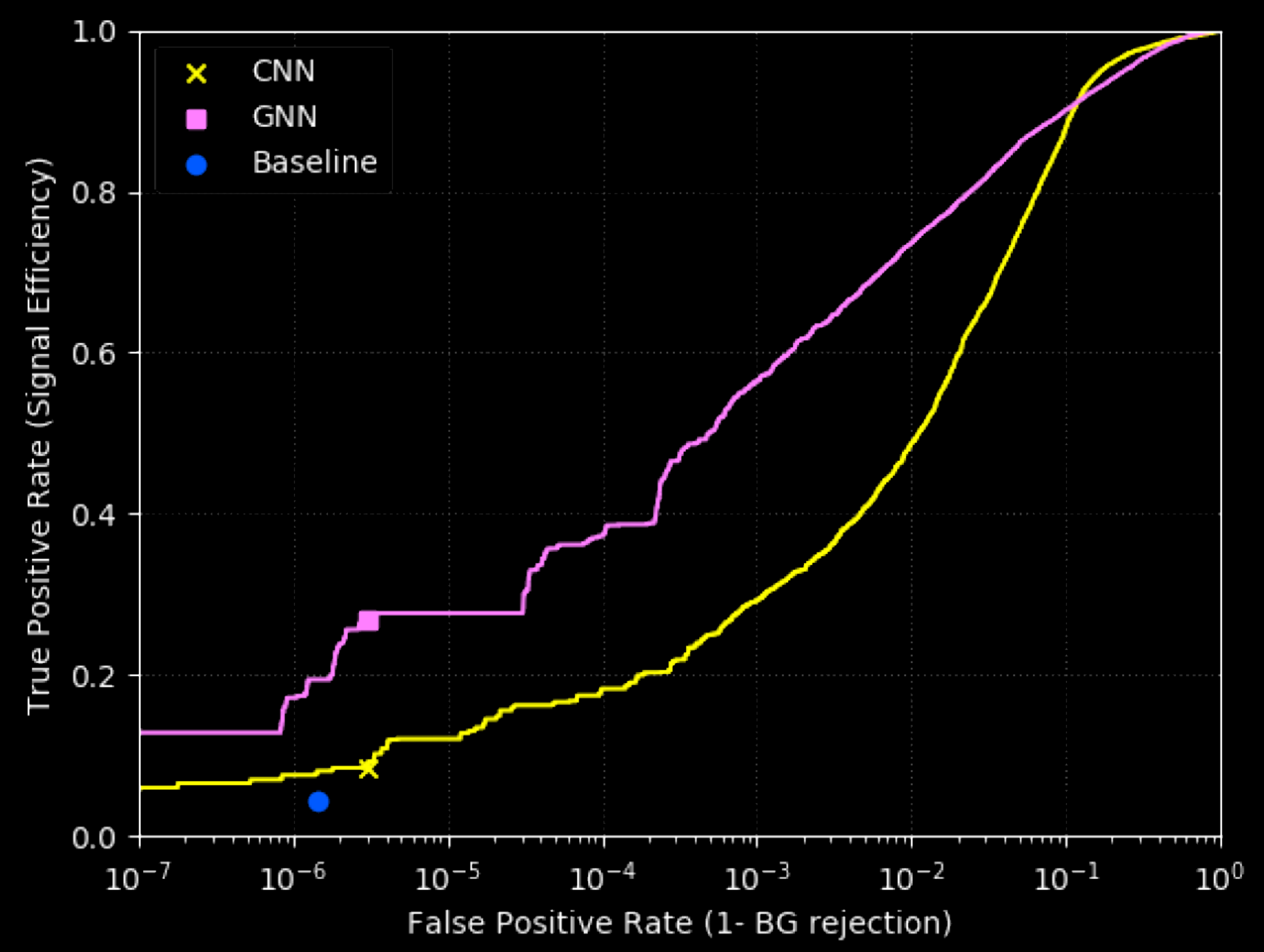


$$\frac{1}{1+d_{xyzct}^2}\cdot sgn(\Delta t)$$

$$d_{xyzct} = \left\| \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \\ ct_2 - ct_1 \end{pmatrix} \right\|$$



M. Ha Minh



Hybrid MLE/DL Methods

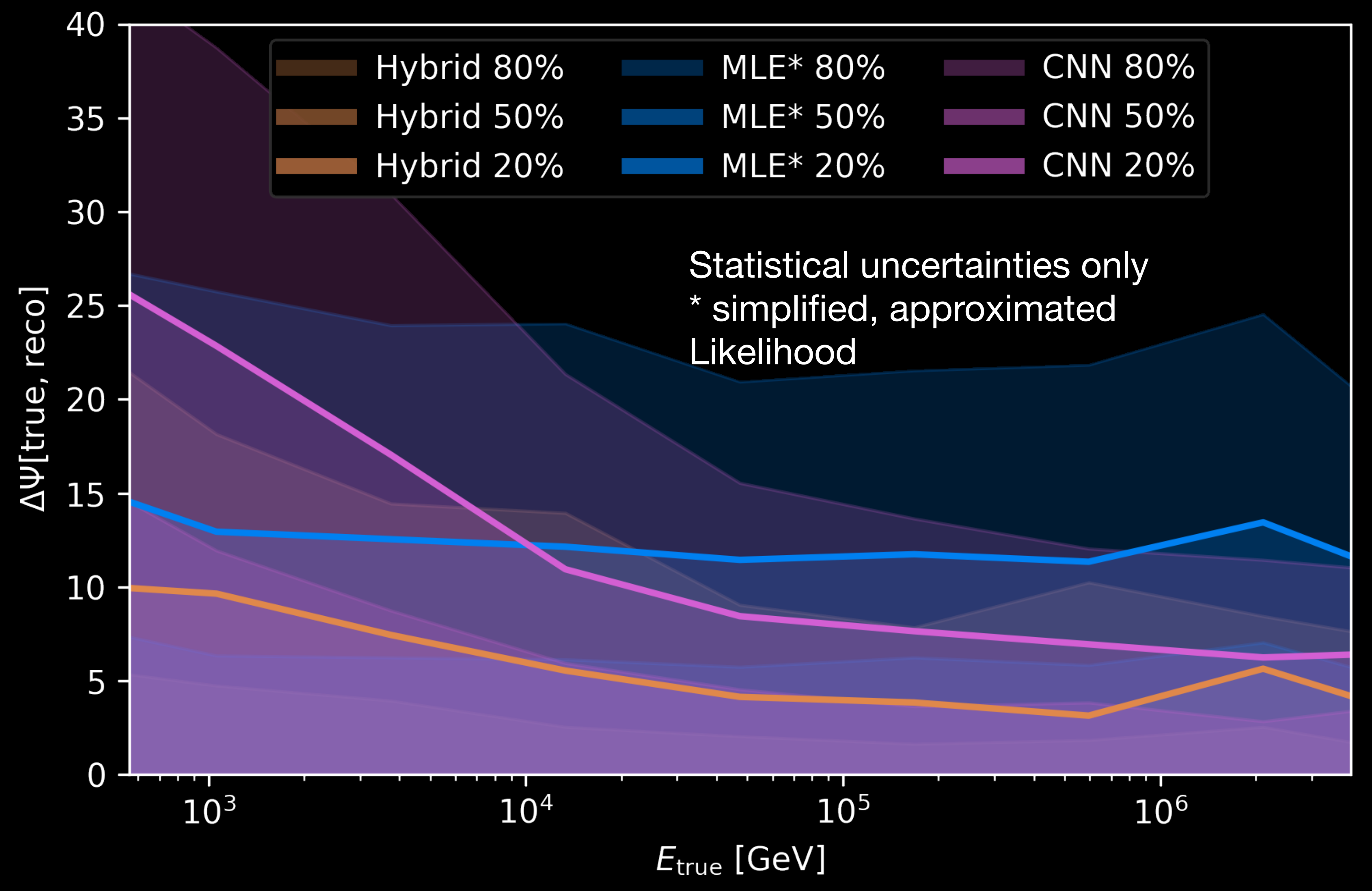
- Methods that **combine ML and "traditional" likelihood-based methods**, e.g.: Generative NN to explicitly approximate Likelihood:
  - Generative NN to approximate arrival time PDF $p(x_i \,|\, \vec{\theta}\,)$ and total expected charge $\vec{\lambda}$ at each DOM
  - Fully-differentiable approximation of MC simulation: gradients and Hessian available for optimization of $\mathscr{L}$ and uncertainty estimation
- Also: work on likelihood-free inference
- **Pros/Cons:**

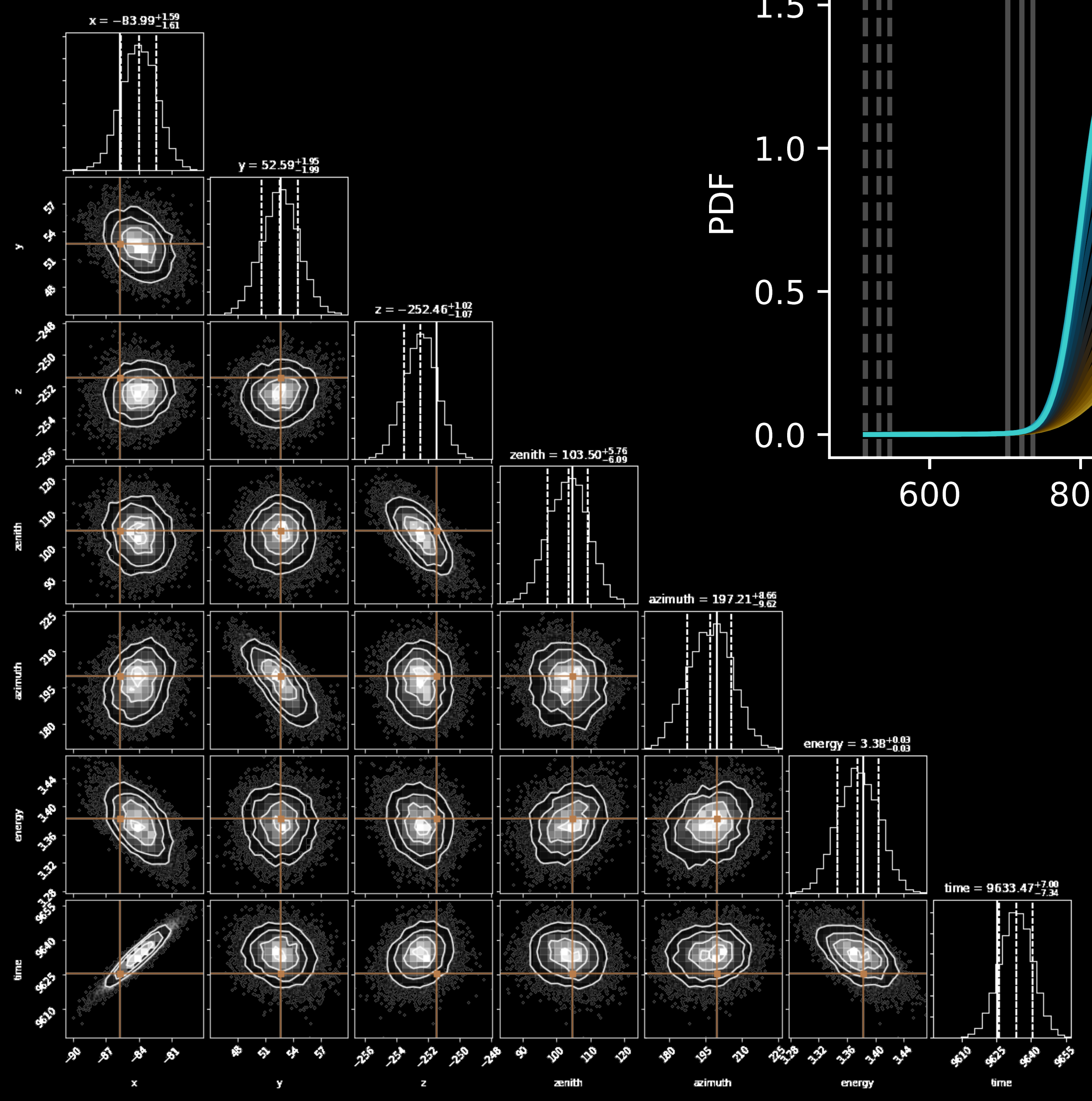
⊕ **Symmetries and domain knowledge can easily be included**

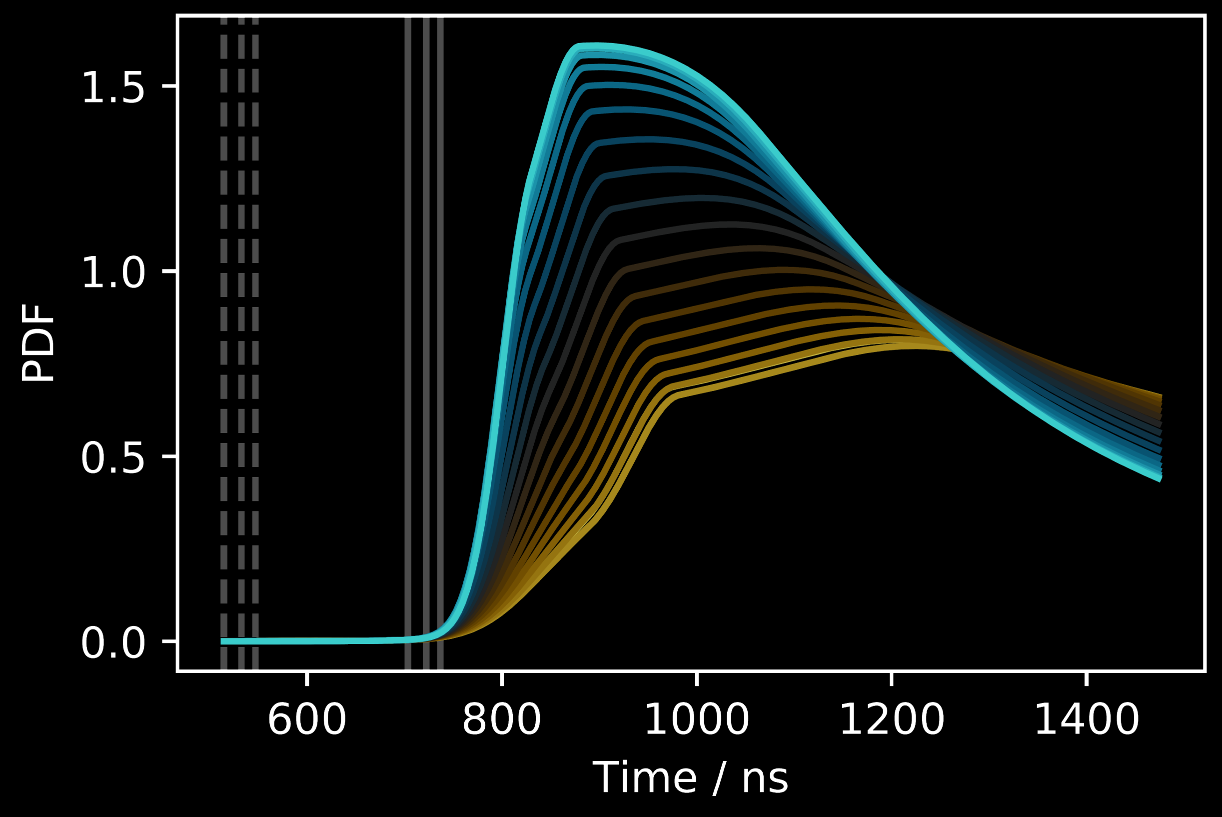
⊕ **Use of complete event information (pulses/waveforms)**

⊕ **Supports arbitrary detector geometry**

⊖ **Increased reconstruction time due to optimization of $\mathscr{L}$**







M. Huennefeld

Other Directions

- Time-Convolutional Networks / Transformer-based networks
- Unsupervised Learning
- many, many more …

Infrastructure 1/2

New ML-based project pop up (almost) once a month!

- We have GPUs available in IceCube, so some of the infrastructure for training exists
- However, larger-scale datasets can be hard to read/convert into formats suitable for training
  - How do you feed 100s of millions of events to your training algorithm efficiently?
  - Maybe you can even simulate these “just in time”?
  - Event-server architectures?

Infrastructure 2/2

New ML-based project pop up (almost) once a month!

- We still need to work on integrating technologies for solutions such as cloud-based accelerators (e.g. TPUs)
- Some of the existing architectures run into issues with training performance - we need specific infrastructure supporting long-running (and potentially parallel) training jobs. Sometimes you need days to weeks of run time.
    - How do we support cloud-based solutions for ML developers?
    - How will they get their training data to the infrastructure?
- Running containerized training tools (as you would in industry) can be sometimes hard/tedious on our infrastructure (Singularity vs. Docker/Podman)
    - There is a whole ecosystem of container-based ML tools which we could potentially use - we should evaluate our options

Summary

- Deep Learning Approaches already approved / deployed:
  - CNN based cascade reconstruction
  - CNN based classification for cascade real-time stream **(we just sent a cascade HESE alert!)**
- Coming up next:
  - low-energy CNN (DeepCore event optimized)
  - Hybrid MLE/DL method with explicit Likelihood
  - RNN based reconstruction
  - …