technische universität dortmund

IceCube South Pole Neutrino Observatory

SFB 876 Providing Information by Resource-Constrained Data Analysis

lehrstuhl physik e5

Exploitation of Symmetries and Domain Knowledge in Deep Learning Architectures

Mirco Huennefeld
mirco.huennefeld@tu-dortmund.de

Workshop on ML for Cosmic-Ray Air Showers
February 1st, 2022

How can we best utilize available information?

Talk Outline

My personal view on Deep Learning

- Deep learning as a tool for function approximation
- Utilization of domain knowledge and symmetries

"Classical" Deep Learning in IceCube

- Choice of data representation and NN architecture
- Convolutional neural networks

Hybrid MLE/DL Method

- Combining maximum-likelihood with Deep Learning

Conclusions and Outlook

Talk Outline

My personal view on Deep Learning

- Deep learning as a tool for function approximation
- Utilization of domain knowledge and symmetries

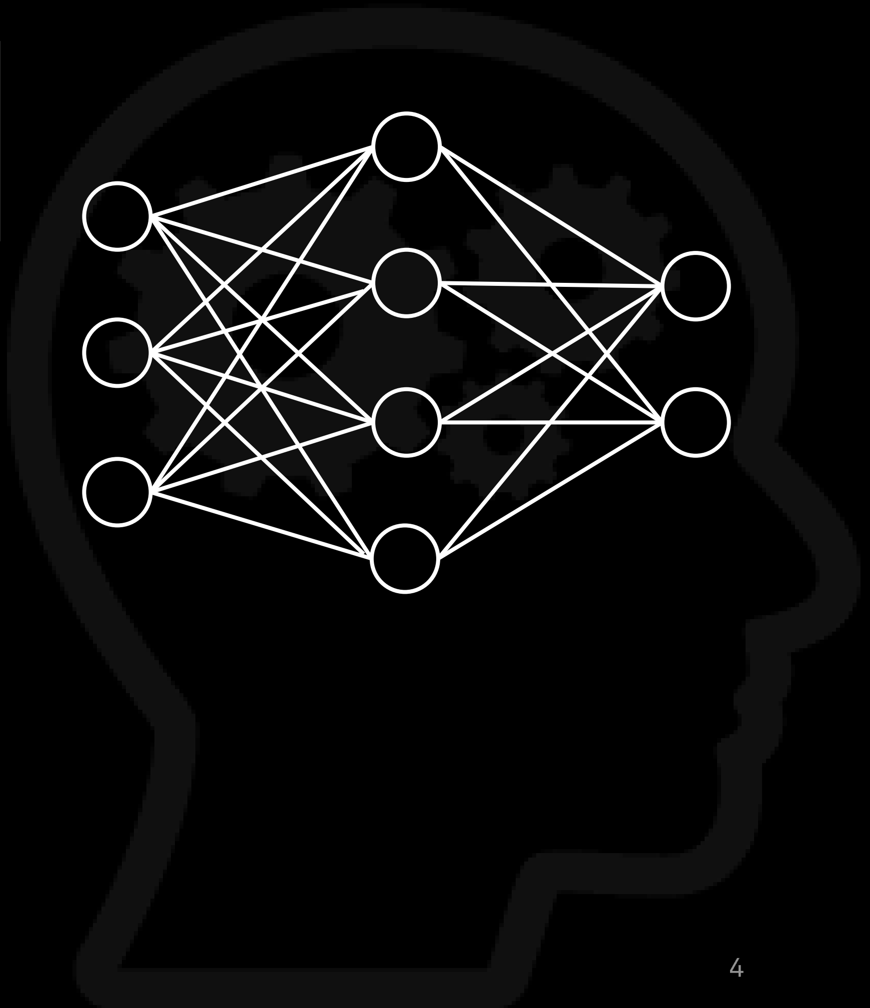
"Classical" Deep Learning in IceCube

- Choice of data representation and NN architecture
- Convolutional neural networks

Hybrid MLE/DL Method

- Combining maximum-likelihood with Deep Learning

Conclusions and Outlook



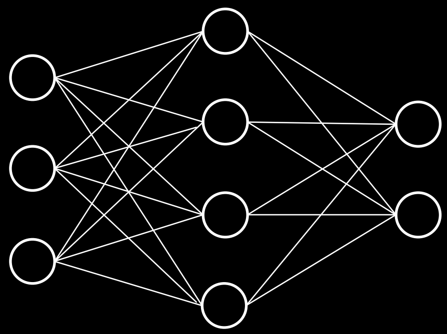
My personal view on Deep Learning (DL)

DL is a powerful tool for function approximation

- Not "intelligent": simply maps input to output space
- NN architectures (CNNs, GNNs, RNNs, ...) are just a way to formulate this mapping while exploiting certain properties of the data
- No need for choice of parameterization
- Scales well to high-dimensional space

This is what makes Deep Learning so powerful

$$f : I \rightarrow O$$



Importance of symmetries & domain knowledge

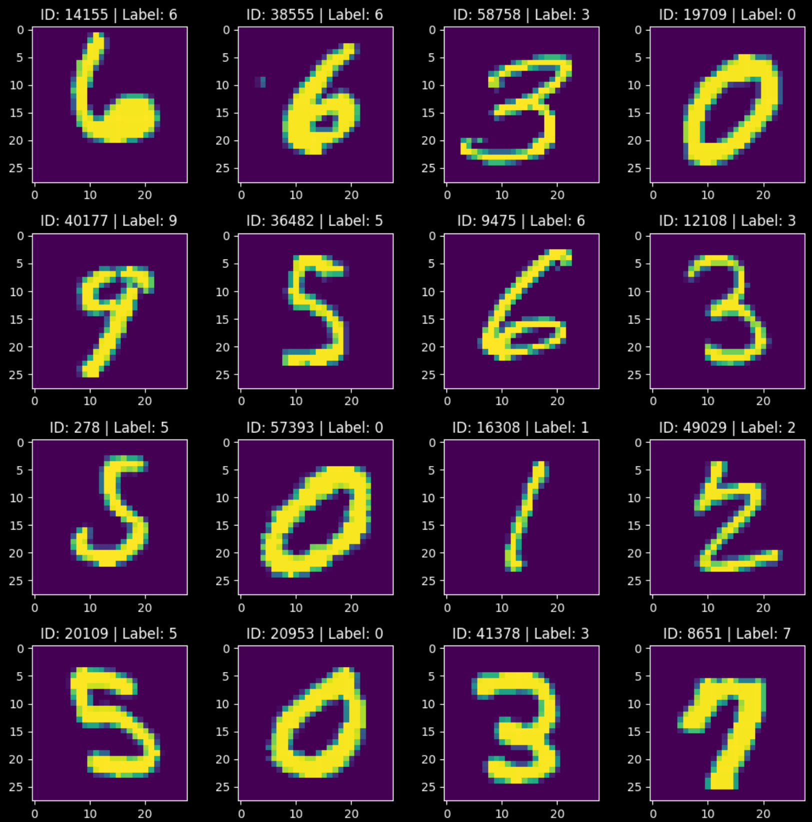
- Necessary to reduce free parameters
- Improves model training and performance
- Generalization/extrapolation is only possible along symmetries or by using domain knowledge

Unique position of Physics in ML:

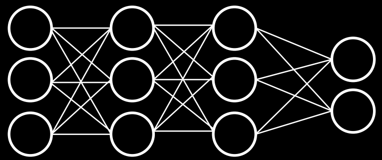
- Data generating process is known extremely well
- Many possible applications: reco/calibration/analysis/...

Need to think outside constraints of typical applications and architectures

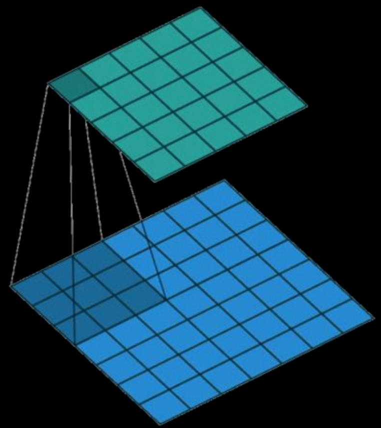
Example 1: Translational Invariance



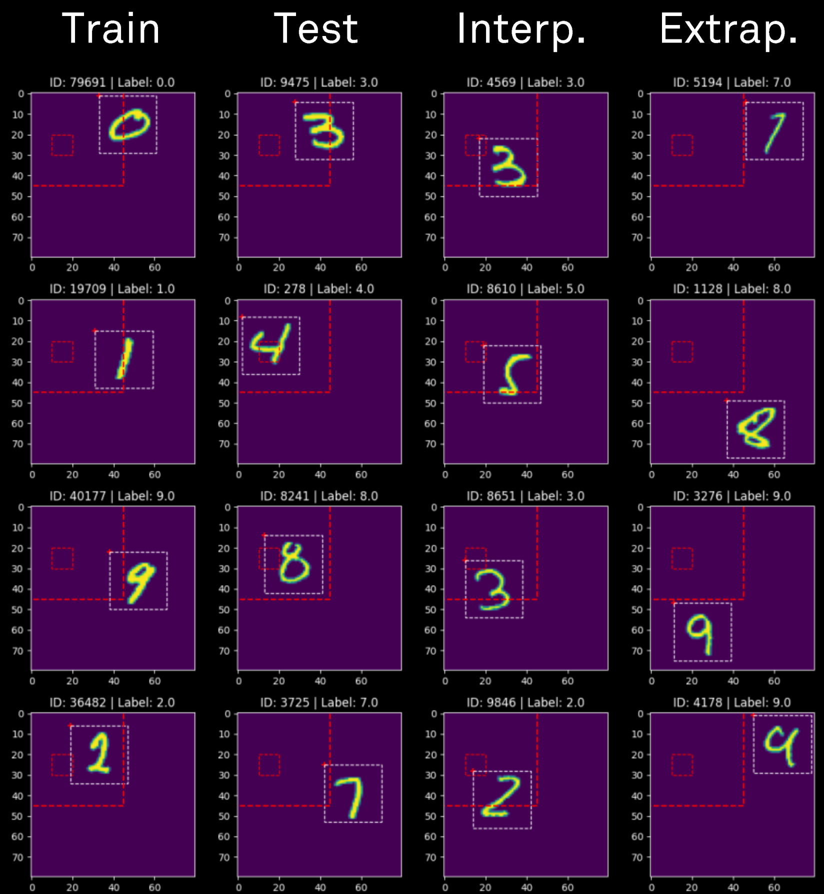
- Classification task with MNIST data
- Compare standard dense vs convolutional neural network (CNN)
- Embed MNIST data in larger images
- Define inter/extrapolation regions



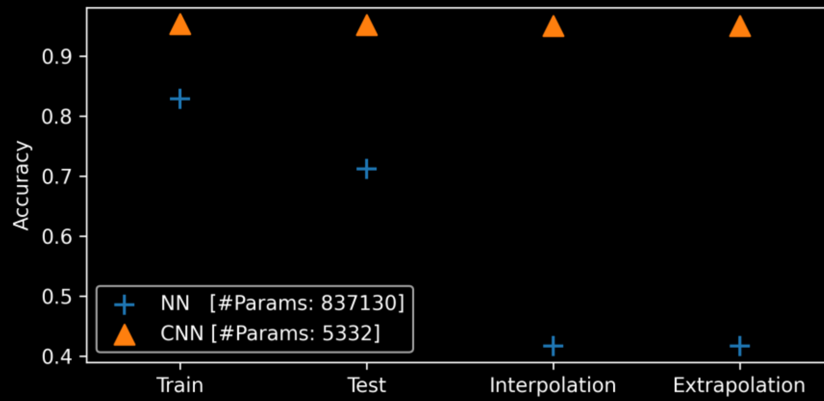
vs



Example 1: Translational Invariance

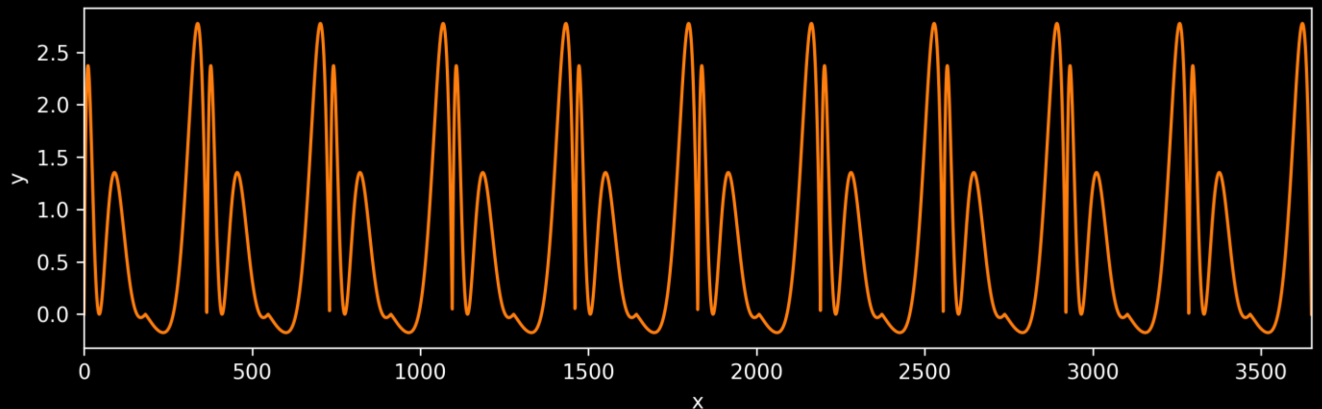


- Embed MNIST data in larger images
- Define inter/extrapolation regions
- Only CNN can generalize: possible via translational invariance

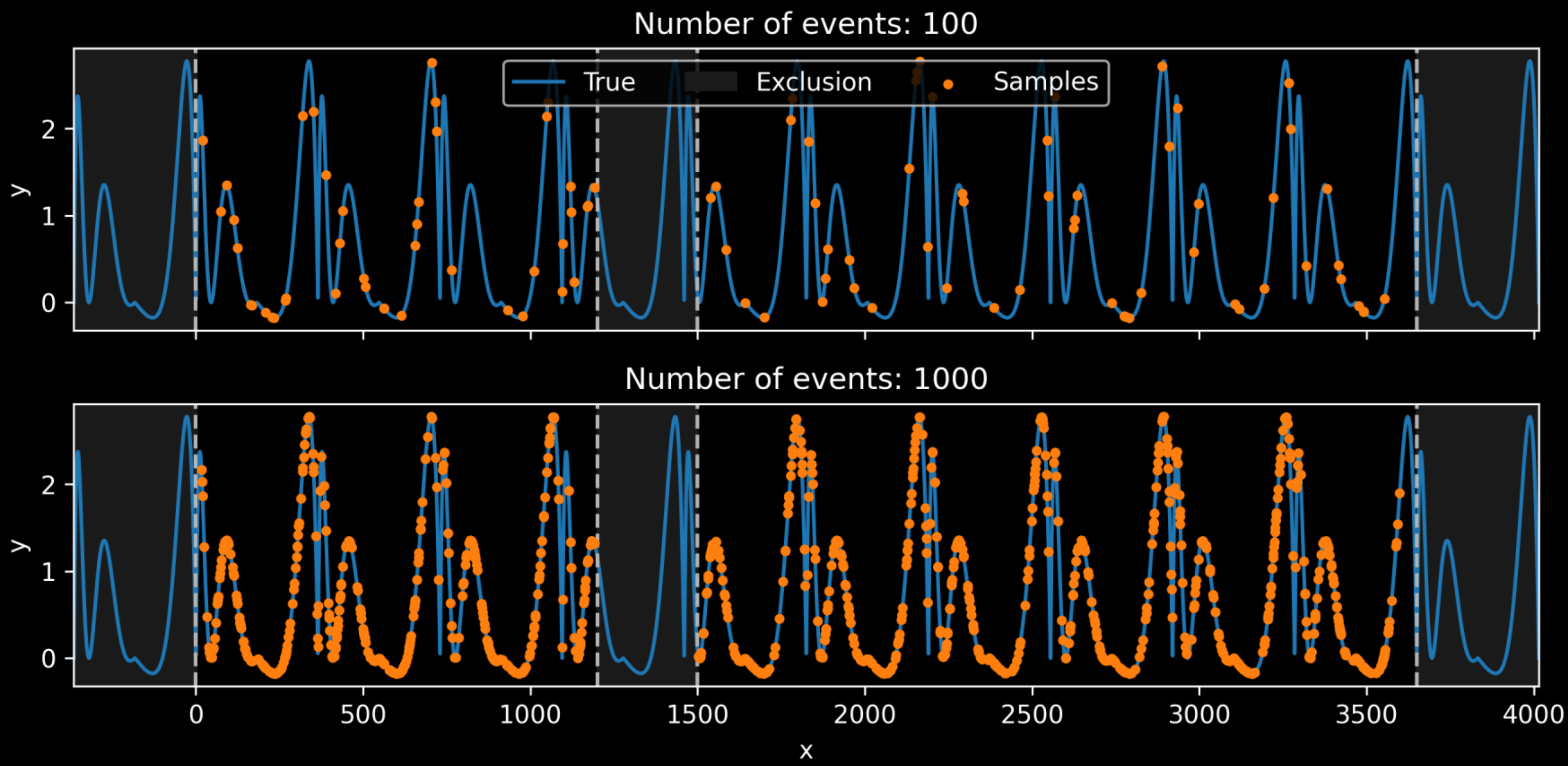


Example 2: Periodicity

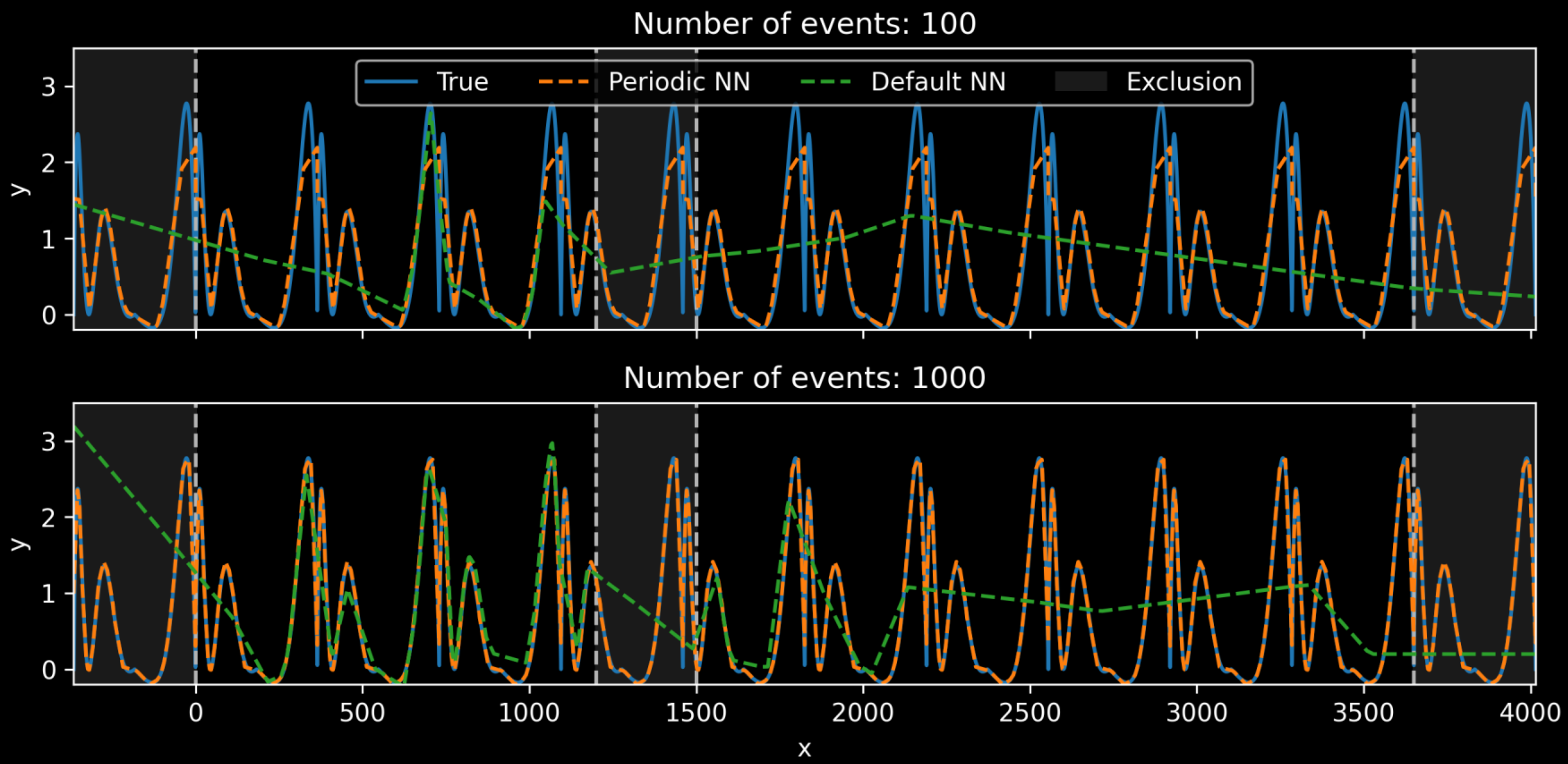
- Assume we want to fit a 1-D function of which we know/are searching for a periodicity with a given period length T
- This info can be embedded directly into NN architecture!
- Compare standard NN vs "periodic" NN: $f(x) = f(x \, mod \, T)$
- Compare different size of training set



Example 2: Periodicity

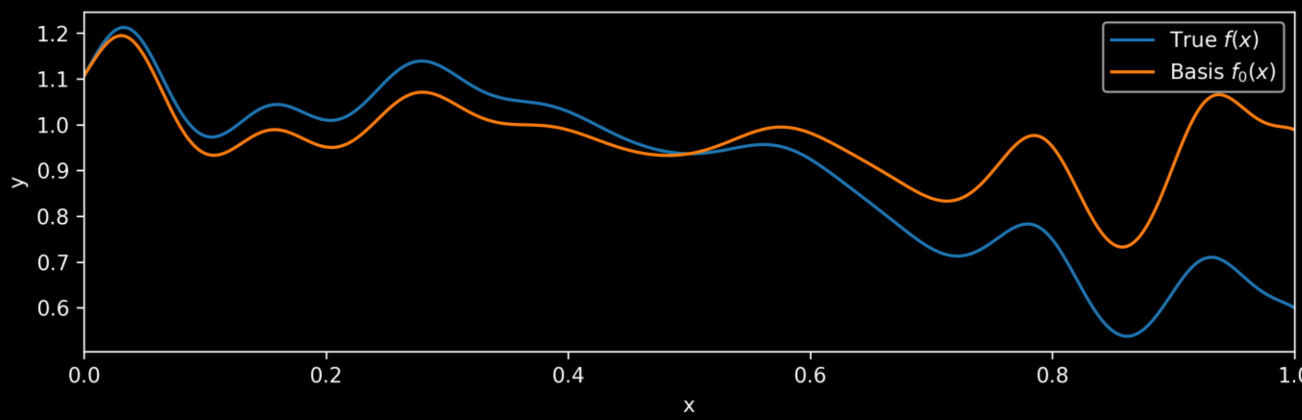


Example 2: Periodicity

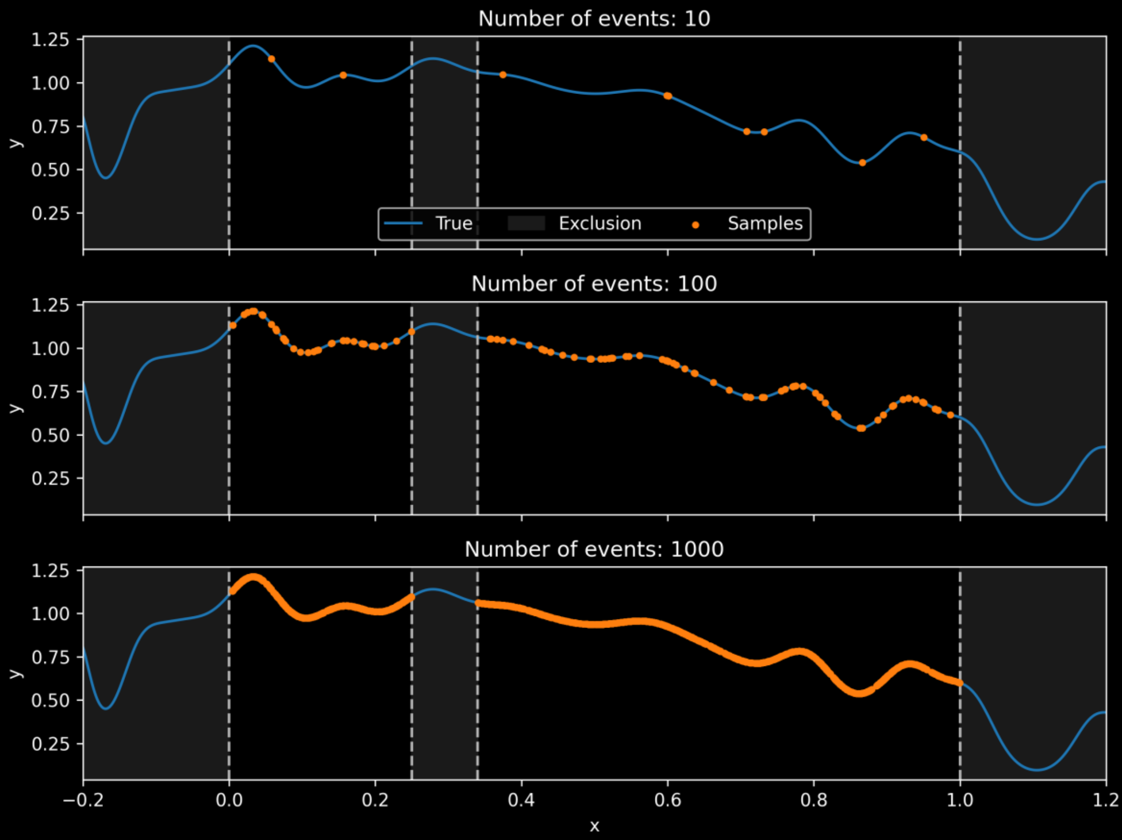


Example 3: known approximate solution

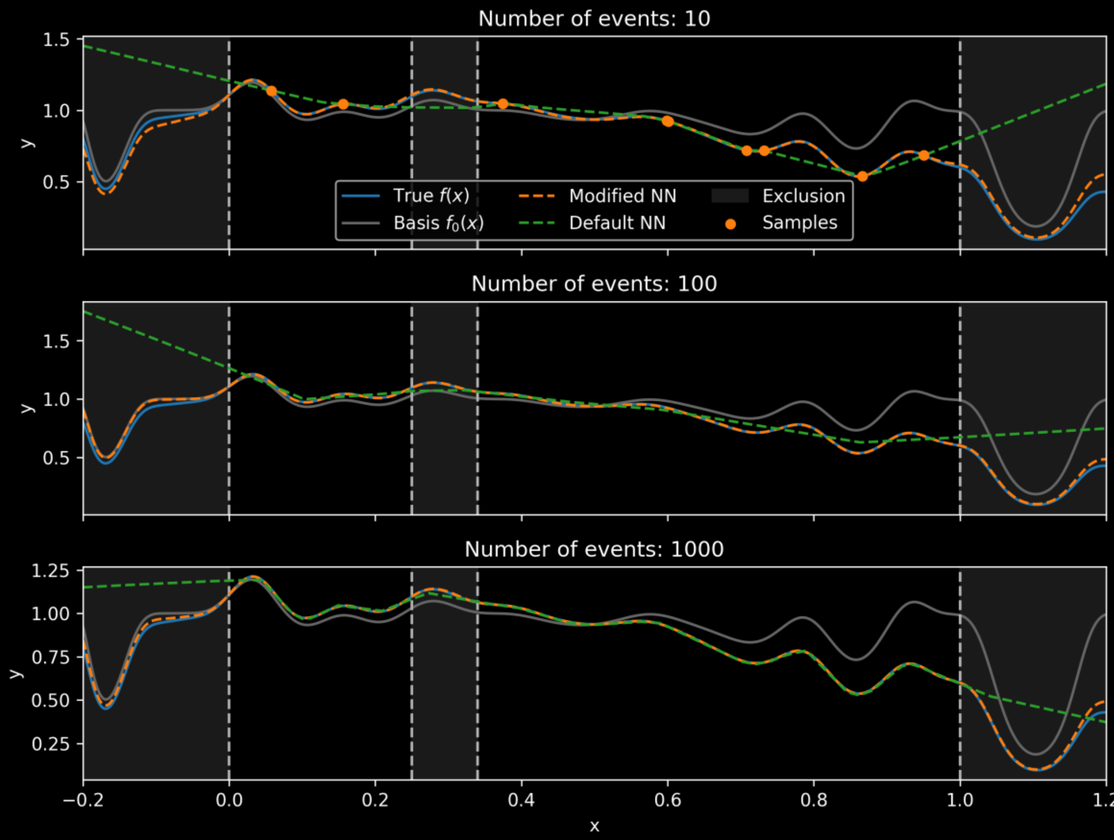
- Assume we want to fit a 1-D function of which we know the approximate shape $f_0(x)$, but true solution deviates slightly
- This info can be embedded directly into NN architecture!
- Compare standard NN vs "modified" NN: $f(x) = f_0(x) \cdot NN(x)$
- Compare different size of training set



Example 3: known approximate solution



Example 3: known approximate solution



Talk Outline

My personal view on Deep Learning

- Deep learning as a tool for function approximation
- Utilization of domain knowledge and symmetries

"Classical" Deep Learning in IceCube

- Choice of data representation and NN architecture
- Convolutional neural networks

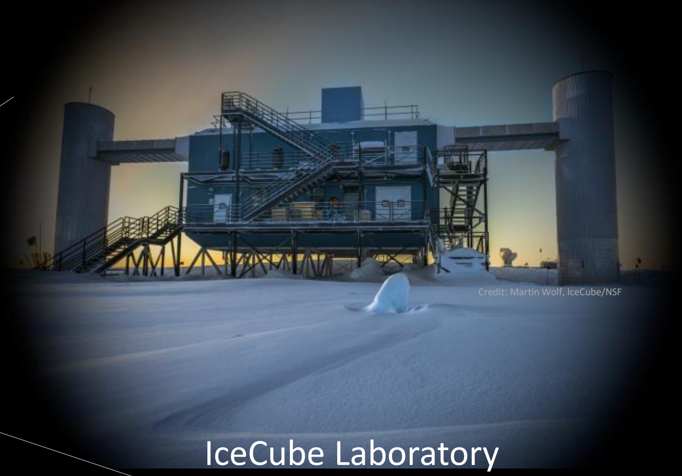
Hybrid MLE/DL Method

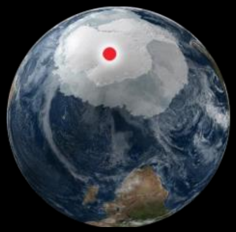
- Combining maximum-likelihood with Deep Learning

Conclusions and Outlook

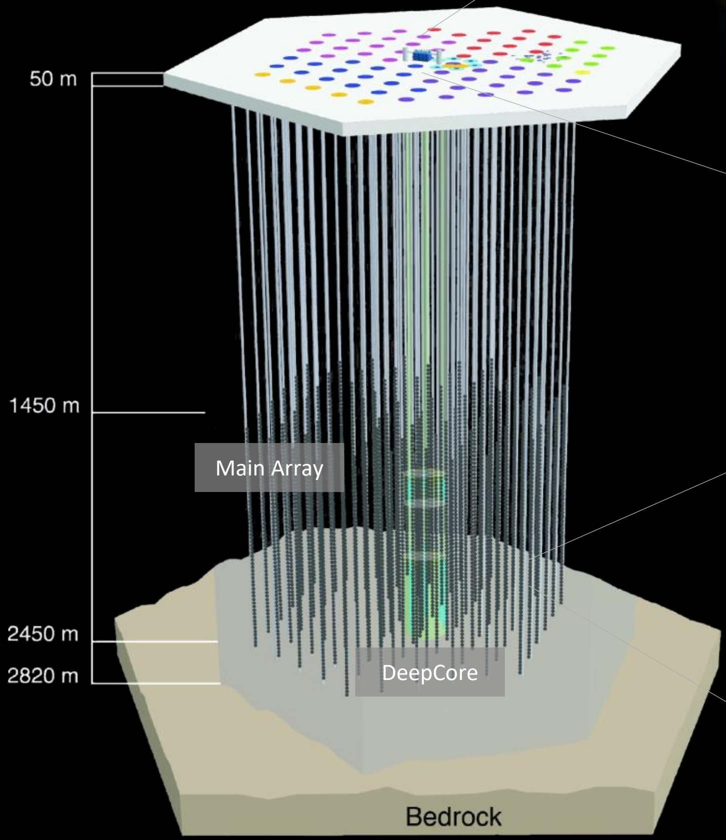
Credit: pngfuel.com
Credit: Felipe Pedreros, IceCube/NSF

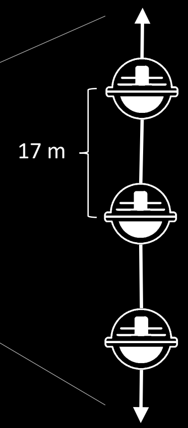
ICECUBE
SOUTH POLE NEUTRINO OBSERVATORY

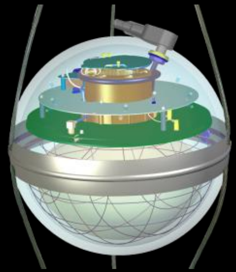




Amundsen-Scott South Pole Station, Antarctica



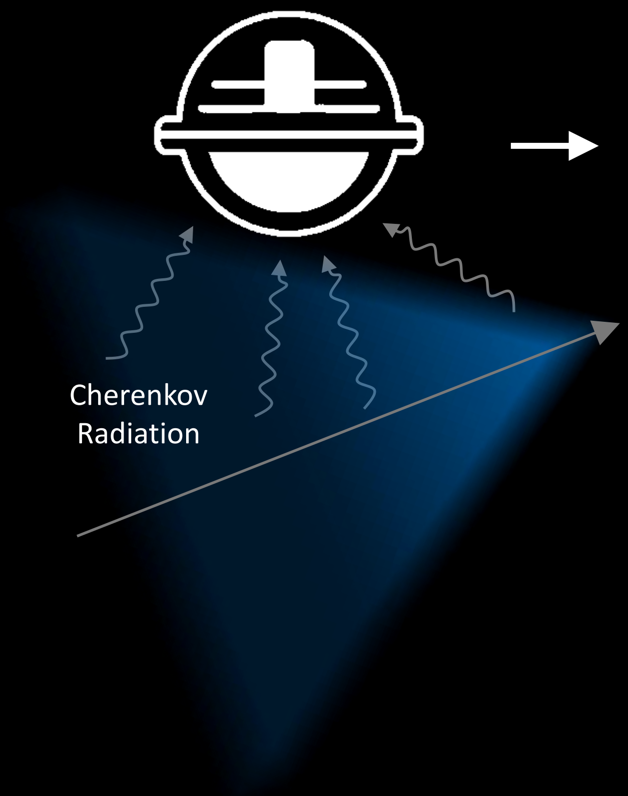


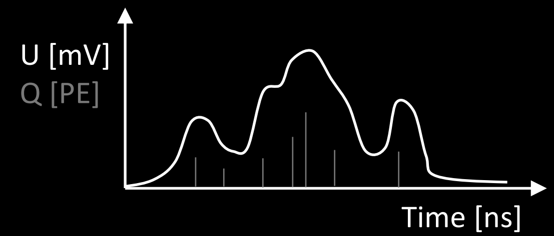


5160 Digital Optical Modules (DOMs)

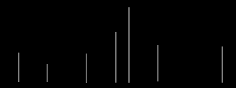
86 Strings:
78 Main Array
8 DeepCore

Detection Mechanism

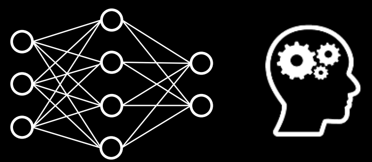




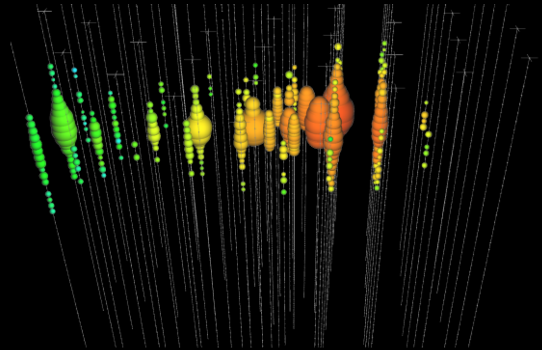
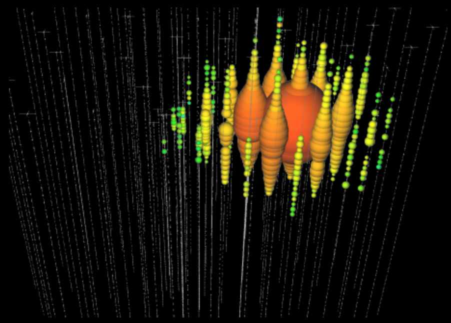
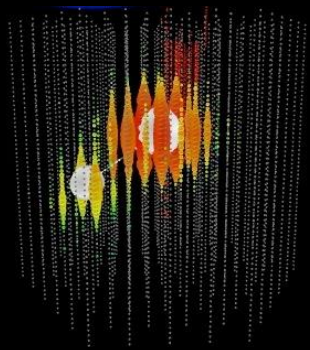
Pulse Series: (t_i, q_i)



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

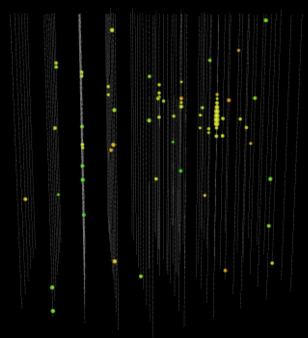


Event Topologies

CC ν_μ	CC ν_e / NC ν_*	CC ν_τ
		
$\nu_\mu + N \rightarrow \mu + X$	$\nu_e + N \rightarrow \mathrm{e} + X$ $\nu_* + N \rightarrow \nu_* + X$	$\nu_\tau + N \rightarrow \tau + X$
Track	Cascade	Cascade / Track / Double-Cascade

Event Topologies

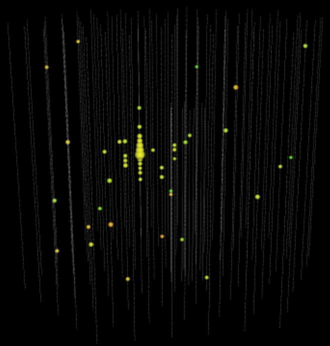
CC ν_μ



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

Track

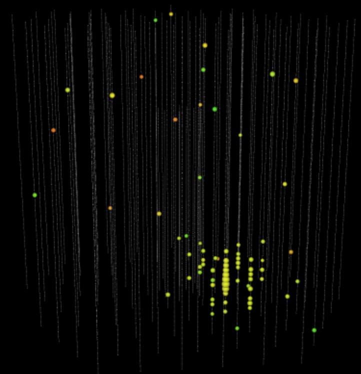
CC ν_e / NC ν_*



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

Cascade

CC ν_τ

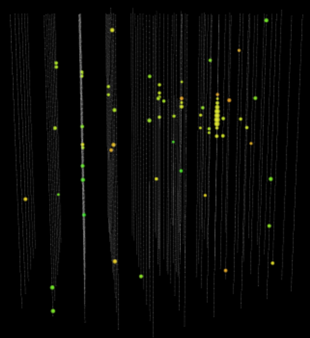


$$\nu_\tau + N \rightarrow \tau + X$$

Cascade / Track / Double-Cascade

Event Topologies

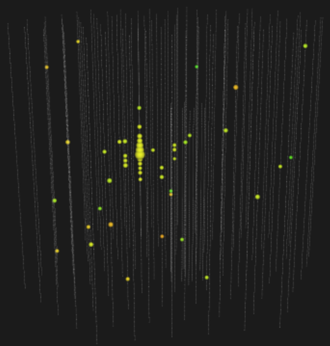
CC ν_μ



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

Track

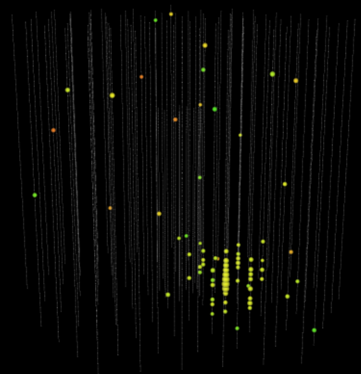
CC ν_e / NC ν_*



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

Cascade

CC ν_τ

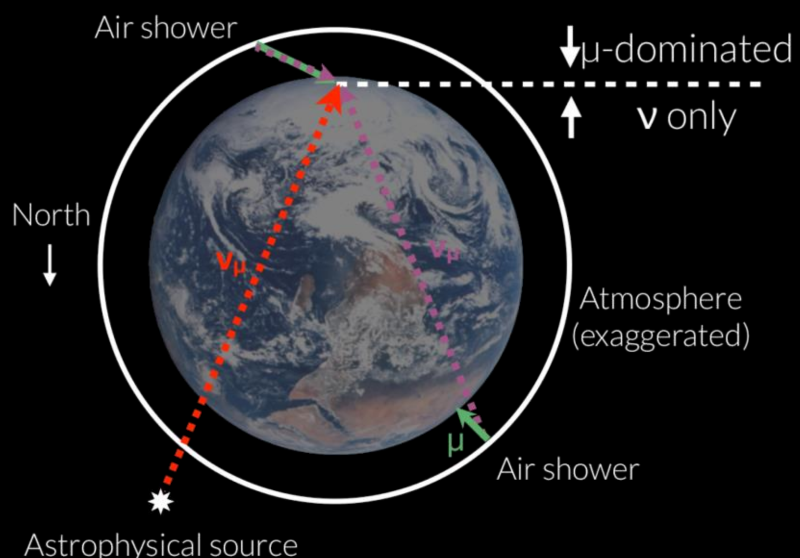


$$\nu_\tau + N \rightarrow \tau + X$$

Cascade / Track / Double-Cascade

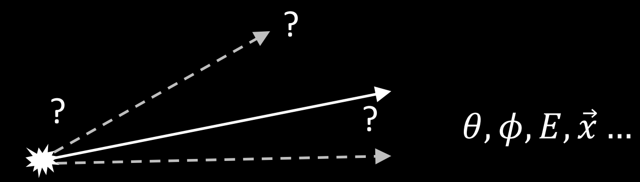
Main Reconstruction Tasks

Event Selection and Classification

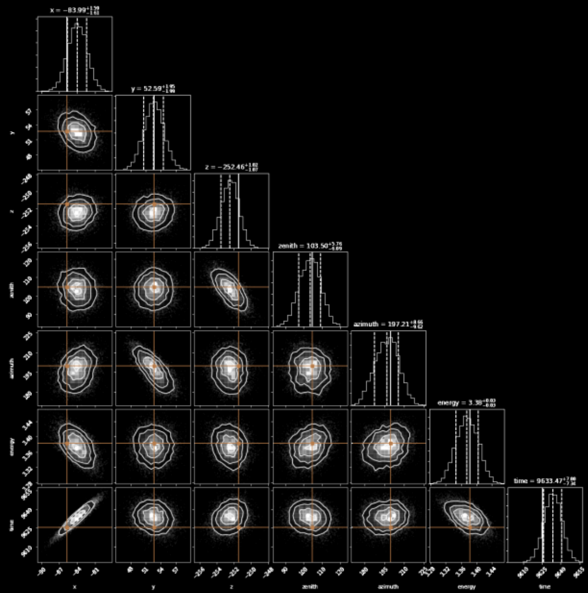


Rates:
Atmospheric Muons: ~10^3 Hz
Atmospheric Neutrinos: ~10^{-3} Hz
Astrophysical Neutrinos: ~10^{-7} Hz

Estimation of Event Parameters

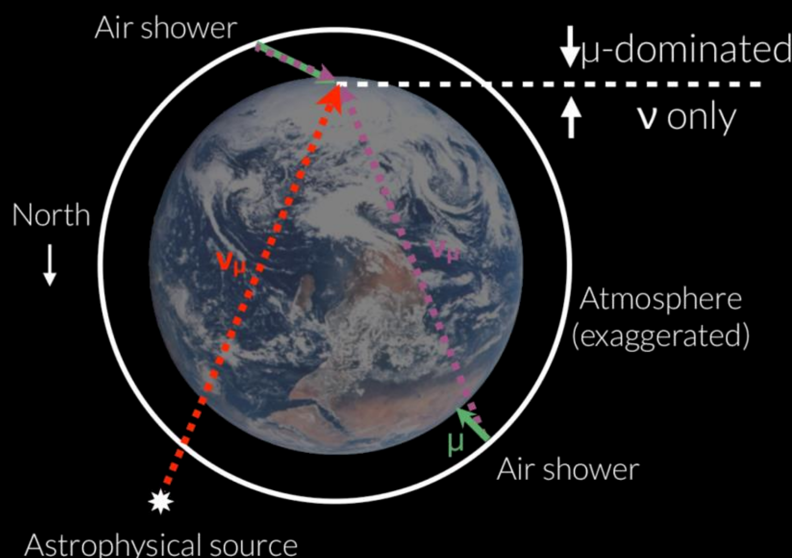


Uncertainty Estimation



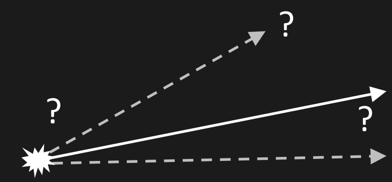
Main Reconstruction Tasks

Event Selection and Classification



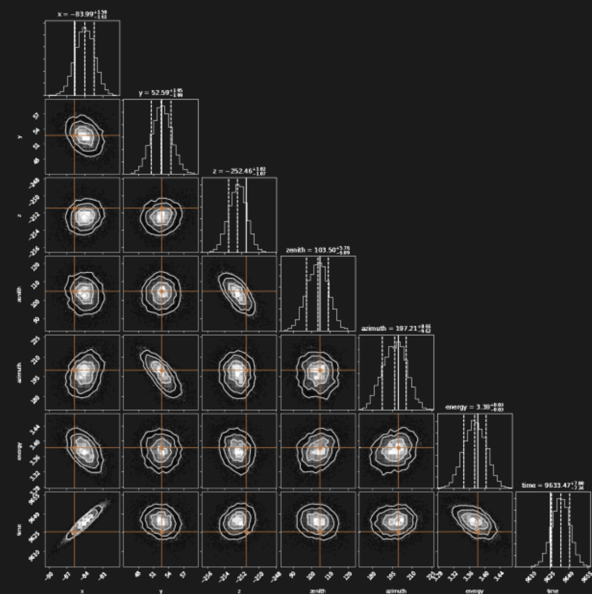
Rates:
Atmospheric Muons: ~10^3 Hz
Atmospheric Neutrinos: ~10^{-3} Hz
Astrophysical Neutrinos: ~10^{-7} Hz

Estimation of Event Parameters



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

Uncertainty Estimation



Deep Learning in IceCube

Going Deep

- 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

Deep Learning in IceCube

Going Deep

- 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

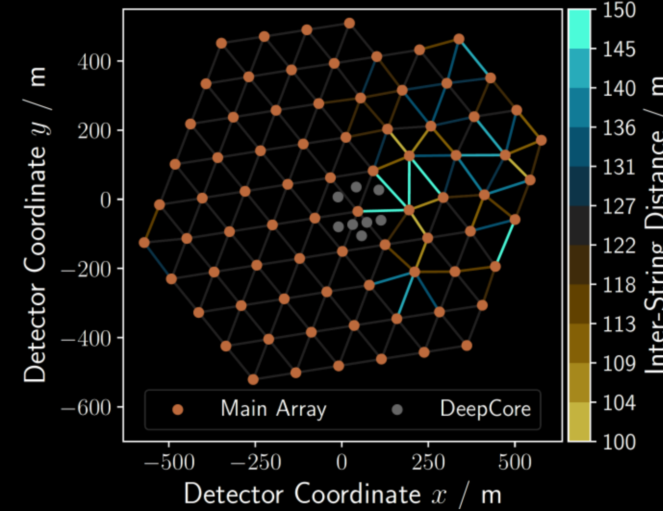
Architectures Investigated

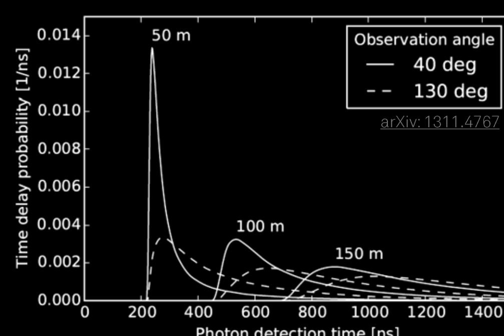
- Convolutional Neural Network (CNN)*
- Recurrent Neural Network (RNN)
- Graph Neural Network (GNN)
- Hybrid Maximum-Likelihood Estimation (MLE) / Deep Learning (DL) approaches**

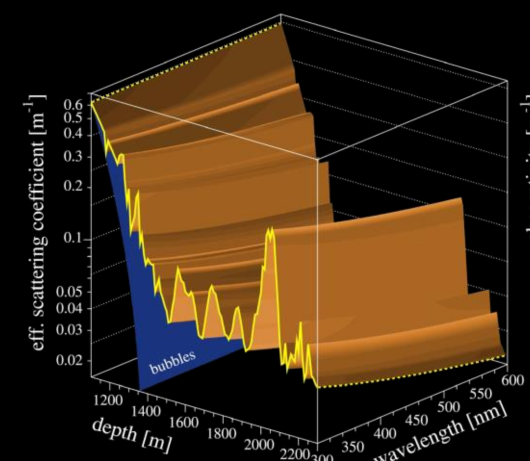
Focus in this talk is on these publications:
* DOI: 10.1088/1748-0221/16/07/P07041
** DOI: 10.22323/1.395.1065

What do we know about our Data?

- Detector geometry:
  - 3 detector parts: main array, upper & lower DeepCore
  - Deviations from symmetric detector grid
- Underlying physics of neutrino interaction are invariant under translation and rotation
- Inhomogeneous photon propagation due to dust impurities and crystal structure of ice
- Light yield scales linearly with deposited energy
- General shape of photon arrival time PDF
- Photons (and in good approximation: the measured pulses) are independent of each other







Source: 10.1029/2005JD006687 [4]

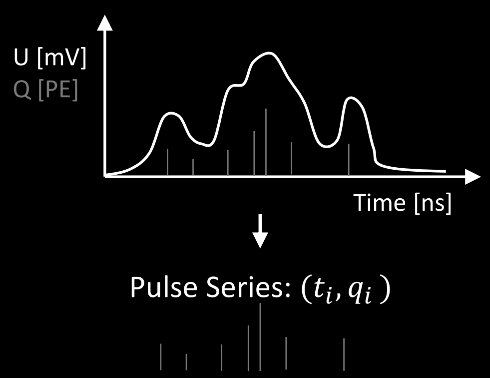
Convolutional Neural Network (CNN)

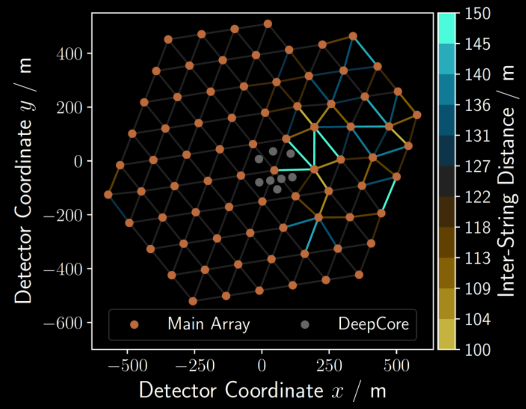
Idea:

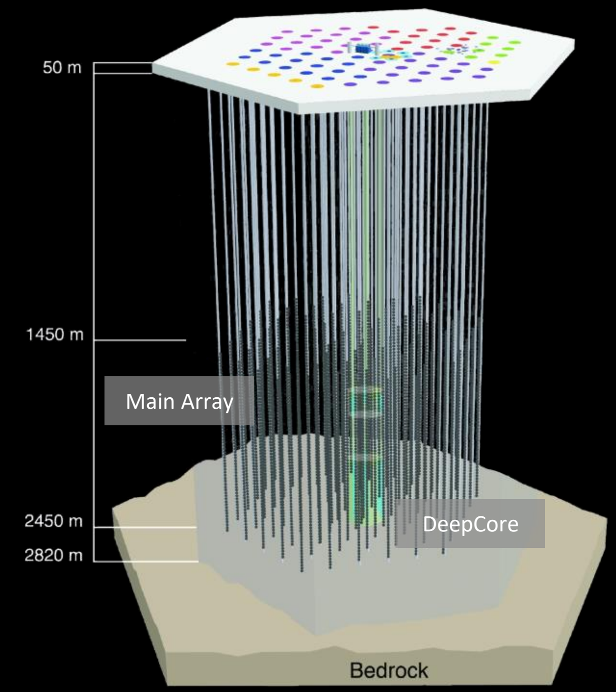
- Use CNNs to exploit translational invariance
- CNNs are easy to train and usually provide a good first benchmark

Challenges:

- How to deal with 3 detector parts and triangular grid?
- CNNs require uniform input:
  what to do with variable number of pulses?



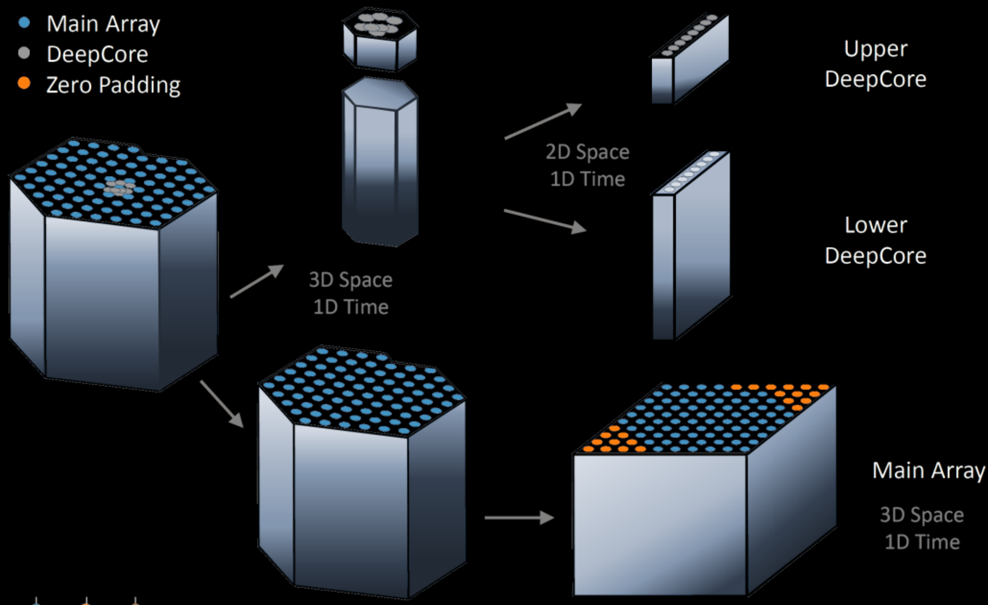


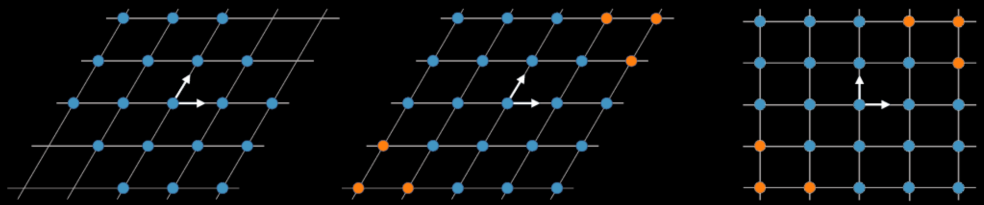


Convolutional Neural Network (CNN)

Detector Geometry

- Separate detector into 3 input tensors
- Transform triangular grid to orthogonal data matrix
- Apply mapping to effectively employ hexagonal convolution kernels



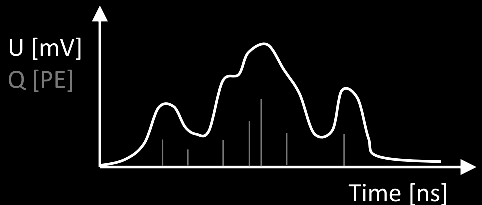


Hexagonal Convolution Kernels

Convolutional Neural Network (CNN)

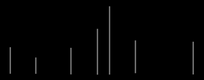
Input Data

- Waveforms:
    - Arbitrary length and very high-dimensional
- Pulse Series:
    - Highly variable length
    - Very efficient data representation
- Summary Statistics on Pulse Series
    - Constant input size
    - Loss of information due to dimensionality reduction
    - Option to further improve data/MC agreement



→

Pulse Series: (t_i, q_i)



→

Summary Statistics

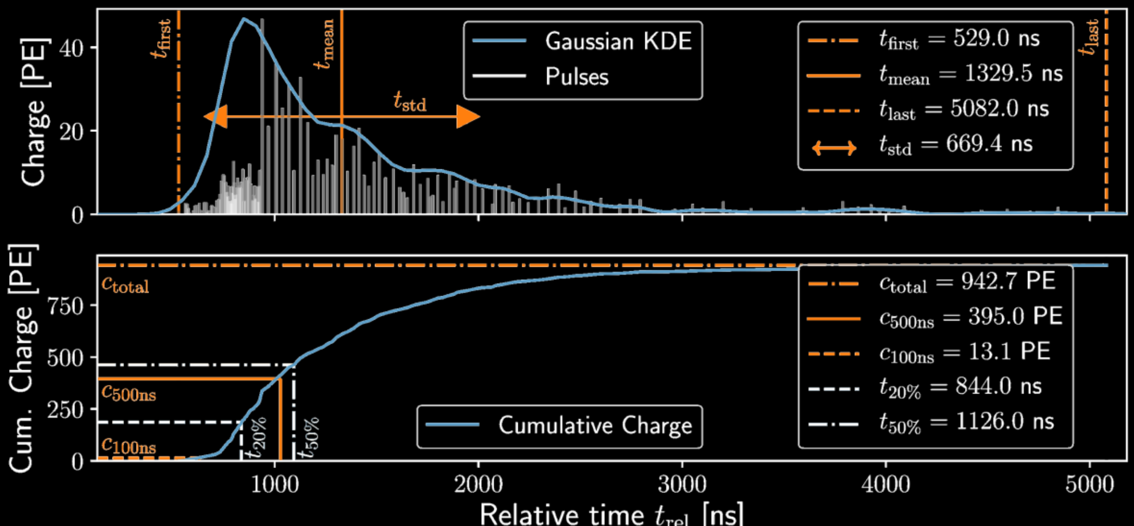
$$\sum q_i, \frac{1}{N}\sum t_i, \ldots$$

Convolutional Neural Network (CNN)

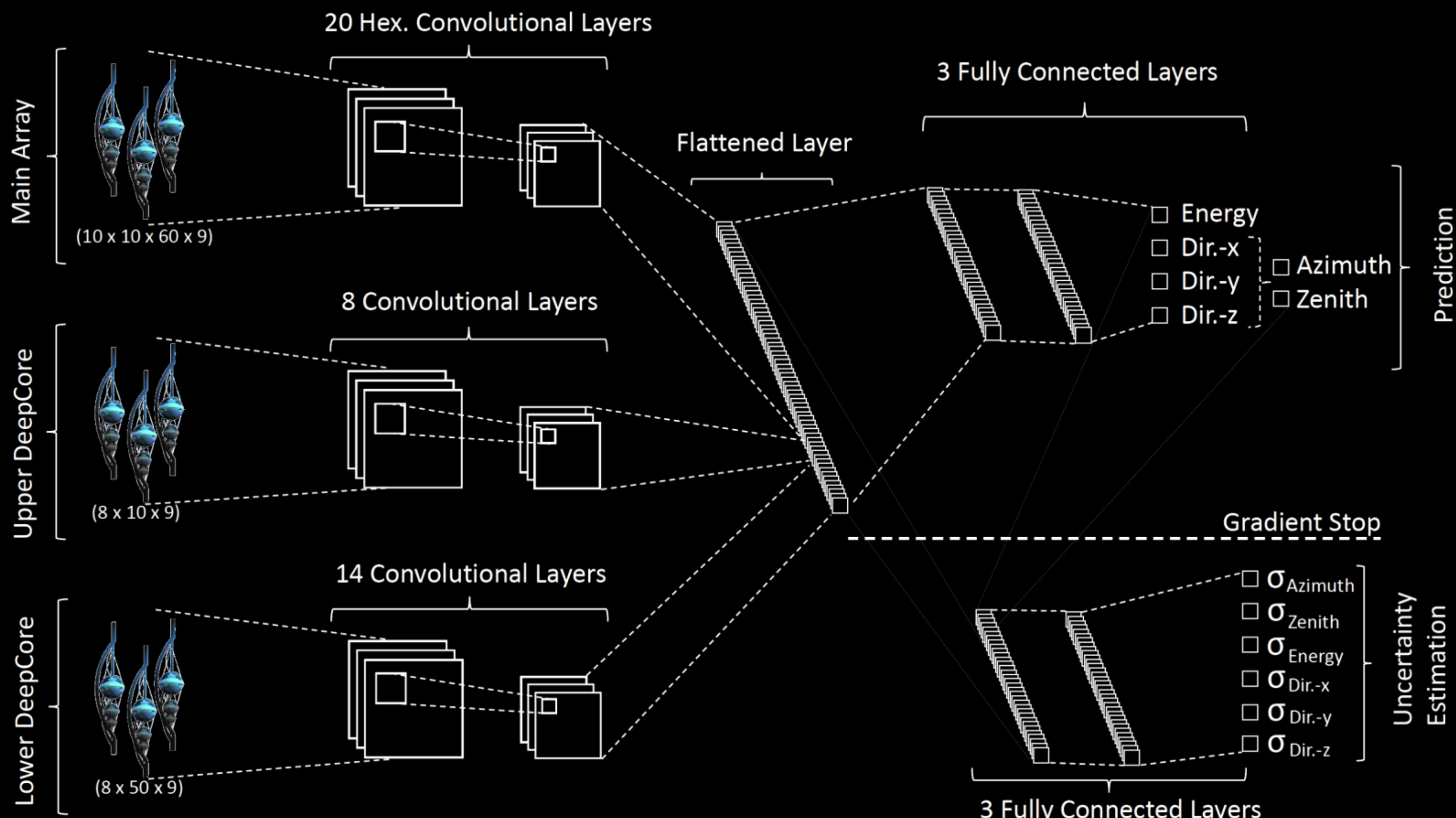
Input Data

- Waveforms:
  - Arbitrary length and very high-dimensional
- Pulse Series:
  - Highly variable length
  - Very efficient data representation

- Summary Statistics on Pulse Series
  - Constant input size
  - Loss of information due to dimensionality reduction
  - Option to further improve data/MC agreement



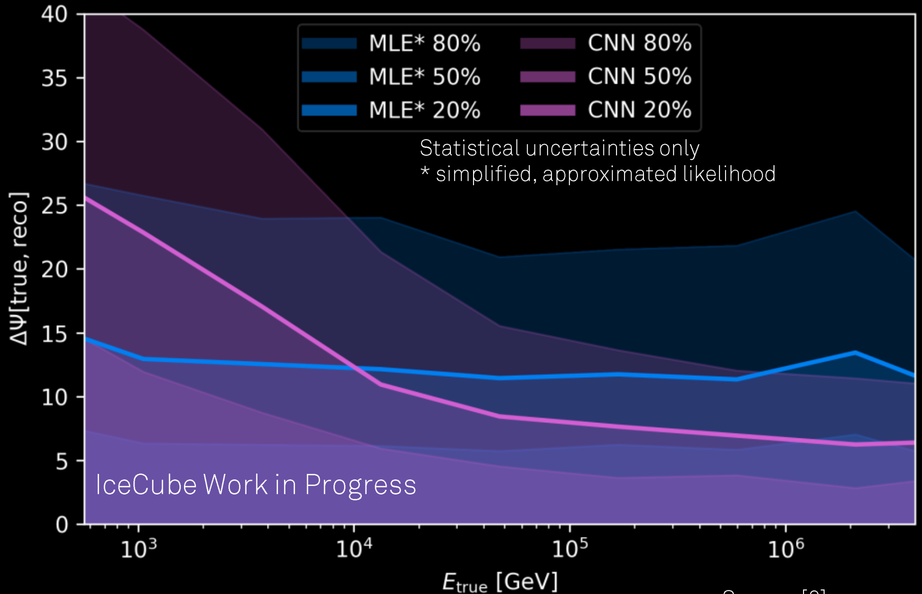
Convolutional Neural Network (CNN)



Convolutional Neural Network (CNN)

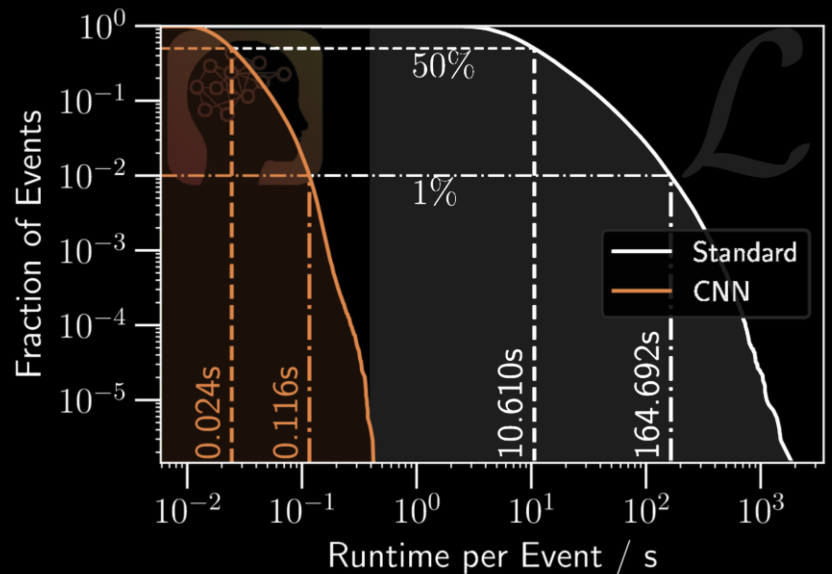
Performance

Cascade Angular Resolution



Source: [6]

Reconstruction Runtime

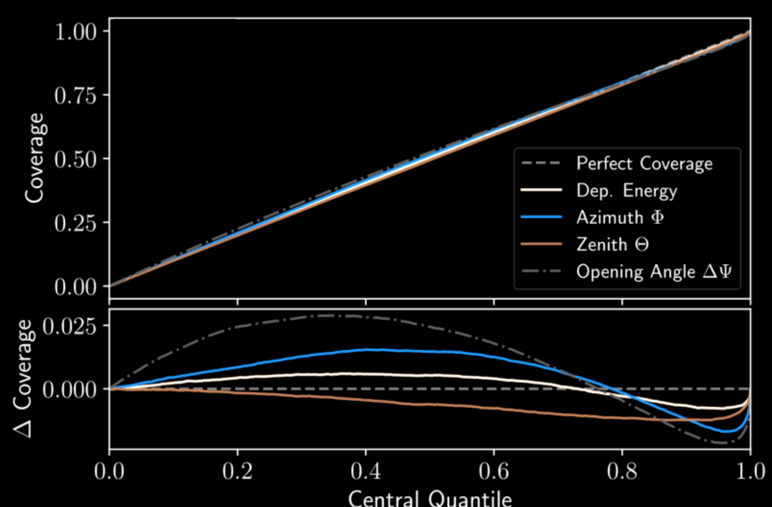


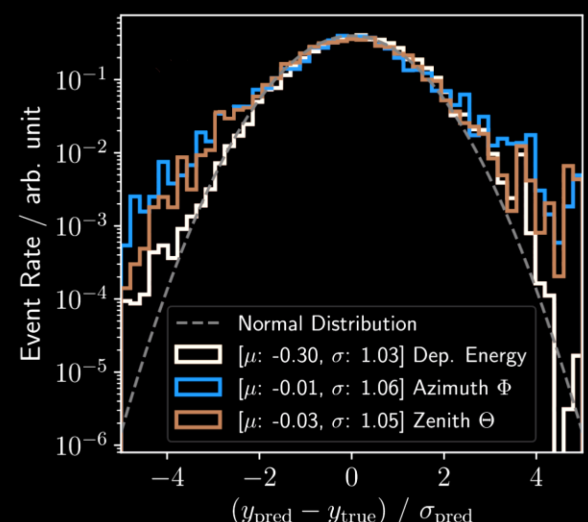
Convolutional Neural Network (CNN)

Uncertainty Estimation

- Point predictions aren't enough for use in analyses: need uncertainties
- Assume Gaussian Likelihood and train NN to estimate y_{pred} and σ_{pred}

$$\text{loss} = \frac{1}{n}\sum_{i=1}^{n}\left(\ln\left(\sigma^2_{\text{pred}_i}\right) + \frac{(y_{\text{pred}_i} - y_{\text{true}_i})^2}{\sigma^2_{\text{pred}_i}}\right)$$

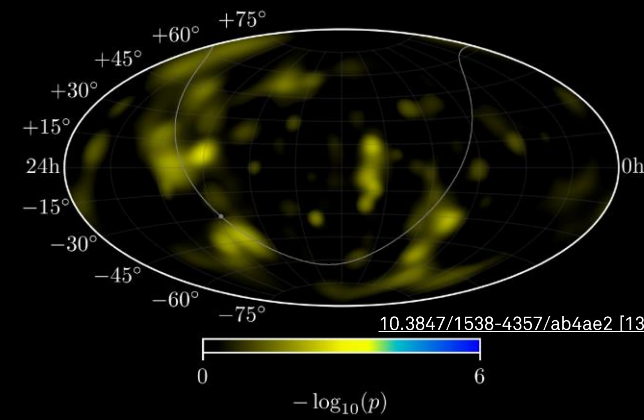




Convolutional Neural Network (CNN)

Summary

- CNNs can improve reconstruction accuracy
- Speed up reconstruction by orders of magnitude
- Results have good data/MC agreement and are robust
- CNNs ([5, 6, 7]) in unblinded analyses:
  - Cascade-based neutrino search
  - Cascade real-time alert stream



10.3847/1538-4357/ab4ae2 [13]

Pros/Cons:

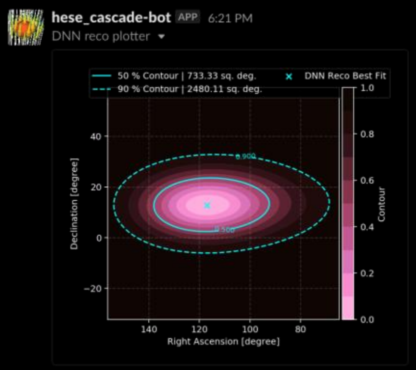
⊕ Exploit (approximate) translational invariance in data

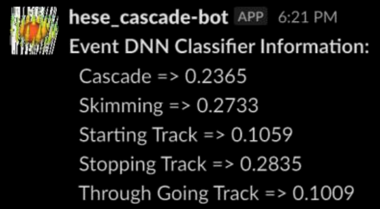
⊖ CNN assumes symmetric grid

⊖ Cannot naturally account for inhomogeneities in detector medium

⊖ Loss of information due to summary statistics

⊖ Inclusion of additional domain knowledge is difficult





Talk Outline

My personal view on Deep Learning

- Deep learning as a tool for function approximation
- Utilization of domain knowledge and symmetries

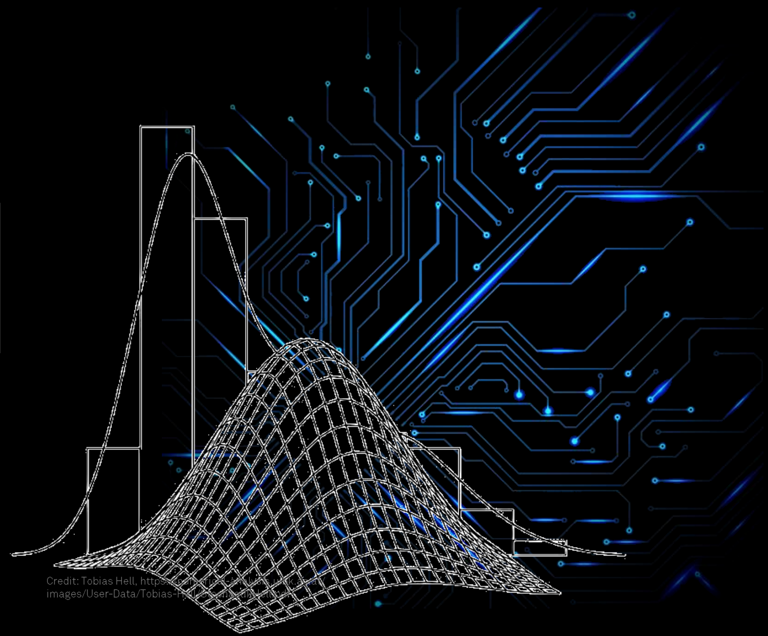
"Classical" Deep Learning in IceCube

- Choice of data representation and NN architecture
- Convolutional neural networks

Hybrid MLE/DL Method

- Combining maximum-likelihood with Deep Learning

Conclusions and Outlook



Maximum Likelihood Estimation (MLE)

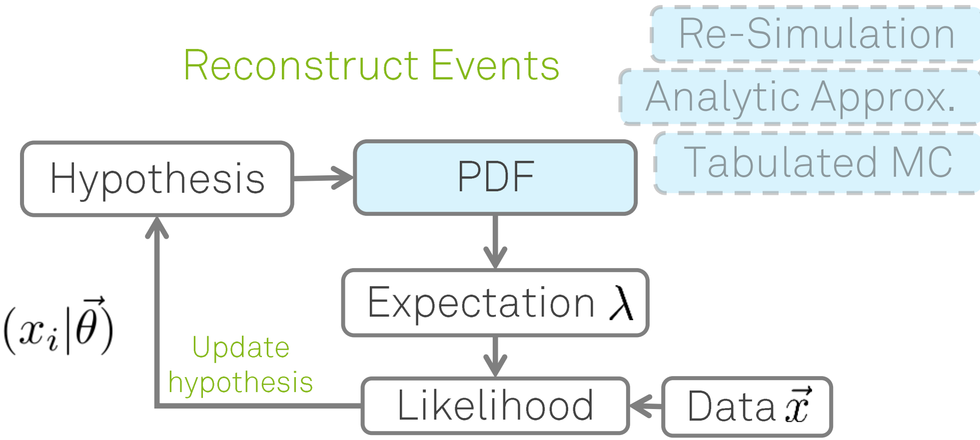
Alter event hypothesis until it matches data
Cascade Hypothesis:

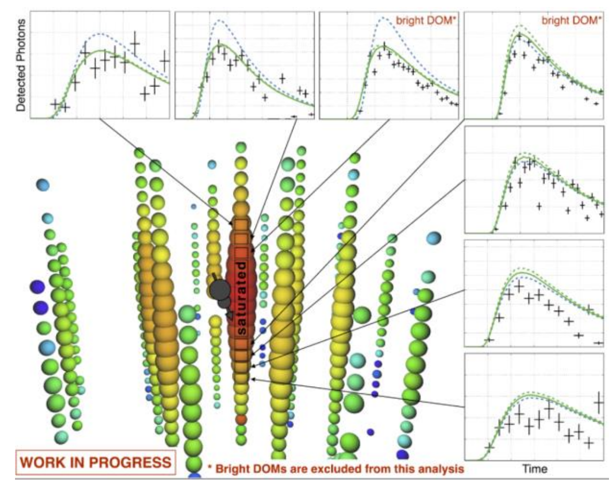
$$\vec{\theta} = (x, y, z, \varphi, \theta, E, t)$$ 7 free parameters

Properties:

$$\mathcal{L}(\vec{x}|\vec{\theta}) = \prod_i p(x_i|\vec{\theta})$$

- ⊕ Physics knowledge is incorporated into the Likelihood and PDF
- ⊕ Exact detector geometry can be used
- ⊕ In theory: optimum of what we can do
- ⊖ Often forced to simplify PDF and Likelihood
- ⊖ Difficult to find global minimum





Credit: HESE team

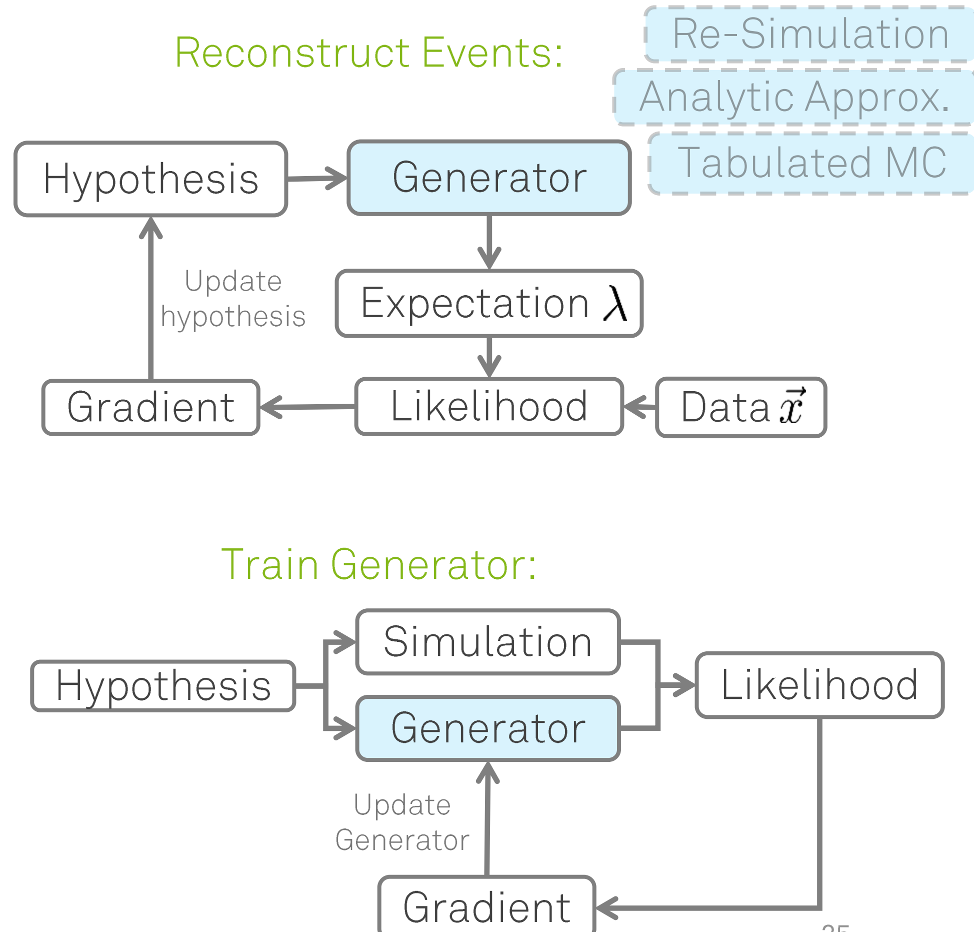
Event-Generator – General Idea

Combine strengths of neural networks and maximum-likelihood methods

Properties of Generator NN:

- Fast approximation of MC simulation
- Physics knowledge & symmetries can easily be included
- Exact detector geometry can be used
- Use in reverse mode for reconstruction

→ Fully differentiable: Gradient Descent



Event-Generator – Architecture

Generator NN learns mapping:

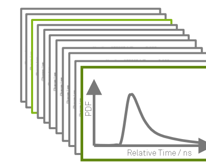
$$f:\vec{\theta}\rightarrow\vec{\lambda},\ \vec{p}(t_i|\vec{\theta})$$

$\vec{\theta}$: source parameters
$\vec{\lambda}$: expected DOM charge
$\vec{p}(t_i|\vec{\theta})$: pulse arrival PDFs

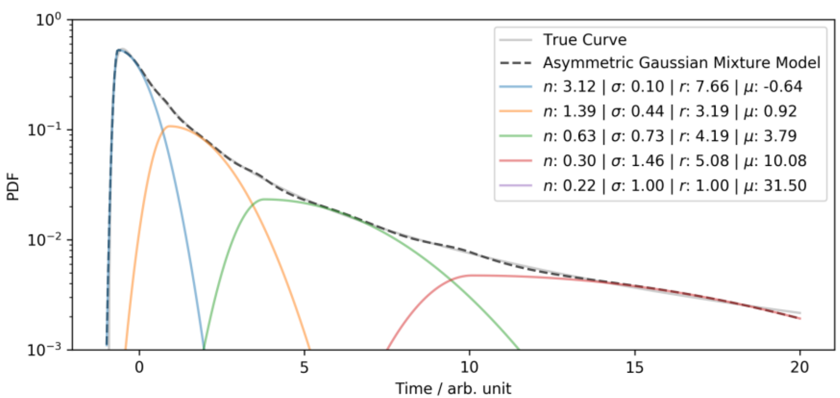
$$\underbrace{\vec{\theta}=(x,y,z,\varphi,\theta,E,t)}_{\text{Cascade Hypothesis}} \rightarrow \text{Generator} \rightarrow$$

Neural Network that maps input [?, 7] to output [?, 5160, n] with number of mixture model components n

5160 PDFs for each of the DOMs



Parameterization of time PDF:

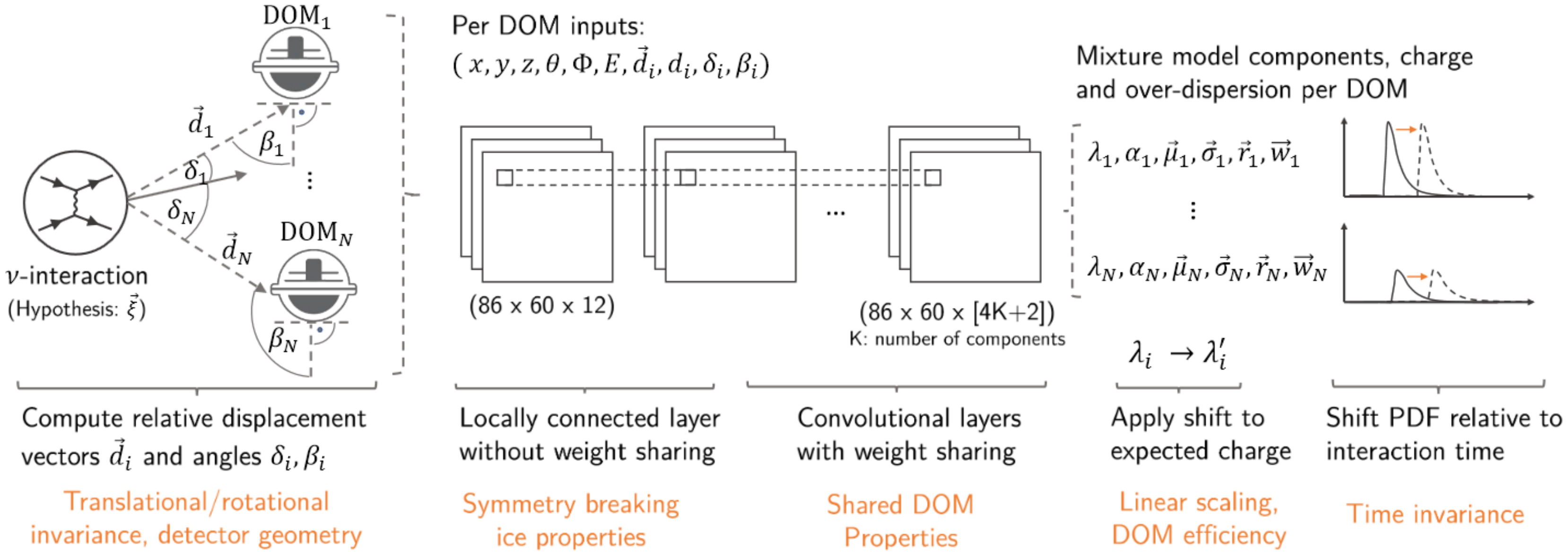


$$g(x|\mu,\sigma,r)=\begin{cases} N\cdot\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), & x\leq\mu \\ N\cdot\exp\left(-\frac{(x-\mu)^2}{2(\sigma r)^2}\right), & \text{otherwise}\end{cases}$$

$$N=\frac{2}{\sqrt{(2\pi)}\cdot\sigma(r+1)}$$

10.1007/3-540-70659-3_42

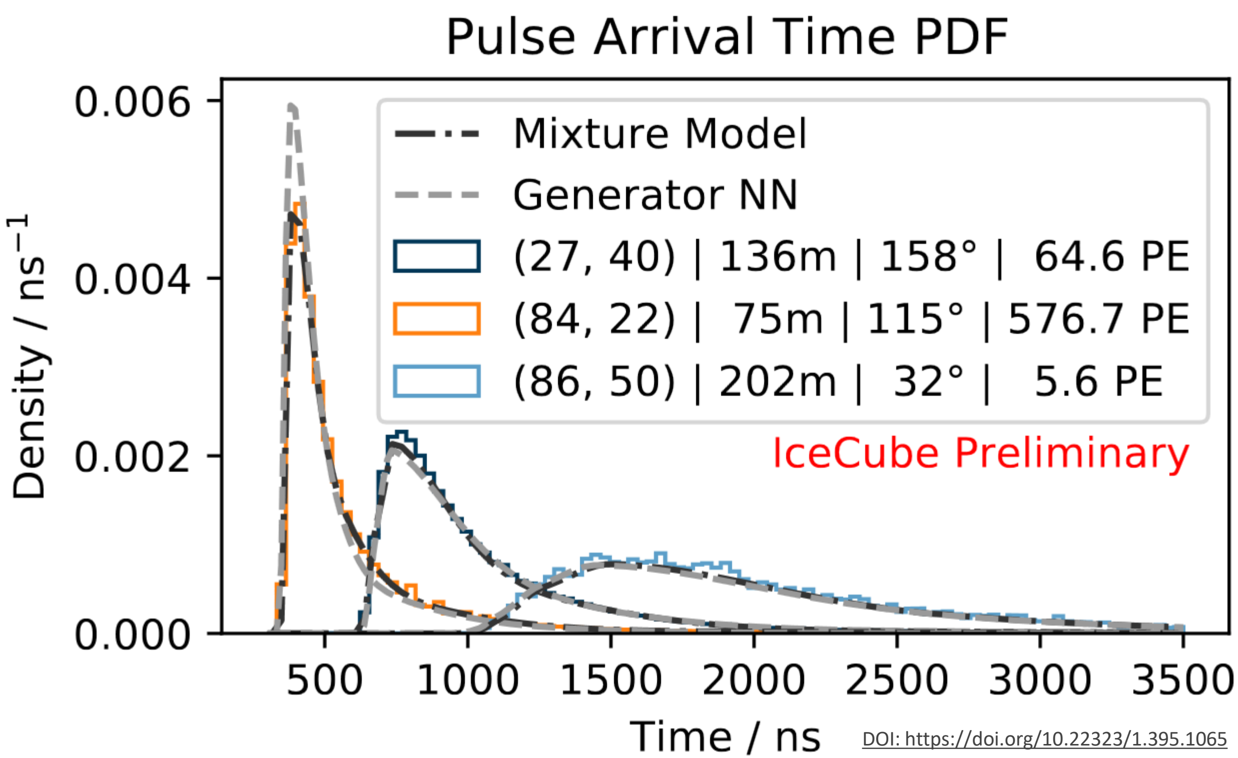
Event-Generator – Generator Architecture



DOI: https://doi.org/10.22323/1.395.1065

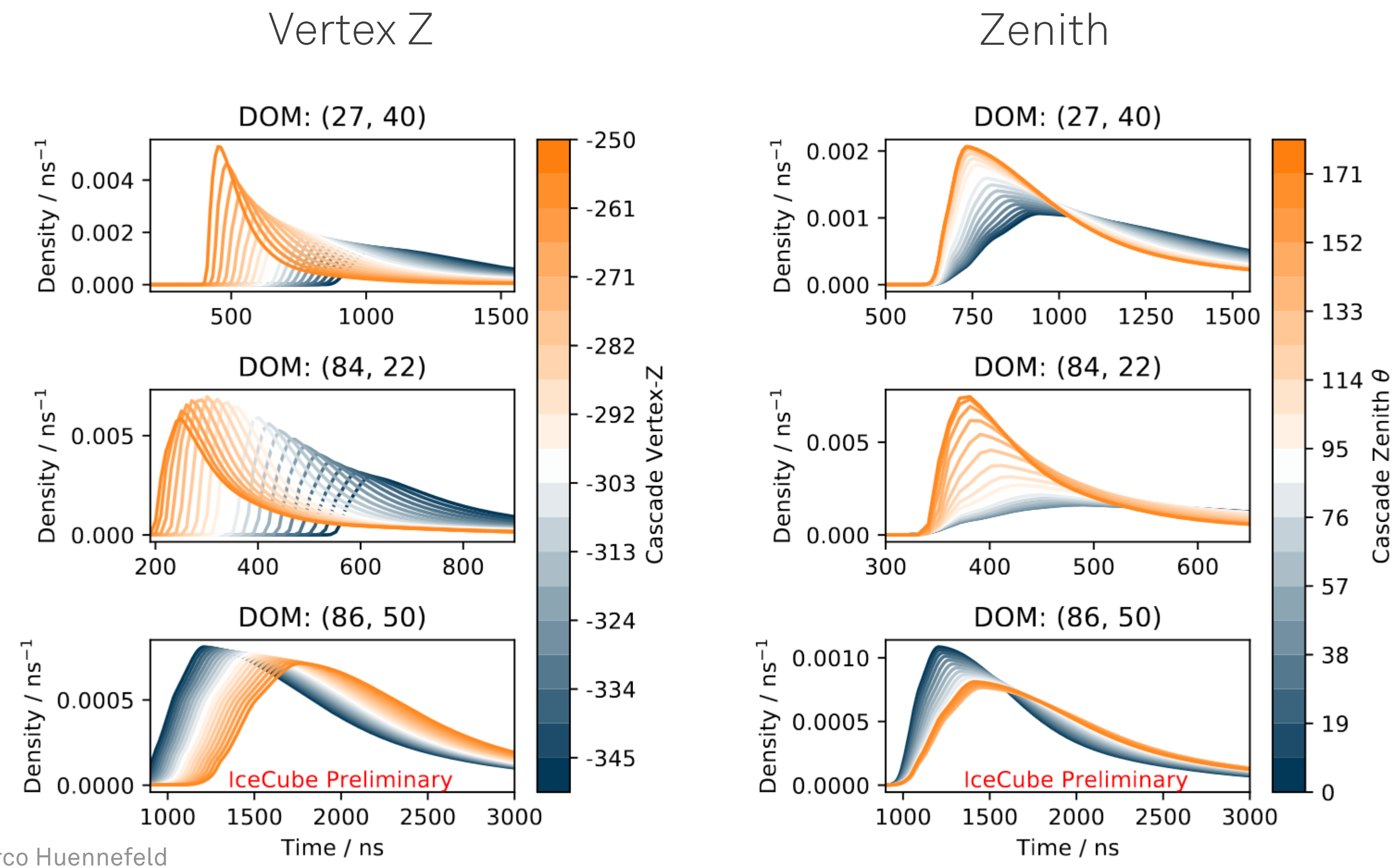
- Can decouple physics and detector effects
- Easier to include information in forward direction when not convolved with detector response yet
- We know how to do this (we simulate the data!)

Event-Generator – Example Outputs



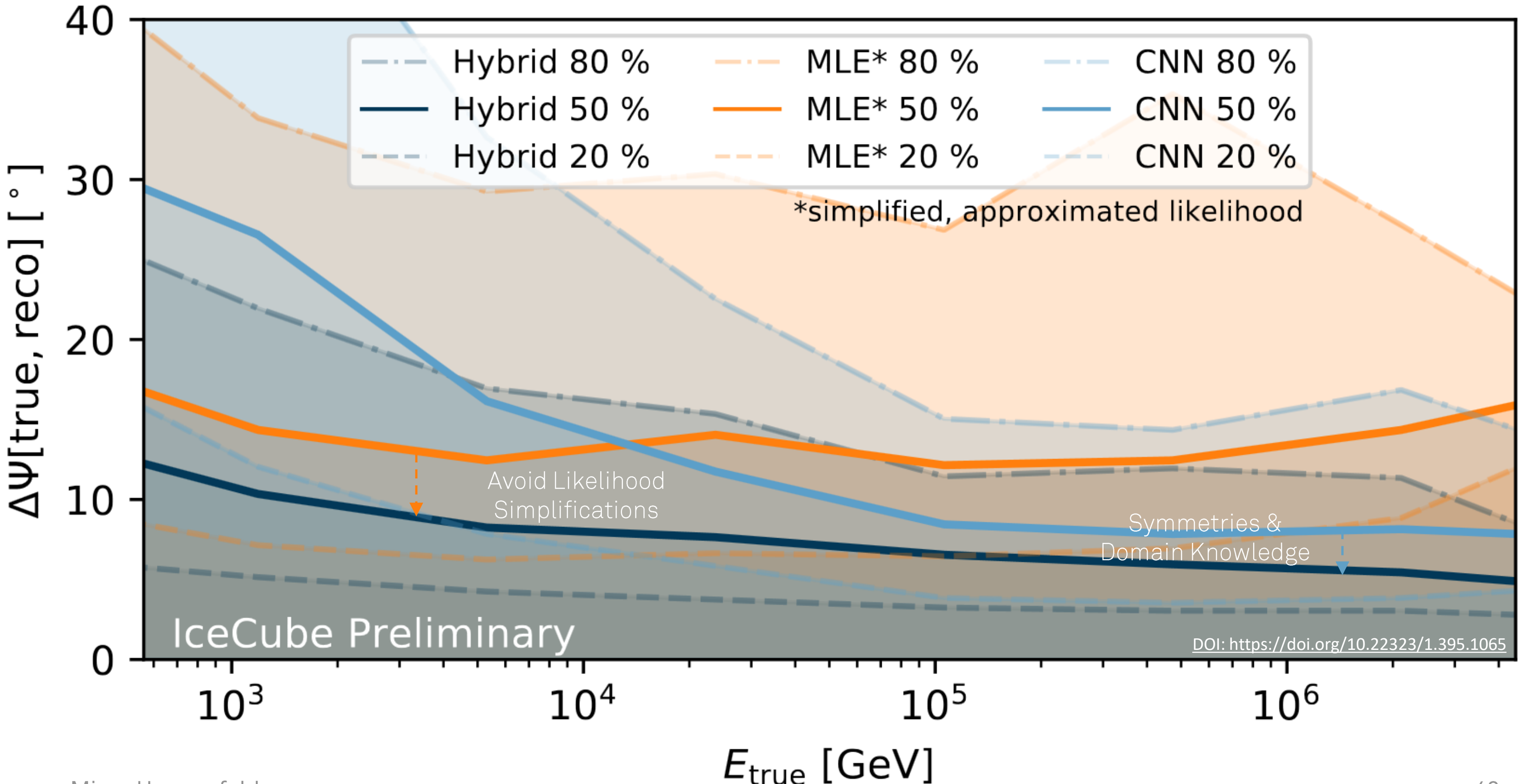
DOI: https://doi.org/10.22323/1.395.1065

Event-Generator – Example Outputs



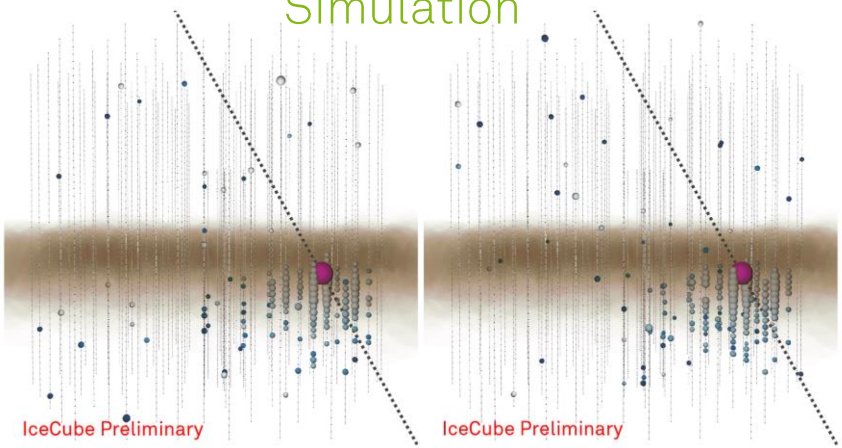
DOI: https://doi.org/10.22323/1.395.1065

Event-Generator – Cascade Angular Resolution



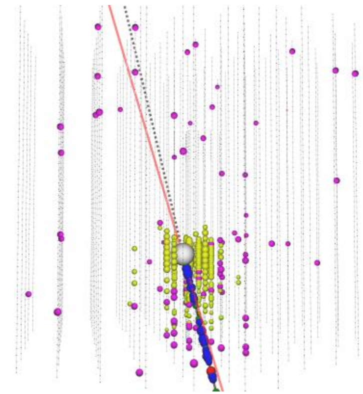
Event-Generator – Additional Applications

Simulation

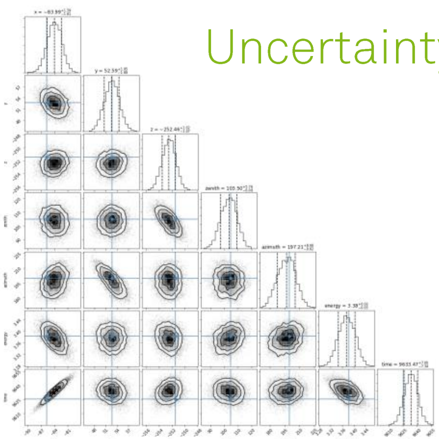


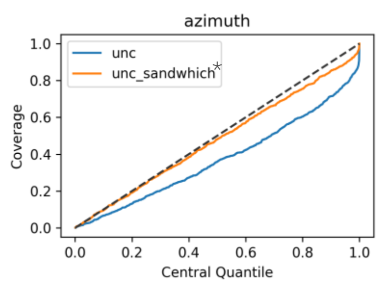
DOI: https://doi.org/10.22323/1.395.1065

Goodness -of-fit



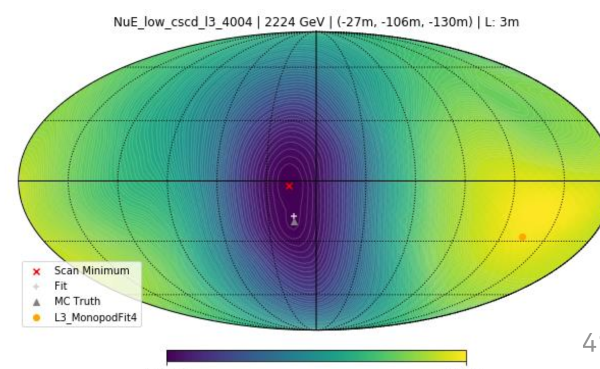
Uncertainty Estimation



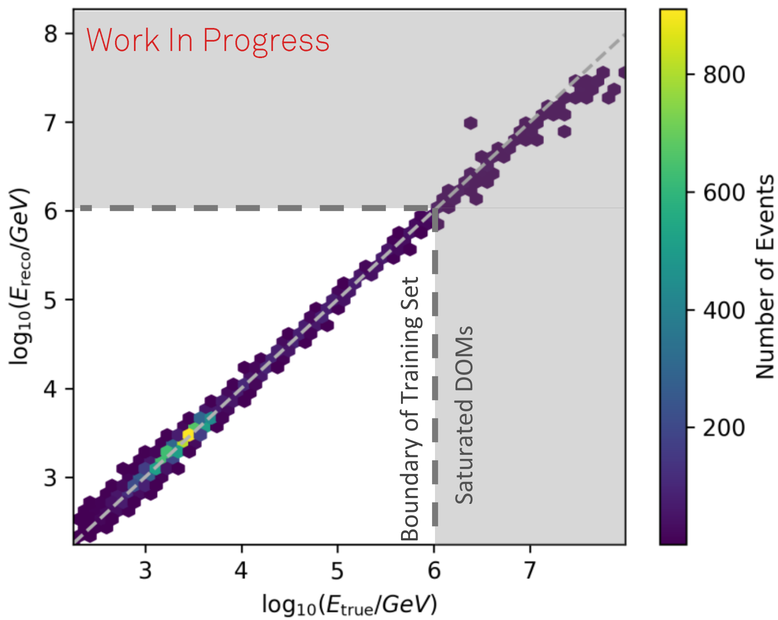


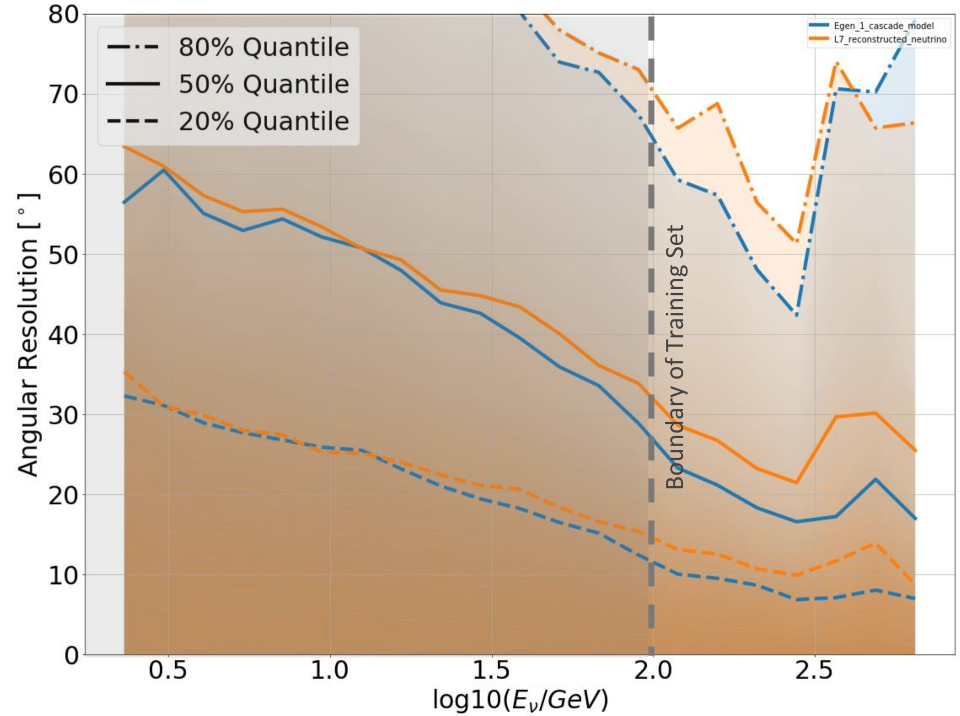
*http://www.stat.umn.edu/geyer/5601/notes/sand.pdf

Likelihood Scans



Event-Generator – Extrapolation





Model can extrapolate (until DOMs start to saturate) beyond training data due to incorporation of linear light scaling into architecture

Talk Outline

My personal view on Deep Learning

- Deep learning as a tool for function approximation
- Utilization of domain knowledge and symmetries

"Classical" Deep Learning in IceCube

- Choice of data representation and NN architecture
- Convolutional neural networks

Hybrid MLE/DL Method

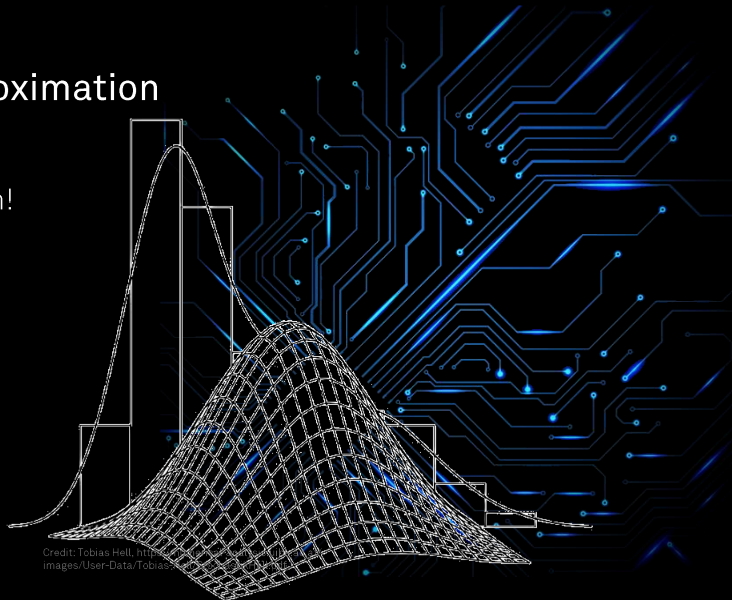
- Combining maximum-likelihood with Deep Learning

Conclusions and Outlook

Conclusions and Outlook

Deep Learning is an incredibly powerful tool for function approximation

- Various NN architectures are different ways to define this function
- Applicable in many areas outside of classification and reconstruction!



Importance of exploiting symmetries and domain knowledge

- Reduces amount of necessary parameters
- Facilitates training and improves performance
- More robust models
- Allows for generalization

Hybrid & data driven methods

- This is where DL's properties as function approximators are extremely valuable!
- Rich set of opportunities for calibration and data/MC improvements
- Data-driven methods: can build hybrid models trained on exp data or utilize DL to derive physics directly from data

This is an exciting area where Physics experiments can greatly benefit from!

→ Utilize DL as the powerful tool it is and be creative!


THE ICECUBE COLLABORATION
AUSTRALIA
University of Adelaide
BELGIUM
Université libre de Bruxelles
Universiteit Gent
Vrije Universiteit Brussel
CANADA
SNOLAB
University of Alberta–Edmonton
DENMARK
University of Copenhagen
GERMANY
Deutsches Elektronen-Synchrotron
ECAP, Universität Erlangen-Nürnberg
Humboldt–Universität zu Berlin
Ruhr-Universität Bochum
RWTH Aachen University
Technische Universität Dortmund
Technische Universität München
Universität Mainz
Universität Wuppertal
Westfälische Wilhelms-Universität Münster
JAPAN
Chiba University
NEW ZEALAND
University of Canterbury
REPUBLIC OF KOREA
Sungkyunkwan University
SWEDEN
Stockholms Universitet
Uppsala Universitet
SWITZERLAND
Université de Genève
UNITED KINGDOM
University of Oxford
UNITED STATES
Clark Atlanta University
Drexel University
Georgia Institute of Technology
Lawrence Berkeley National Lab
Marquette University
Massachusetts Institute of Technology
Michigan State University
Ohio State University
Pennsylvania State University
South Dakota School of Mines and Technology
Southern University and A&M College
Stony Brook University
University of Alabama
University of Alaska Anchorage
University of California, Berkeley
University of California, Irvine
University of Delaware
University of Kansas
University of Maryland
University of Rochester
University of Texas at Arlington
University of Wisconsin–Madison
University of Wisconsin–River Falls
Yale University
FUNDING AGENCIES
Fonds de la Recherche Scientifique (FRS-FNRS)
Fonds Wetenschappelijk Onderzoek-Vlaanderen (FWO-Vlaanderen)
Federal Ministry of Education and Research (BMBF)
German Research Foundation (DFG)
Deutsches Elektronen-Synchrotron (DESY)
Japan Society for the Promotion of Science (JSPS)
Knut and Alice Wallenberg Foundation
Swedish Polar Research Secretariat
The Swedish Research Council (VR)
University of Wisconsin Alumni Research Foundation (WARF)
US National Science Foundation (NSF)
ICECUBE
icecube.wisc.edu