TRIUMF

# Machine learning techniques to enhance water Cherenkov reconstruction

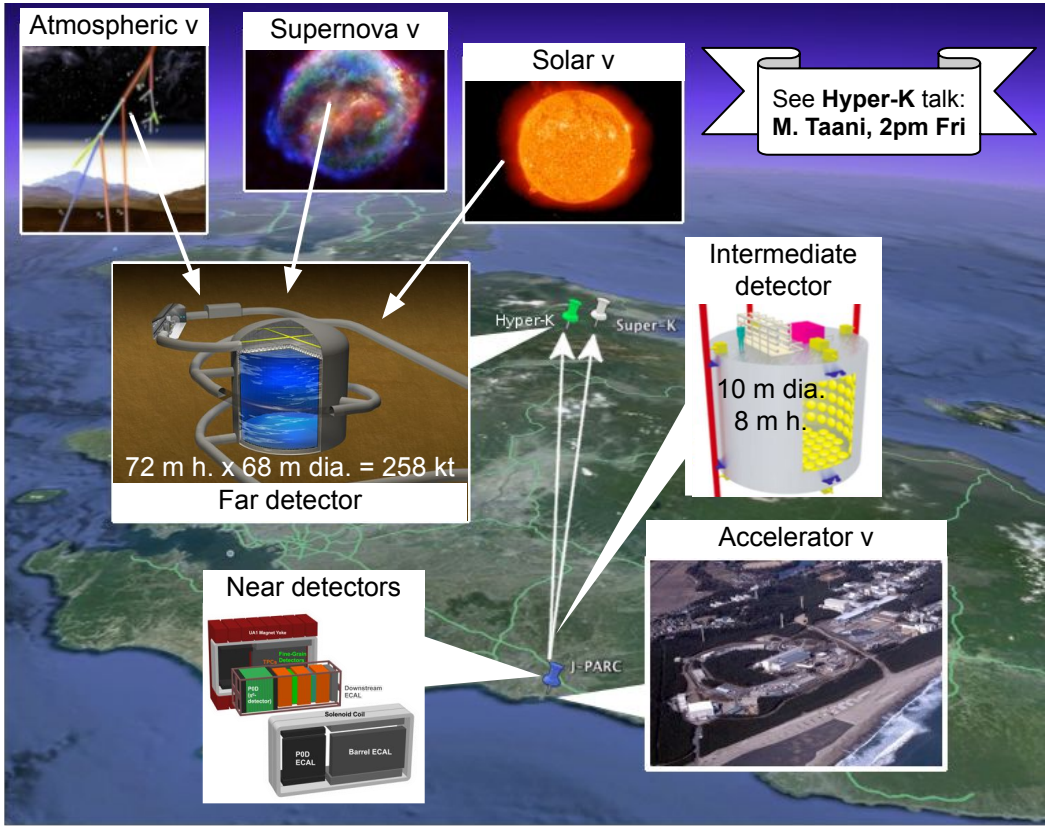
WatChMaL.org

Nick Prouse, TRIUMF

NNN19, University of Medellin, Colombia

7th November 2019

# The Hyper-Kamiokande Experiment




Hyper-K builds on huge success of Super-K

Next generation neutrino experiment with two new water Cherenkov detectors

- 258 kt far detector
- 0.5 kt intermediate detector

Broad & ambitious physics programme through comprehensive upgrades

- 8 x increase in fiducial mass
- Improved photosensor technology
- Reduced systematics with new near & intermediate detector measurements

Improved analysis techniques required to realise Hyper-K's precision measurements

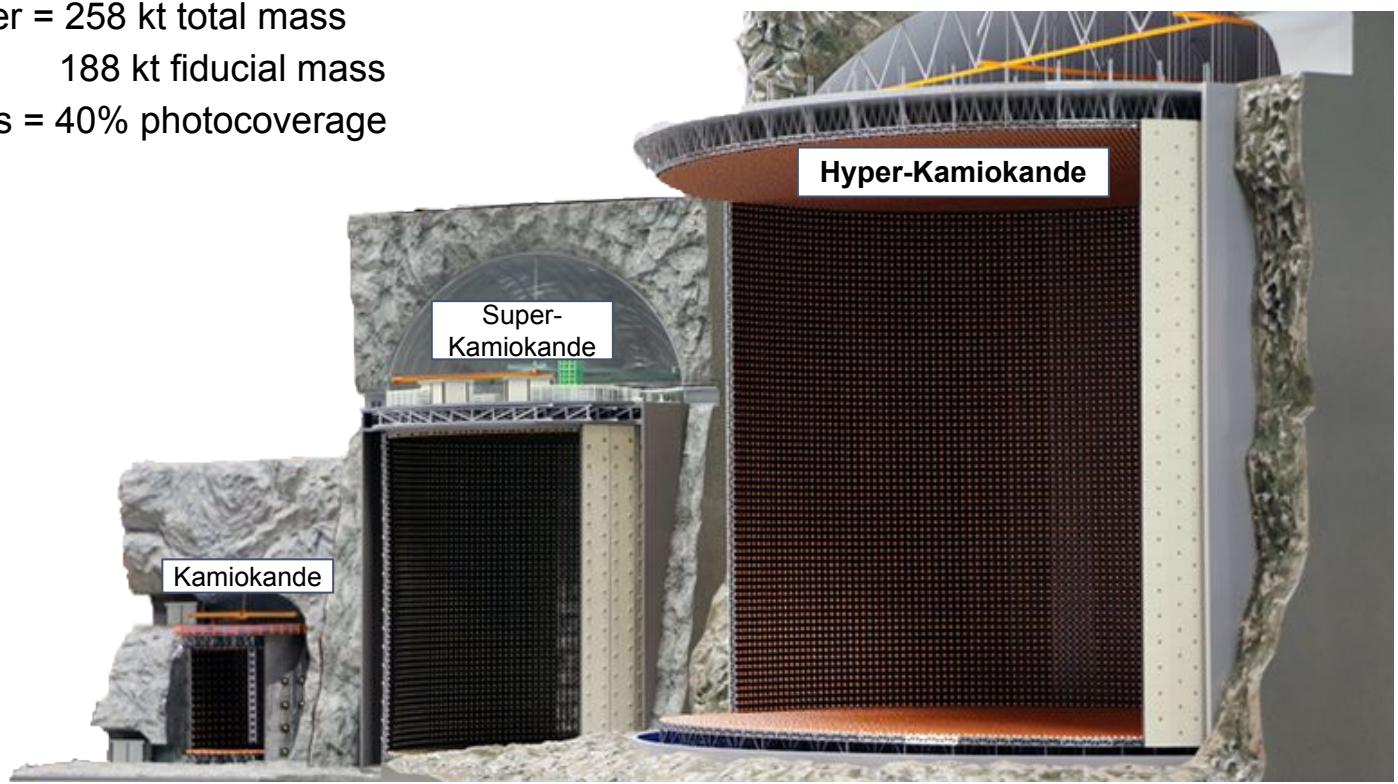
# Hyper-K's WC detectors

**Hyper-K far detector**

72 m tall x 68 m diameter = 258 kt total mass

188 kt fiducial mass

40,000 50 cm B&L PMTs = 40% photocoverage



# Hyper-K's WC detectors

**Hyper-K far detector**

72 m tall x 68 m diameter = 258 kt total mass

188 kt fiducial mass

20,000 50 cm B&L PMTs + multi-PMT modules +

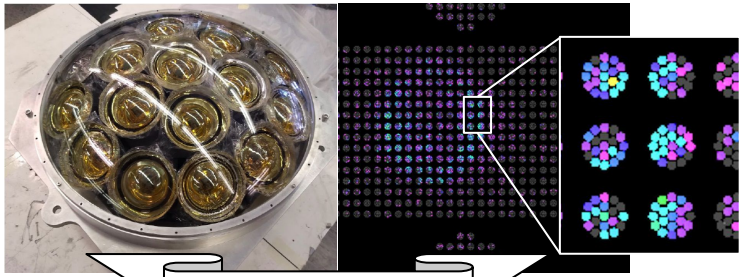
B&L or MCP PMTs ➛ equal / better performance

**Multi-PMT modules**

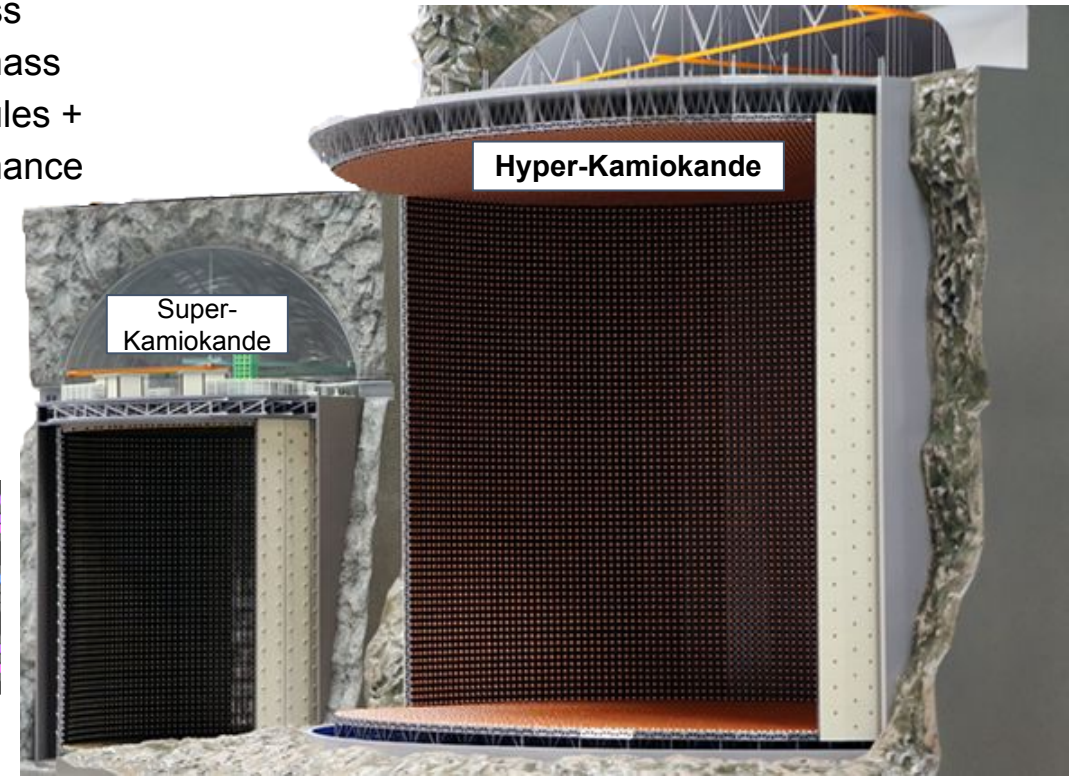
8 cm PMTs: better position resolution

Additional directionality information

< 1 ns timing resolution



See **multi-PMT** talk:
**S. Cuen-Rochin, 2pm Thu**



# Hyper-K's WC detectors

TRIUMF

**Hyper-K far detector**

72 m tall x 68 m diameter = 258 kt total mass

188 kt fiducial mass

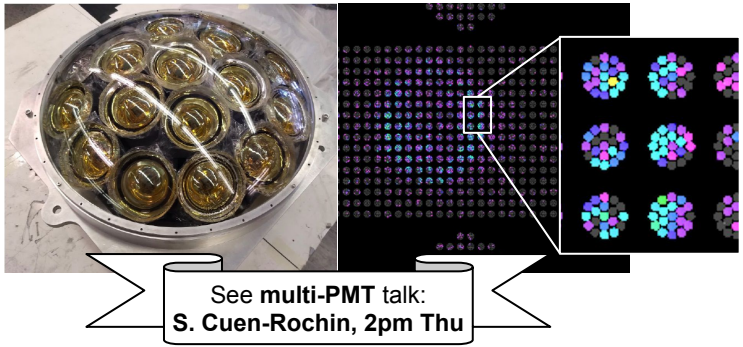
20,000 50 cm B&L PMTs + multi-PMT modules + B&L or MCP PMTs ➛ equal / better performance

**Multi-PMT modules**

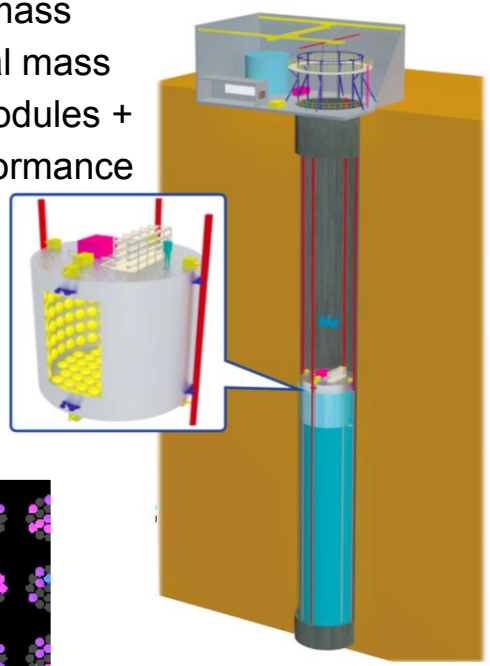
8 cm PMTs: better position resolution

Additional directionality information

< 1 ns timing resolution



See **multi-PMT** talk:
**S. Cuen-Rochin, 2pm Thu**



**Intermediate detector (IWCD)**

8 m tall x 10 m diameter tank

~500 multi-PMT modules

Moves vertically in ~50m tall pit measuring different off-axis angles gives different $\nu$ energy spectra

Measurements of unoscillated flux with high event rate, same tech and target nuclei as far detector

**New detector capabilities need new reconstruction methods**

# Machine learning for WC

TRIUMF

Following success of machine learning in other big data analysis

- **WatChMaL** group formed
  **Wat**er **Ch**erenkov **Ma**chine **L**earning

WatChMaL.org

# Machine learning for WC

TRIUMF

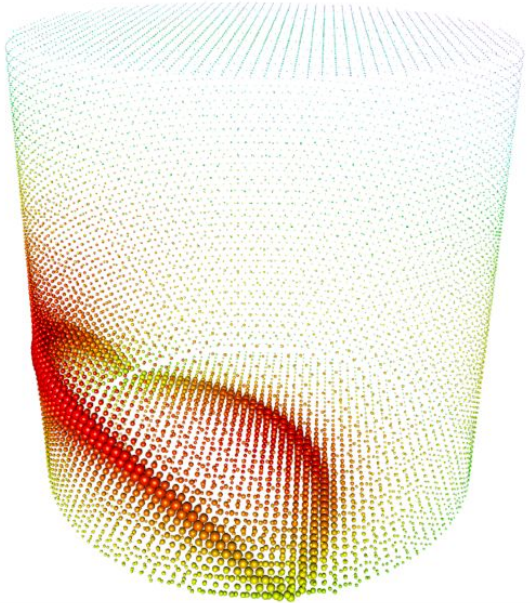
Following success of machine learning in other big data analysis

- **WatChMaL** group formed
  **Wat**er **Ch**erenkov **Ma**chine **L**earning

WatChMaL.org

Common challenges for ML with WC detectors

- Cylindrical geometry
- High-res, sparse data



# Machine learning for WC

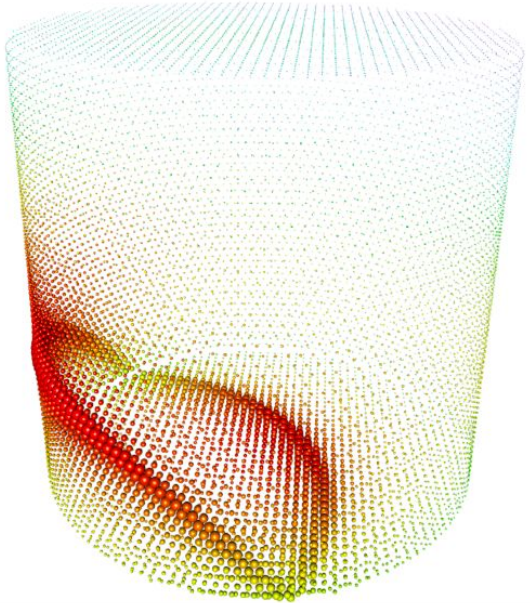
Following success of machine learning in other big data analysis

- **WatChMaL** group formed
  **Wat**er **Ch**erenkov **Ma**chine **L**earning

WatChMaL.org

Common challenges for ML with WC detectors

- Cylindrical geometry
- High-res, sparse data

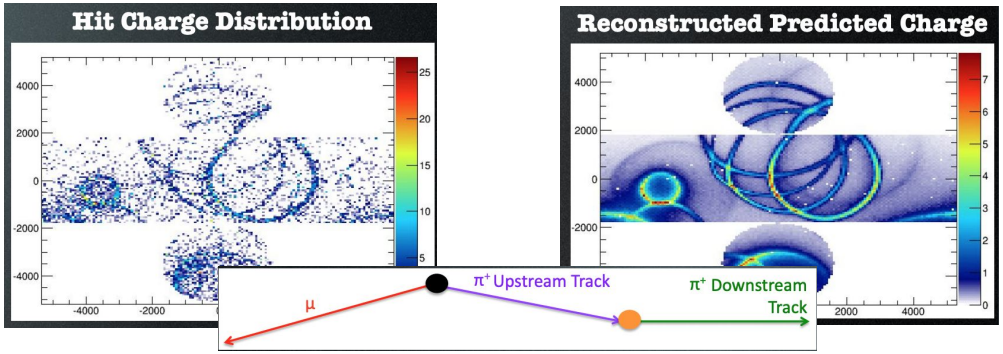


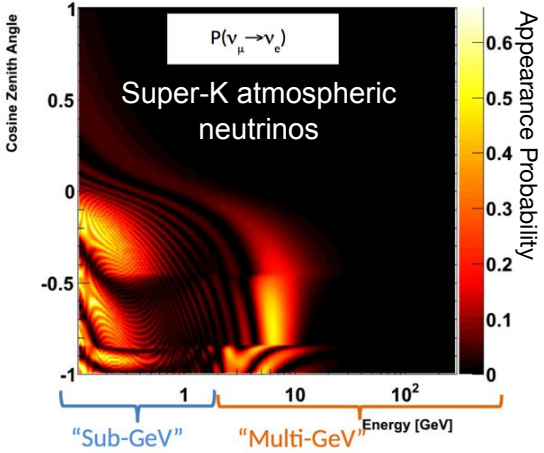
Many physics goals being explored

- Reconstruct complex event topologies
- Discriminate electron and gamma rings
- Improving detector calibration & systematics

# Physics goals

Complex event topologies: multi-GeV events

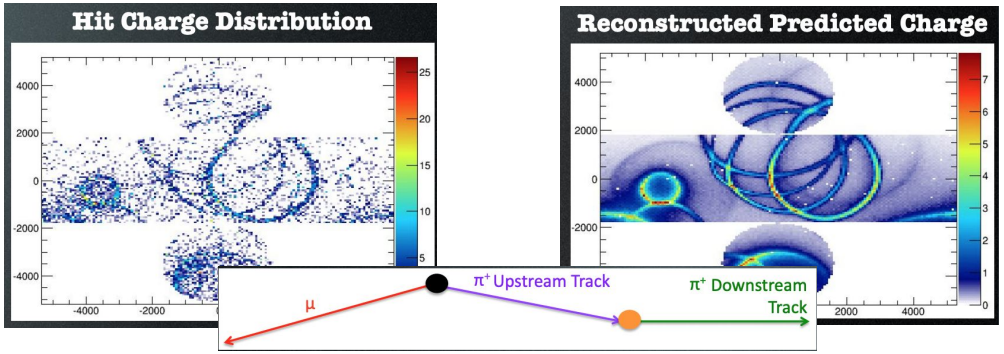


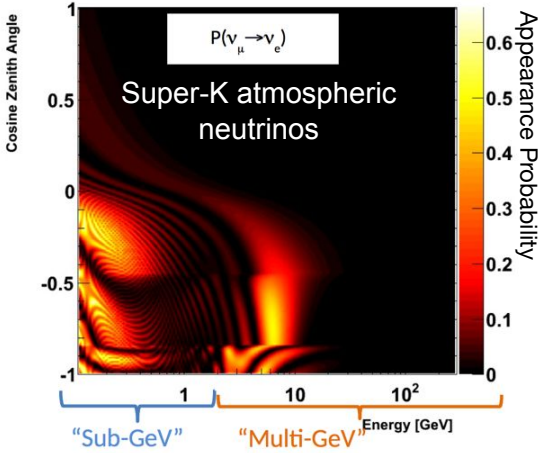



# Physics goals

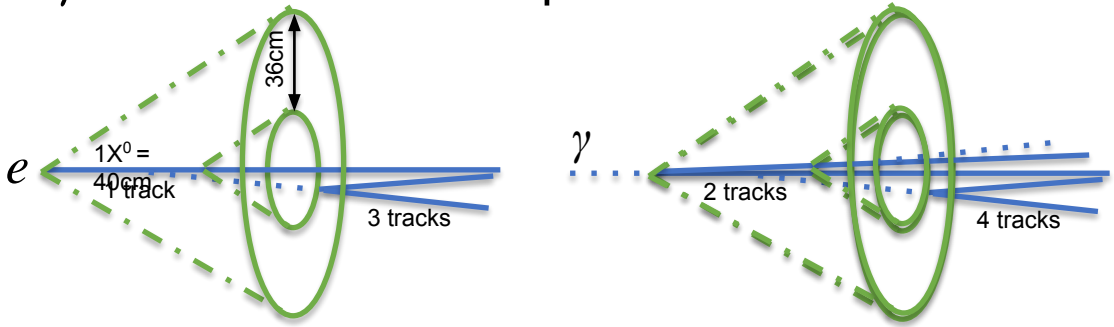
TRIUMF

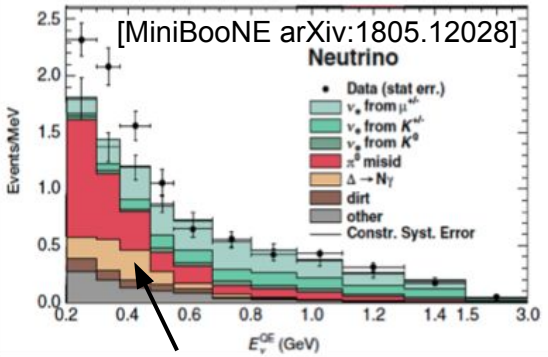
## Complex event topologies: multi-GeV events




## $e^-$ / $\gamma$ discrimination & improved $\pi^0$ identification



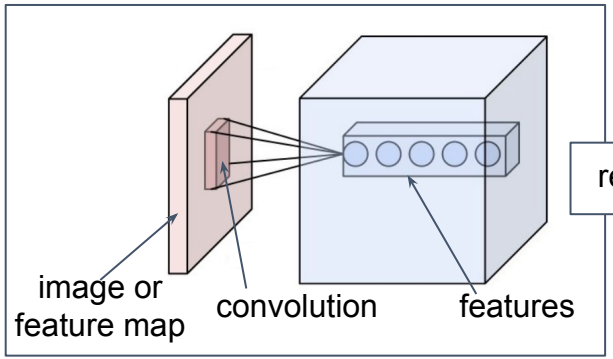




High uncertainty on NC$\gamma$ background

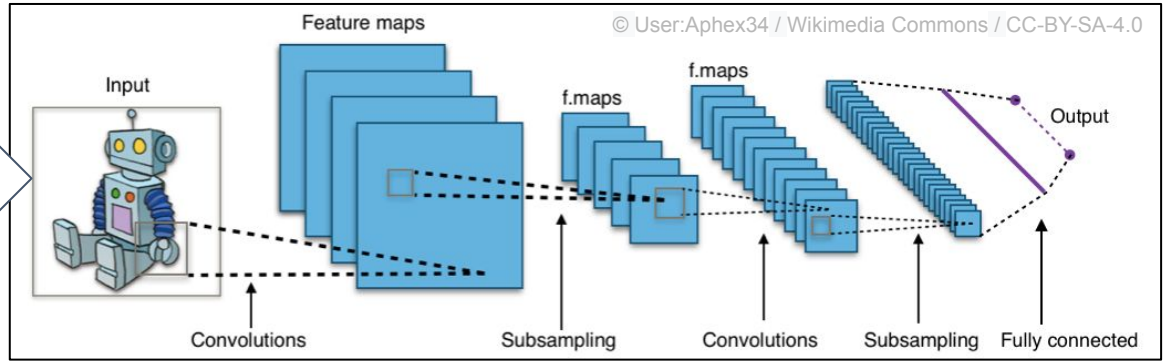
# CNN for $\mu^-$ / $e^-$ / $\gamma$ discrimination

TRIUMF

Convolutional neural networks hugely successful in image processing




repeat

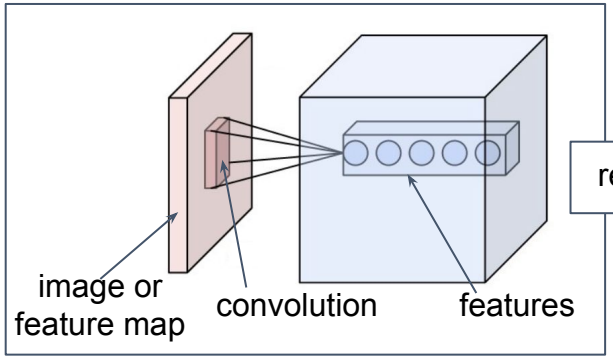



- Start with image with pixel values ('features'): T and Q at each PMT

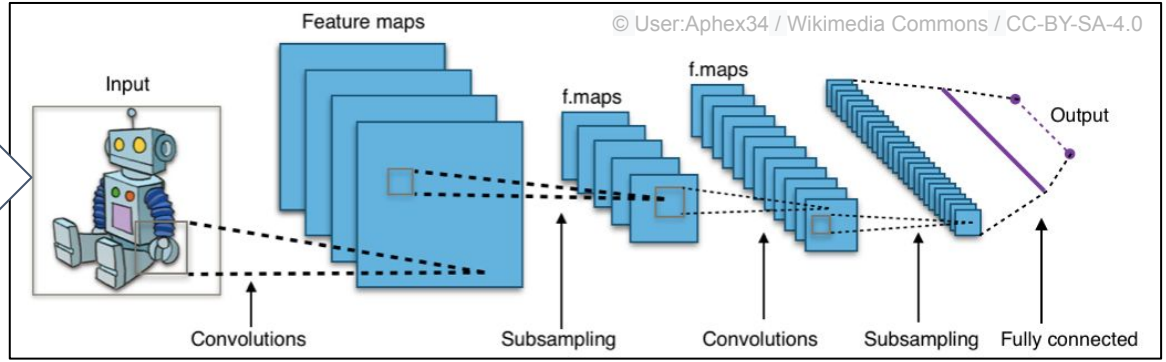
# CNN for $\mu^-$ / $e^-$ / $\gamma$ discrimination

TRIUMF

Convolutional neural networks hugely successful in image processing



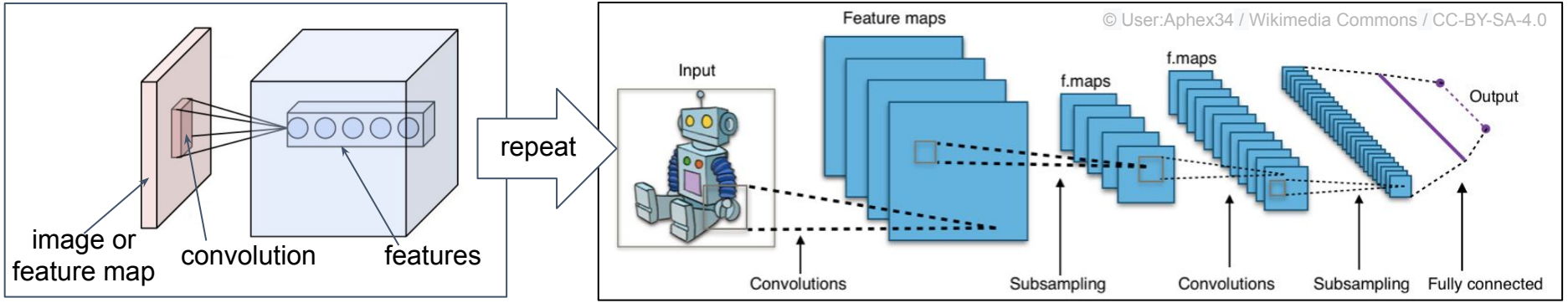

repeat




- Start with image with pixel values ('features'): T and Q at each PMT
- Scan many small (e.g. 3x3) convolution kernels across image
  - Increases number of features

# CNN for $\mu^-$ / $e^-$ / $\gamma$ discrimination

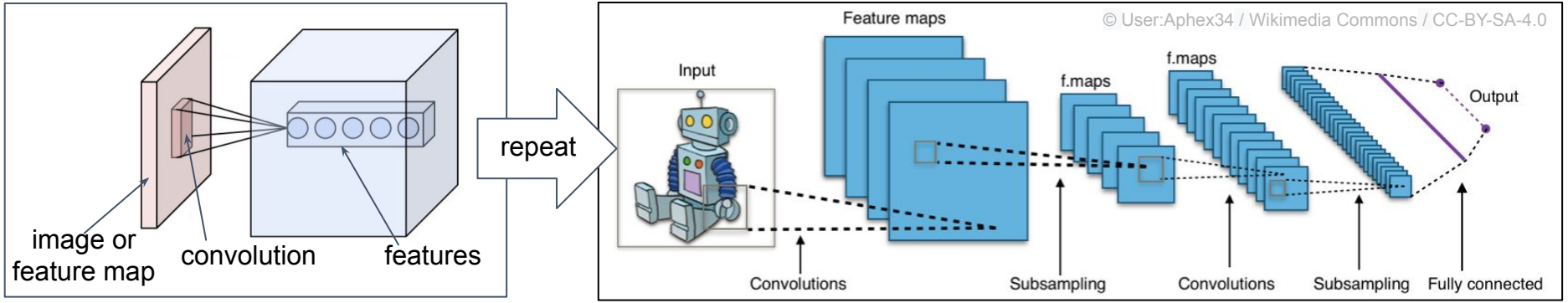
Convolutional neural networks hugely successful in image processing




- Start with image with pixel values ('features'): T and Q at each PMT
- Scan many small (e.g. 3x3) convolution kernels across image
  - Increases number of features
- Downsample image (e.g. 2x2 max-pooling)
  - Decreases number of pixels

# CNN for $\mu^-$ / $e^-$ / $\gamma$ discrimination

Convolutional neural networks hugely successful in image processing




- Start with image with pixel values ('features'): T and Q at each PMT
- Scan many small (e.g. 3x3) convolution kernels across image
  - Increases number of features
- Downsample image (e.g. 2x2 max-pooling)
  - Decreases number of pixels
- End with 1-D array of features, feed into traditional fully-connected neural network
- Learn convolution and final network weights through 'back-propagation' of loss

# CNN for $\mu^-$ / $e^-$ / $\gamma$ discrimination

Abhishek Abhishek,
Julian Ding, Wojtek Fedorko

Initial studies performed with Hyper-K intermediate detector simulations

- 3 million events each of $e^-$, $\mu^-$, $\gamma$ with $E_{kin} \in$ [0, 1 GeV]
- Particles (or $e^+ + e^-$ from $\gamma$) at tank centre, directed towards barrel wall
- End-caps not included here

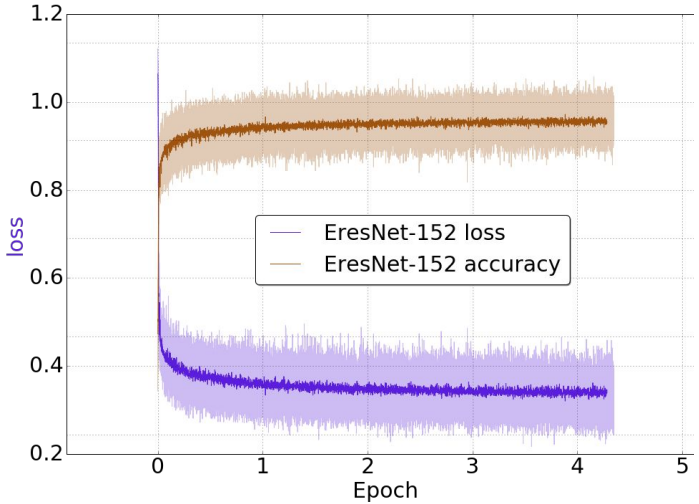
# CNN for $\mu^-$ / $e^-$ / $\gamma$ discrimination

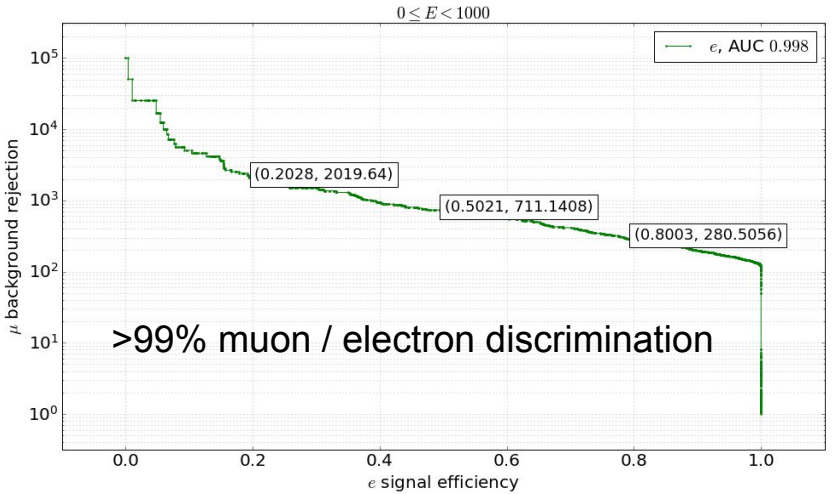
Abhishek Abhishek,
Julian Ding, Wojtek Fedorko

Initial studies performed with Hyper-K intermediate detector simulations

- 3 million events each of $e^-$, $\mu^-$, $\gamma$ with $E_{kin} \in [0, 1\ \text{GeV}]$
- Particles (or $e^+ + e^-$ from $\gamma$) at tank centre, directed towards barrel wall
- End-caps not included here

Various standard CNN architectures investigated for particle classification



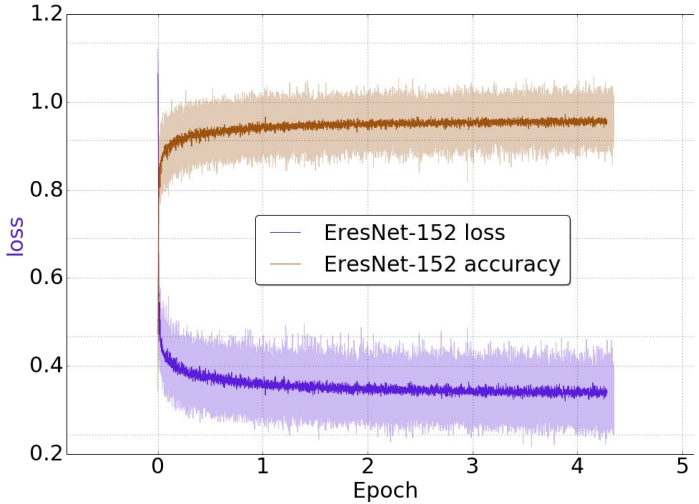
# CNN for $\mu^-$ / $e^-$ / $\gamma$ discrimination

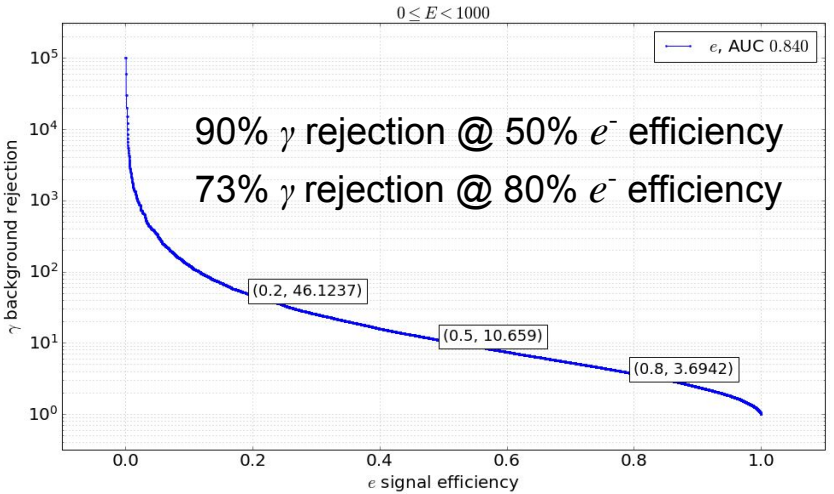
Abhishek Abhishek,
Julian Ding, Wojtek Fedorko

Initial studies performed with Hyper-K intermediate detector simulations

- 3 million events each of $e^-$, $\mu^-$, $\gamma$ with $E_{kin} \in [0, 1\ \text{GeV}]$
- Particles (or $e^+ + e^-$ from $\gamma$) at tank centre, directed towards barrel wall
- End-caps not included here

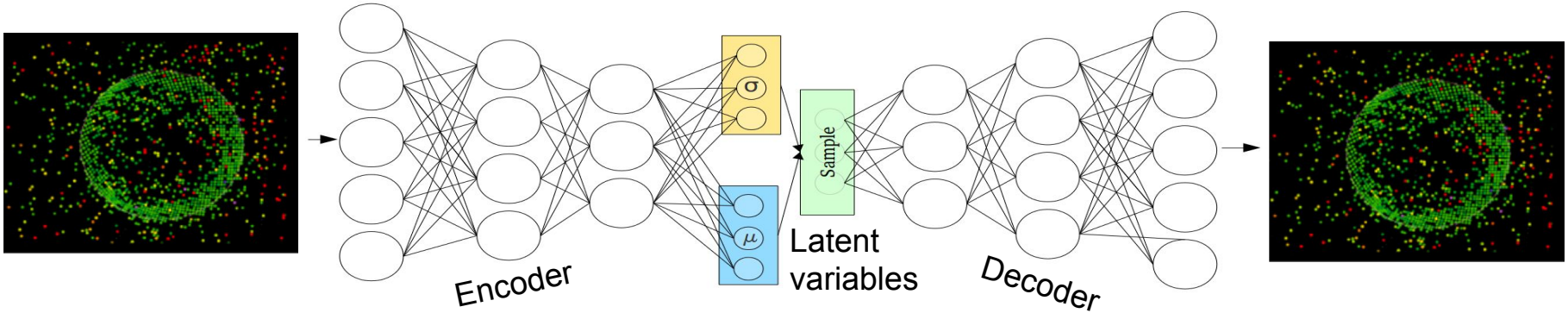
Various standard CNN architectures investigated for particle classification



# Variational autoencoders

Abhishek Abhishek
Wojtek Fedorko

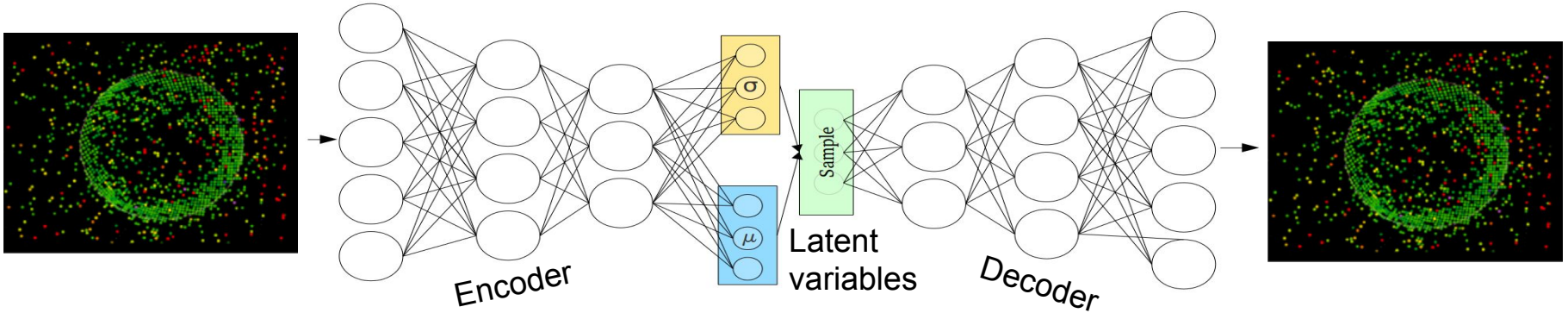



Design generative architecture to reproduce original event

- 1-D array from CNN encodes event information, train to decode this
- Can train to find features without any truth information

# Variational autoencoders

Abhishek Abhishek
Wojtek Fedorko




Design generative architecture to reproduce original event

- 1-D array from CNN encodes event information, train to decode this
- Can train to find features without any truth information

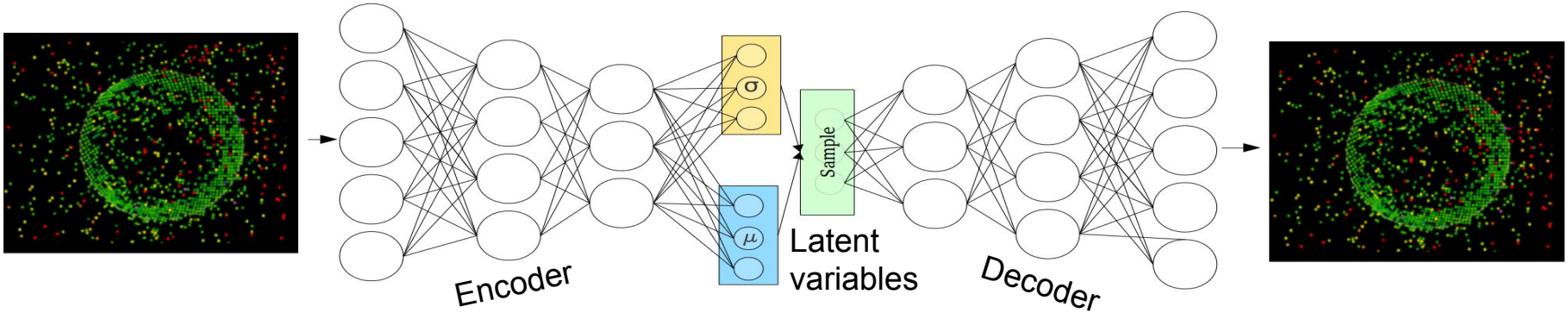
Train latent variables to be continuous Gaussian distributed variables

- Output two vectors: mean μ and variance σ
- Add Kullback–Leibler divergence term to loss

Hope that learnt latent space represents real physical quantities

# Variational autoencoders

Abhishek Abhishek
Wojtek Fedorko




Opens possibility for various new techniques:

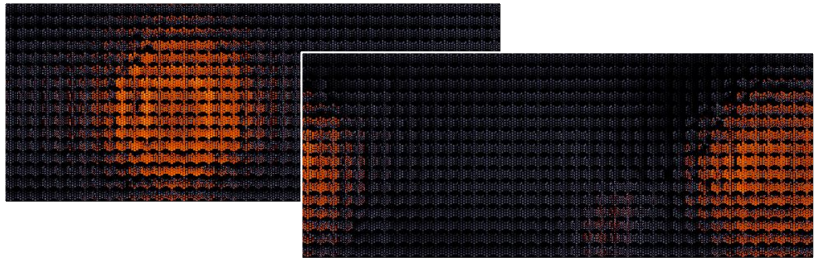
- Train on unlabeled real data, calibration samples, etc.
- Interpolate between events to for missing regions of parameter space
  - e.g. learn from limited control samples, understand full space
- Sample from distribution of latent variables to generate new events, from network trained fully on real data events
- Can still train classification, regression, etc. using latent variables

# Variational autoencoders

Abhishek Abhishek
Wojtek Fedorko

Early results show some promise

- Basic ring features seem to be captured
- Interpolation of energy working well
- But generated events still have some artificial artifacts



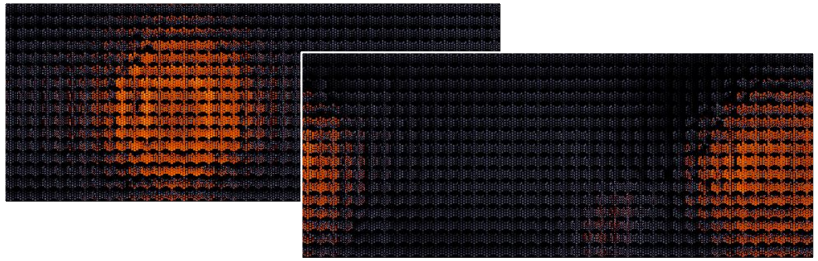
Randomly generated new events

# Variational autoencoders

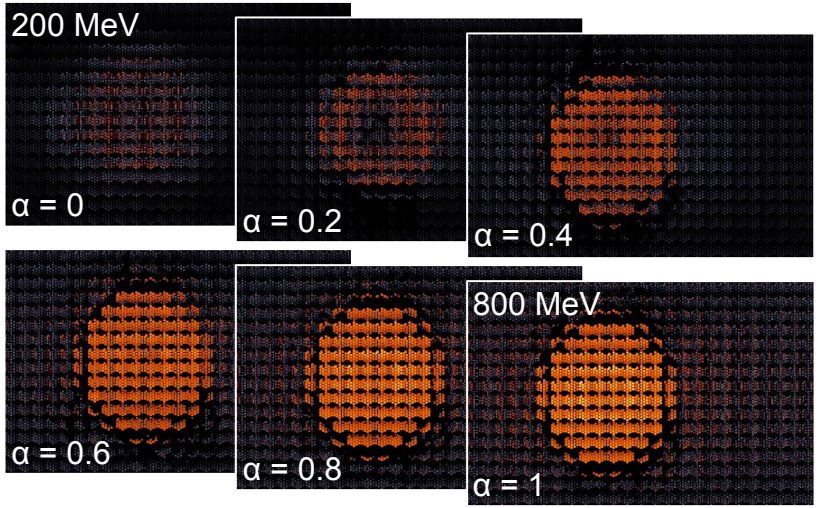
Abhishek Abhishek
Wojtek Fedorko

Early results show some promise

- Basic ring features seem to be captured
- Interpolation of energy working well
- But generated events still have some artificial artifacts



Randomly generated new events




Interpolate between 200 MeV and 800 MeV events

# Variational autoencoders

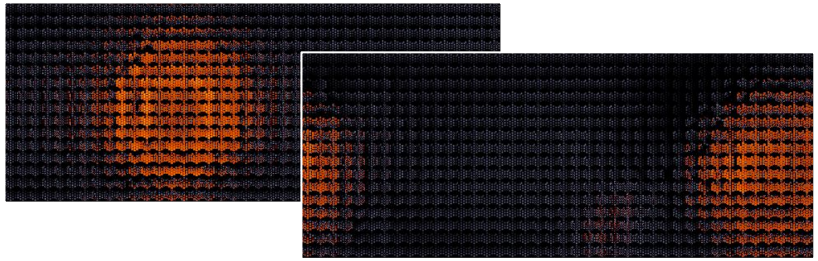
Abhishek Abhishek
Wojtek Fedorko

Early results show some promise

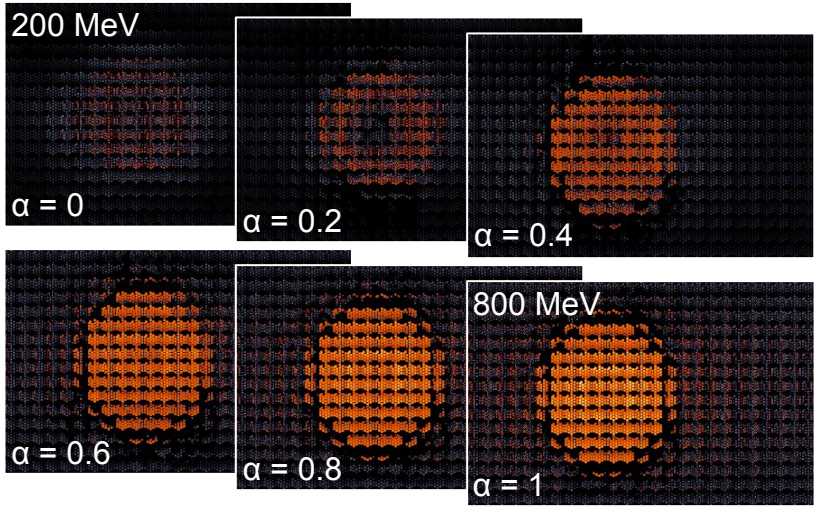
- Basic ring features seem to be captured
- Interpolation of energy working well
- But generated events still have some artificial artifacts

Started to investigate semi-supervised learning techniques

- Train using small labelled sample and large unlabelled sample
- Semi-supervised VAE already outperforms CNN with small labelled sample



Randomly generated new events




Interpolate between 200 MeV and 800 MeV events

# Variational autoencoders

Abhishek Abhishek
Wojtek Fedorko

Compare CNN with semi-supervised CNN (SS-CNN)

- Use limited amount of labelled data (11,250, 22,500 or 45,000 events)
- Use large amount of unlabelled data (7.2 million events)

# Variational autoencoders

Abhishek Abhishek
Wojtek Fedorko

Compare CNN with semi-supervised CNN (SS-CNN)

- Use limited amount of labelled data (11,250, 22,500 or 45,000 events)
- Use large amount of unlabelled data (7.2 million events)
- Small but significant improvement from semi-supervised approach
- Appear to get more benefit when demanding high efficiencies

| Number of training examples | $\gamma$ background rejection (%) at 50% $e^-$ signal efficiency | | $\gamma$ background rejection (%) at 80% $e^-$ signal efficiency | |
|---|---|---|---|---|
| | SS-CNN | CNN | SS-CNN | CNN |
| 11,250 | **77.6** | 76.4 | **50.7** | 46.3 |
| 22,500 | **80.4** | 78.1 | **54.3** | 48.5 |
| 45,000 | **80.7** | 79.4 | **55.9** | 49.9 |

See also: arXiv:1911.02369

# Full cylindrical geometry

Abhishek Abhishek
Wojtek Fedorko

Cylindrical geometry poses significant challenge to using standard CNNs

- Each wall represented by simple grid
- End-caps only fill portion of rectangle
- Connecting end-caps and barrel is challenging

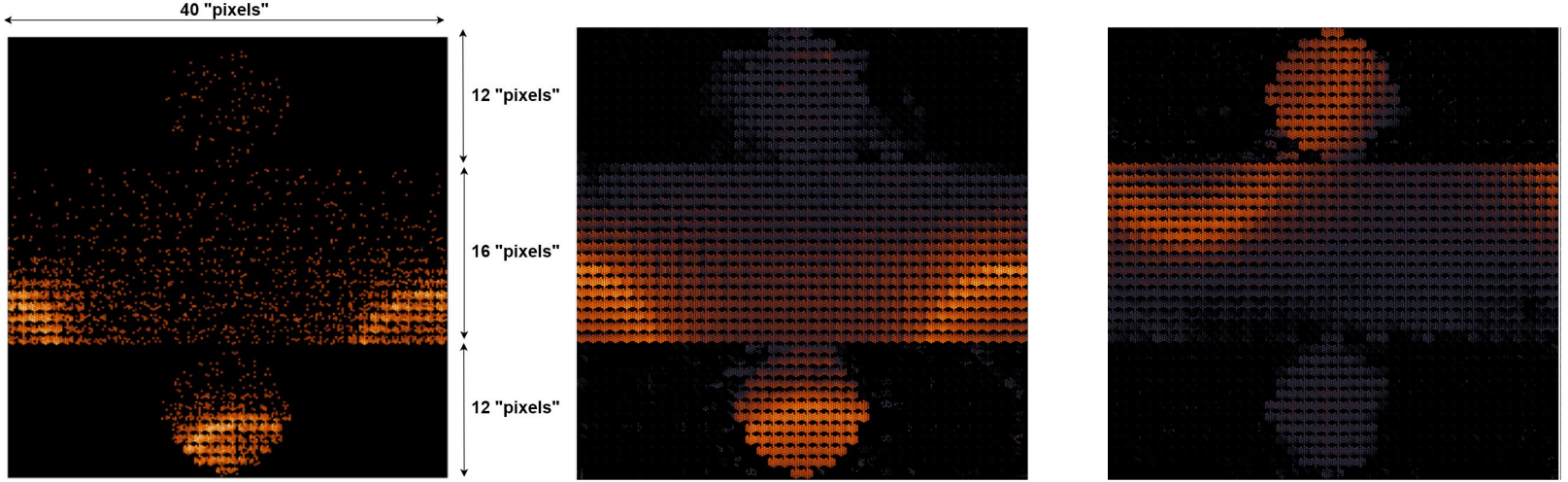
# Full cylindrical geometry

Abhishek Abhishek
Wojtek Fedorko

Cylindrical geometry poses significant challenge to using standard CNNs

- Each wall represented by simple grid
- End-caps only fill portion of rectangle
- Connecting end-caps and barrel is challenging

Initially try naive pasting each wall into flat image



# Geometrical CNNs

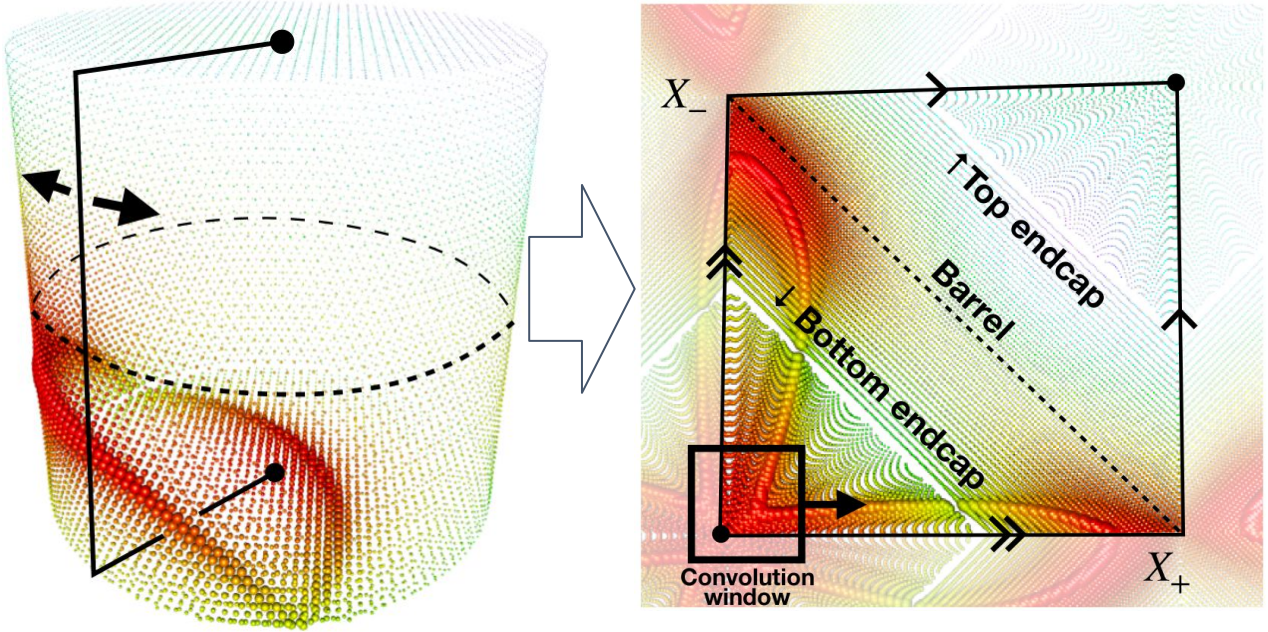
Lukas Berns
Tokyo Institute of Technology

- **Motivation:** want benefits of CNN in a network suitable for WC tank
  - Reduced # of trained weights from using convolutions
  - Capture geometry of cylinder like the translation invariance of CNN

# Geometrical CNNs

Lukas Berns
Tokyo Institute of Technology

- **Motivation:** want benefits of CNN in a network suitable for WC tank
  - Reduced # of trained weights from using convolutions
  - Capture geometry of cylinder like the translation invariance of CNN




Transform cylindrical surface on to rectangle

- Gives physical boundary conditions
- Keep uniform density of PMTs
- Place new points onto grid points

# Geometrical CNNs

Lukas Berns
Tokyo Institute of Technology

**Taylor convolutions - motivation:**

CNN applies convolution in e.g. 3x3 local area on grid

How can we do similar using physical 3-D space?

# Geometrical CNNs

Lukas Berns
Tokyo Institute of Technology

**Taylor convolutions - motivation:**

CNN applies convolution in e.g. 3x3 local area on grid

How can we do similar using physical 3-D space?

**What information is used by CNNs' convolutions?**

At each point, function value at and near that point on previous layer

# Geometrical CNNs

Lukas Berns
Tokyo Institute of Technology

**Taylor convolutions - motivation:**

CNN applies convolution in e.g. 3x3 local area on grid

How can we do similar using physical 3-D space?

**What information is used by CNNs' convolutions?**

At each point, function value at and near that point

**Taylor expansion of function describes same local information**

Define 'convolution' using taylor expansion in physical spatial dimensions

- Learn coefficients corresponding to function value, derivatives
- Apply convolution-like filter on grid points, weighted by pre-computed weights according to taylor expansion in physical space

# Geometrical CNNs

Lukas Berns
Tokyo Institute of Technology

Learned weights are now geometrical objects, can transform to enforce invariance to translations, rotations, etc.

- Apply random rotations about tank axis during training enforces tank's rotational symmetry
- Reduced degrees of freedom should help to train from few events

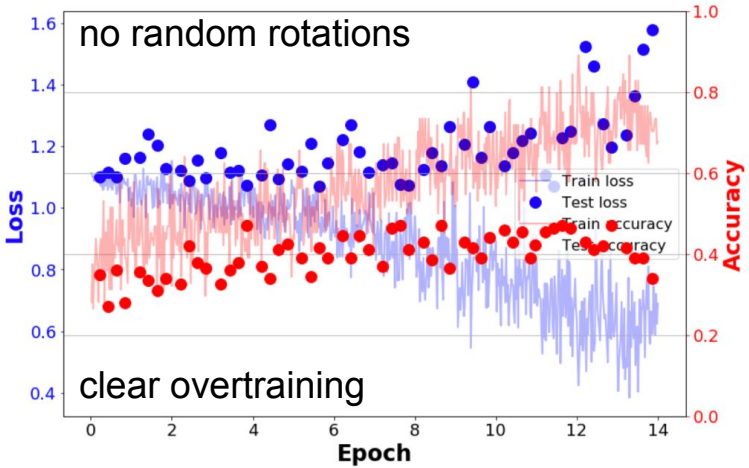
# Geometrical CNNs

Lukas Berns
Tokyo Institute of Technology

Learned weights are now geometrical objects, can transform to enforce invariance to translations, rotations, etc.

- Apply random rotations about tank axis during training enforces tank's rotational symmetry
- Reduced degrees of freedom should help to train from few events

Trained on only 1000 events



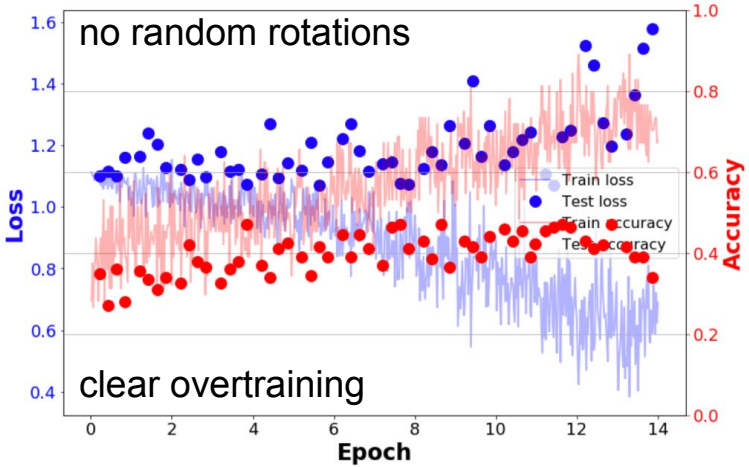
# Geometrical CNNs

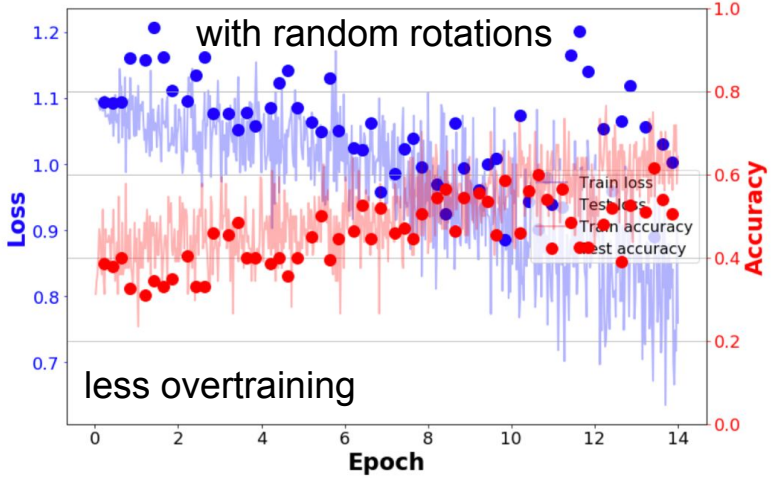
Lukas Berns
Tokyo Institute of Technology

Learned weights are now geometrical objects, can transform to enforce invariance to translations, rotations, etc.

- Apply random rotations about tank axis during training enforces tank's rotational symmetry
- Reduced degrees of freedom should help to train from few events

Trained on only 1000 events



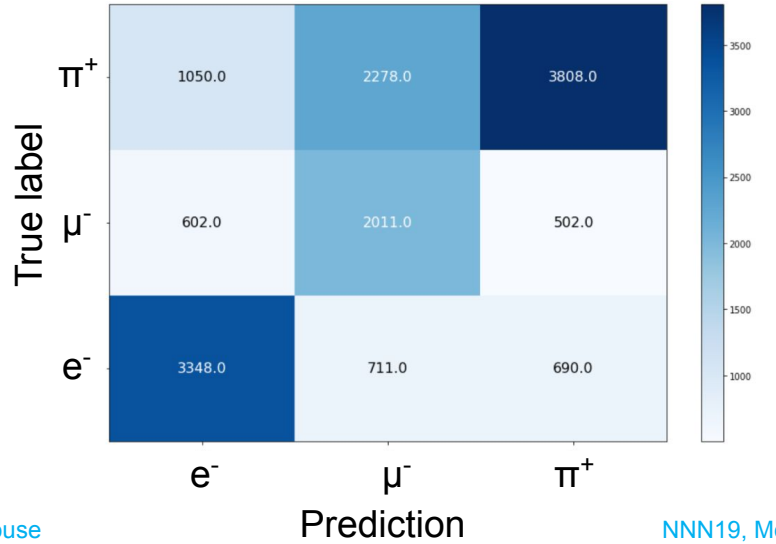



# Geometrical CNNs

Lukas Berns
Tokyo Institute of Technology

Learned weights are now geometrical objects, can transform to enforce invariance to translations, rotations, etc.

- Apply random rotations about tank axis during training enforces tank's rotational symmetry
- Reduced degrees of freedom should help to train from few events




Some discrimination even training on only 1000 events

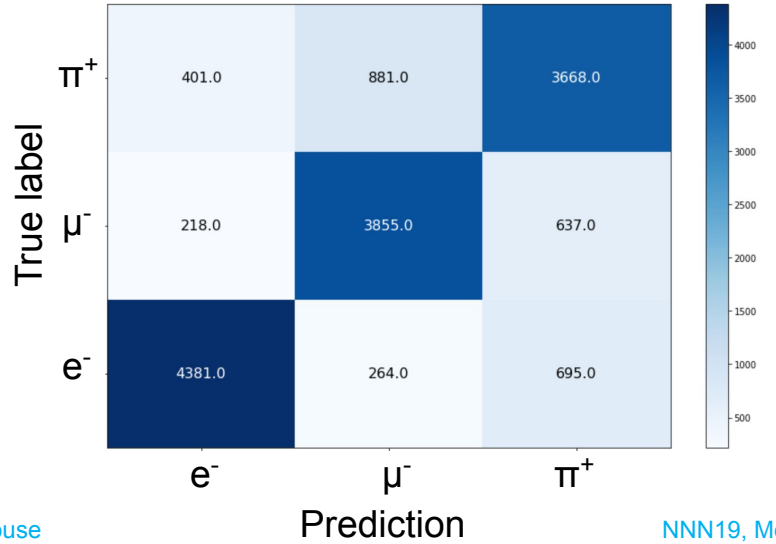
- Uniformly distributed in tank
- Isotropic directions

# Geometrical CNNs

Lukas Berns
Tokyo Institute of Technology

Learned weights are now geometrical objects, can transform to enforce invariance to translations, rotations, etc.

- Apply random rotations about tank axis during training enforces tank's rotational symmetry
- Reduced degrees of freedom should help to train from few events




Better with 50,000 events,
but expect further improvements

# Other architectures

TRIUMF

Also investigating other architectures suitable for cylindrical tanks

**Point cloud NNs**

- Each hit is point in 4-D space
- Learns symmetric functions on unordered collection of points
- Easily adapt to any geometry
- Invariant to spatial rotations, translations

# Other architectures

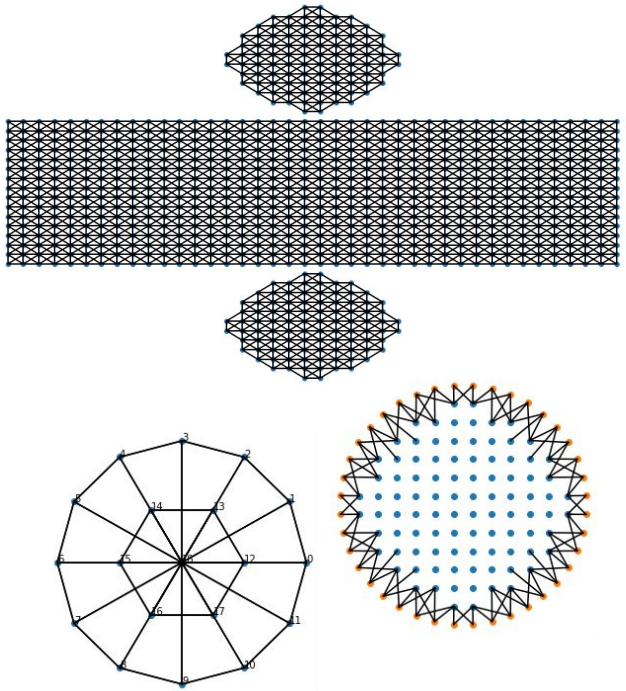
Also investigating other architectures suitable for cylindrical tanks

## Point cloud NNs

- Each hit is point in 4-D space
- Learns symmetric functions on unordered collection of points
- Easily adapt to any geometry
- Invariant to spatial rotations, translations

## Graph CNNs

- PMTs are nodes connected in graph
- Adjacency matrix defines connections
- Learn filters applied at each node
- Good for complex geometries (cylinder, mPMTs)



# Summary

Several physics goals where ML can allow new / improved analyses

- Discriminating particle types not previously possible
- Reconstructing new complex event topologies

# Summary

TRIUMF

Several physics goals where ML can allow new / improved analyses

- Discriminating particle types not previously possible
- Reconstructing new complex event topologies

WatChMaL group starting to explore many areas

- Sharing experience across collaborations
- Common approach to deal with detector geometry, etc.

# Summary

TRIUMF

Several physics goals where ML can allow new / improved analyses

- Discriminating particle types not previously possible
- Reconstructing new complex event topologies

WatChMaL group starting to explore many areas

- Sharing experience across collaborations
- Common approach to deal with detector geometry, etc.

Initial results look promising

- Electron / gamma discrimination is working!
- Generative techniques for semi or unsupervised learning
- Geometrical CNNs and other architectures to work with cylindrical tank

# Summary

TRIUMF

Several physics goals where ML can allow new / improved analyses

- Discriminating particle types not previously possible
- Reconstructing new complex event topologies

WatChMaL group starting to explore many areas

- Sharing experience across collaborations
- Common approach to deal with detector geometry, etc.

Initial results look promising

- Electron / gamma discrimination is working!
- Generative techniques for semi or unsupervised learning
- Geometrical CNNs and other architectures to work with cylindrical tank

**Thank you for your attention!**

# Additional slides

# CNN (& VAE) Architecture

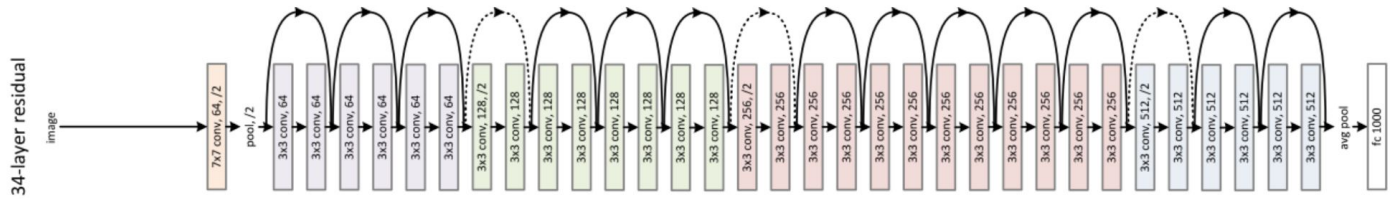
Abhishek Abhishek,
Julian Ding, Wojtek Fedorko

Originally introduced in arXiV:1512.03385, the **residual learning framework** allows for development of very deep CNN architectures.

Modified the original **ResNet** architecture to allow for usage in the VAE framework with a asymmetric decoder.

Some of important changes include :

1. Removing the large $7 \times 7$ convolutional layer and max-pooling at the start
2. Removing the average pooling layer at the end
3. Modifying the bottleneck used in the deeper architectures to allow for reversibility of the VAE encoder



# CNN & VAE Architecture

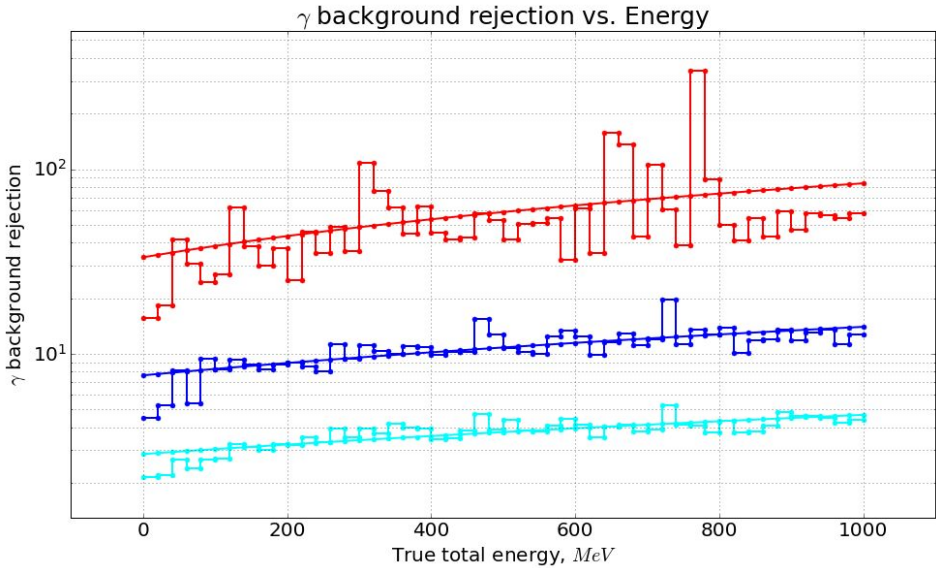
Abhishek Abhishek,
Julian Ding, Wojtek Fedorko

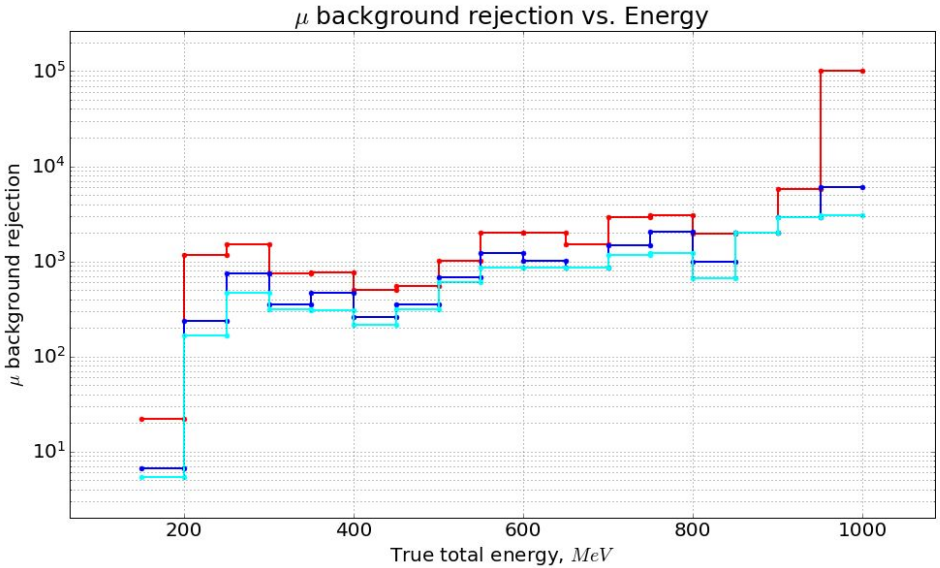
$e^- - \mu^- - \gamma$ classification results using the **IWCD** geometry show significant improvement over the shallower **LeNet** architecture as well as previous best results :

| Model | $\gamma$ Rejection at 20% e signal efficiency | γ Rejection at 50% e signal efficiency | γ Rejection at 80% e signal efficiency | AUC |
|---|---|---|---|---|
| LeNet-9 CNN | 95 (20.23) | 85.5 (7.01) | 64.6 (2.86) | 0.780 |
| ResNet-50 CNN | 97.9 (48.80) | 90.4 (10.55) | 72.0 (3.64) | 0.836 |
| ResNet-101 CNN | 97.8 (46.26) | 90.3 (10.46) | 72.5 (3.69) | 0.836 |
| ResNet-152 CNN | **97.9 (49.36)** | **90.7 (11.04)** | **73.2 (3.80)** | **0.841** |
| DenseNet-121 CNN | 97.7 (44.01) | 89.7 (9.75) | 70.1 (3.35) | 0.823 |

# CNN vs Energy

Abhishek Abhishek,
Julian Ding, Wojtek Fedorko



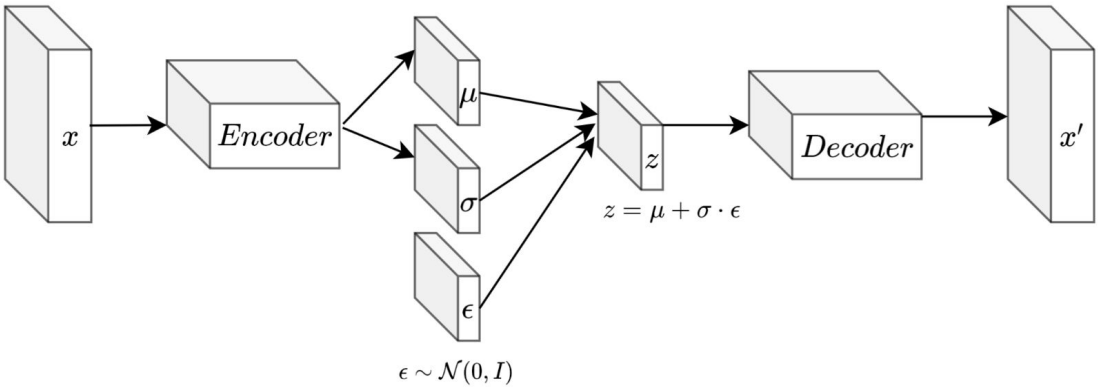
# VAE details

Abhishek Abhishek,
Julian Ding, Wojtek Fedorko

A **Variational AutoEncoder (VAE)** is a generative model where the goal is to learn a model distribution $p_{model}(x)$ which resembles the data distribution $p_{data}(x)$ as closely as possible for some high-dimensional data $x \in \mathcal{X}$.

It achieves this by maximizing a lower bound (ELBO) to the true log-likelihood $log(p_{\theta,\phi}(x))$ of the data under the model :

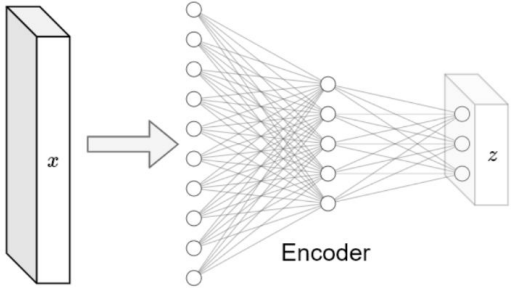
$$\underbrace{log(p_{\theta,\phi}(x))}_{\text{True log-likelihood}} \geq \underbrace{E_{z\sim q_\phi(z|x)}[log(p_\theta(x|z))]}_{\text{MSE Loss}} - \underbrace{D_{KL}[q_\phi(z|x)||p(z)]}_{\text{KL Loss}} \quad (1)$$

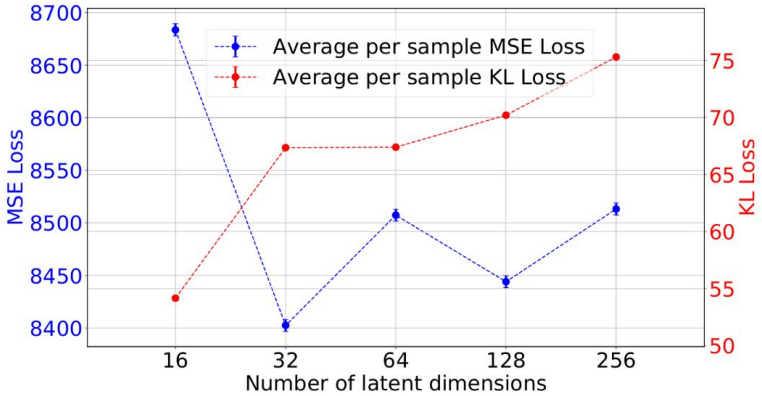


# Variational Autoencoder : Latent dimensions

All of the information about the event display is captured using low dimensional **latent variables**. It is not trivial to determine theoretically the optimal dimensionality of these variables.

We compared VAEs using different number of latent dimensions in terms of the MSE loss and KL divergence loss :



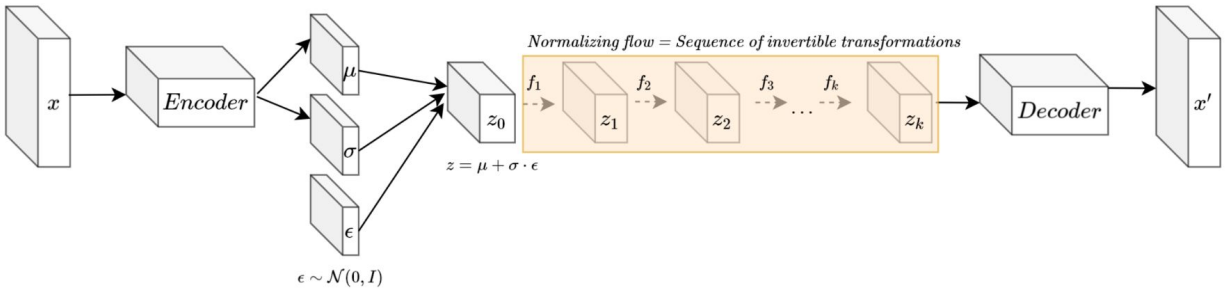
# Normalizing flows

Basic idea : Define a sequence of invertible mappings $f_l$ to transform a random variable $z_0$ with probability density $q_0(z_0)$ into another random variable $z_k$ with probability density $q_k(z_k)$.

The probability density of the random variable $z_{l+1}$ after each transformation $f_l$ is given by :

$$q_{l+1}(z_{l+1}) = q_l(z_l)\left|det\frac{\partial f_l^{-1}}{\partial z_{l+1}}\right| = q_l(z_l)\left|det\frac{\partial f_l}{\partial z_l}\right|^{-1} \quad (2)$$

The above is derived using the jacobian method for transformation of random variables. See **watchmal.org** for more details.



# Finite normalizing flows

TRIUMF

Composing several simple maps and successively applying the transformation on the probability density of the resulting random variables allows us to construct arbitrarily complex densities.

$$z_k = f_k \circ f_{k-1} \circ ... \circ f_2 \circ f_1(z_0) \tag{3}$$

Using equation (2), the probability density of this resulting random variable $z_k$ is given by :

$$q_k(z_k) = \frac{q_0(z_0)}{\prod_{l=1}^{k} \left| det \frac{\partial f_l}{\partial z_l} \right|} \tag{4}$$

The log-probability density of $z_k$ is then :

$$ln(q_k(z_k)) = ln(q_0(z_0)) + \sum_{l=1}^{K} ln(\left| det \frac{\partial f_l}{\partial z_l} \right|) \tag{5}$$

# Effect of normalizing flows

The flows can be interpreted as a sequence of expansions and contractions on the initial density of the latent variables. In the two cases :

1. **Expansion** : $f_l$ pulls the points $z$ away from a region in $R^d$ reducing the density in that region while increasing the density outside that region.
2. **Contraction** : $f_l$ pushes the points $z$ away towards a region in $R^d$ increasing the density in that region while reducing the density outside that region.

This allows use to start with simple approximate posterior distributions $q_\phi(z|x)$ e.g. mean-field approximations and systematically achieve complex and multi-modal distributions.

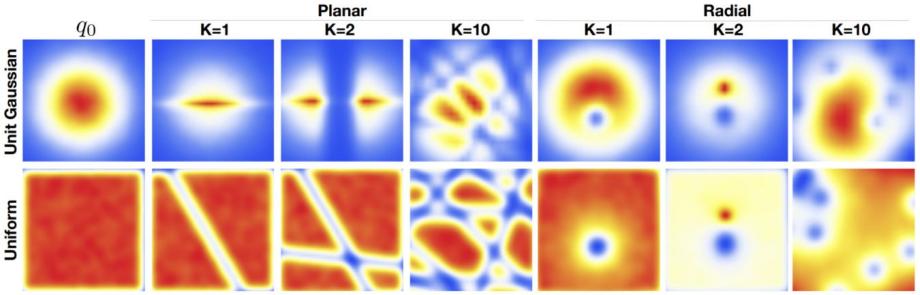
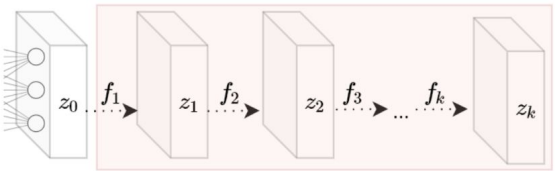


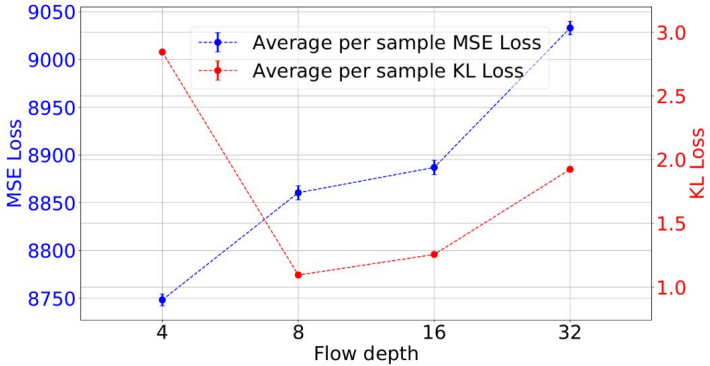

**Figure:** Effect of planar and radial normalizing flows on unit gaussian and uniform distributions. Source : **arXiv:1505.05770**

# Normalizing Flow : Flow depths

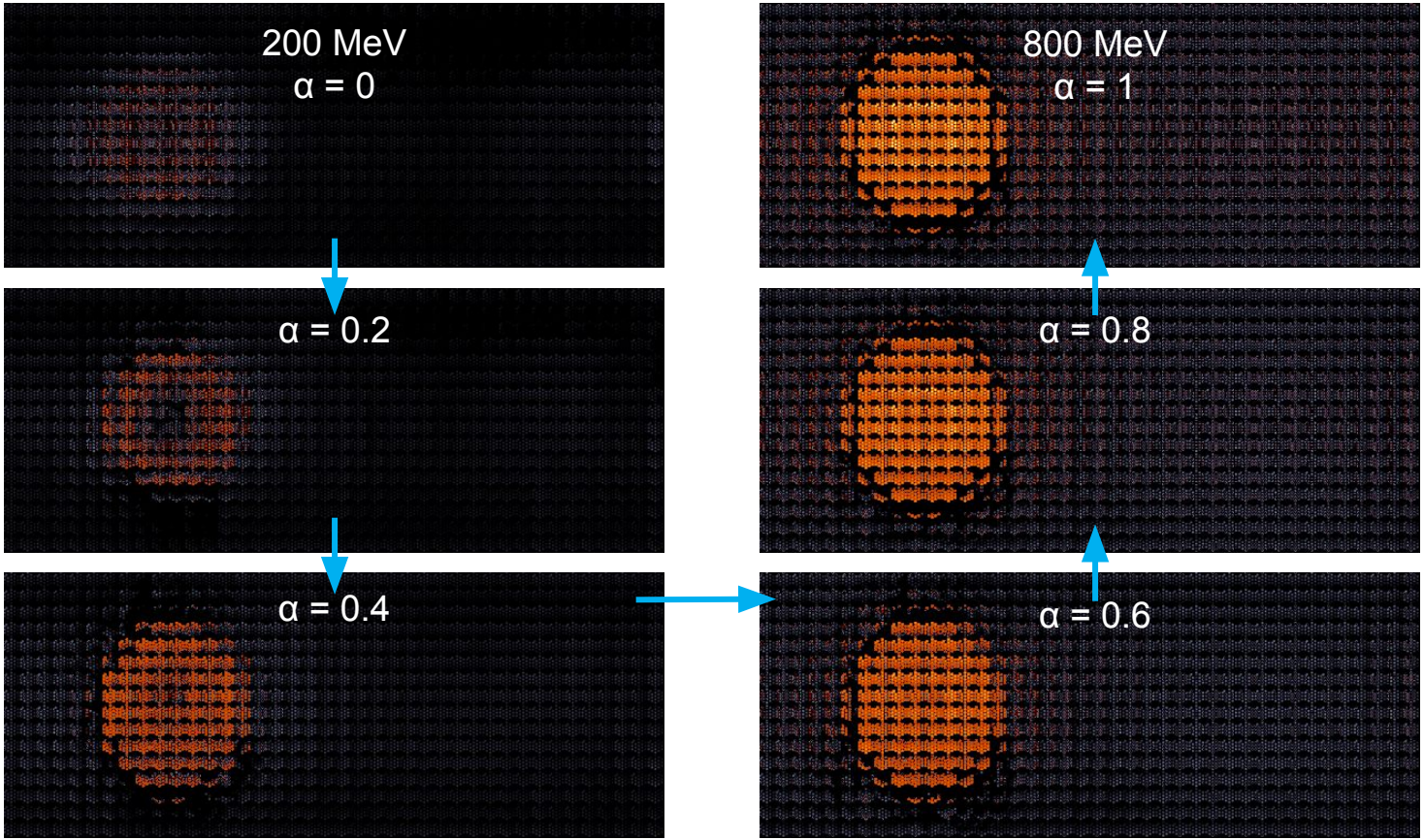
**Normalizing Flows** are an extension of VAE where a sequence of **invertible transformations** are applied to **latent variables** to achieve a more complex distribution (unlike $\mathcal{N}(z|\mu_\phi(x), diag(\sigma^2_\phi(x)))$ used in the VAE.

The number of these invertible transformations in the sequence is a hyper-parameter known as **flow depth**. We compared planar normalizing flow with varying flow depths :

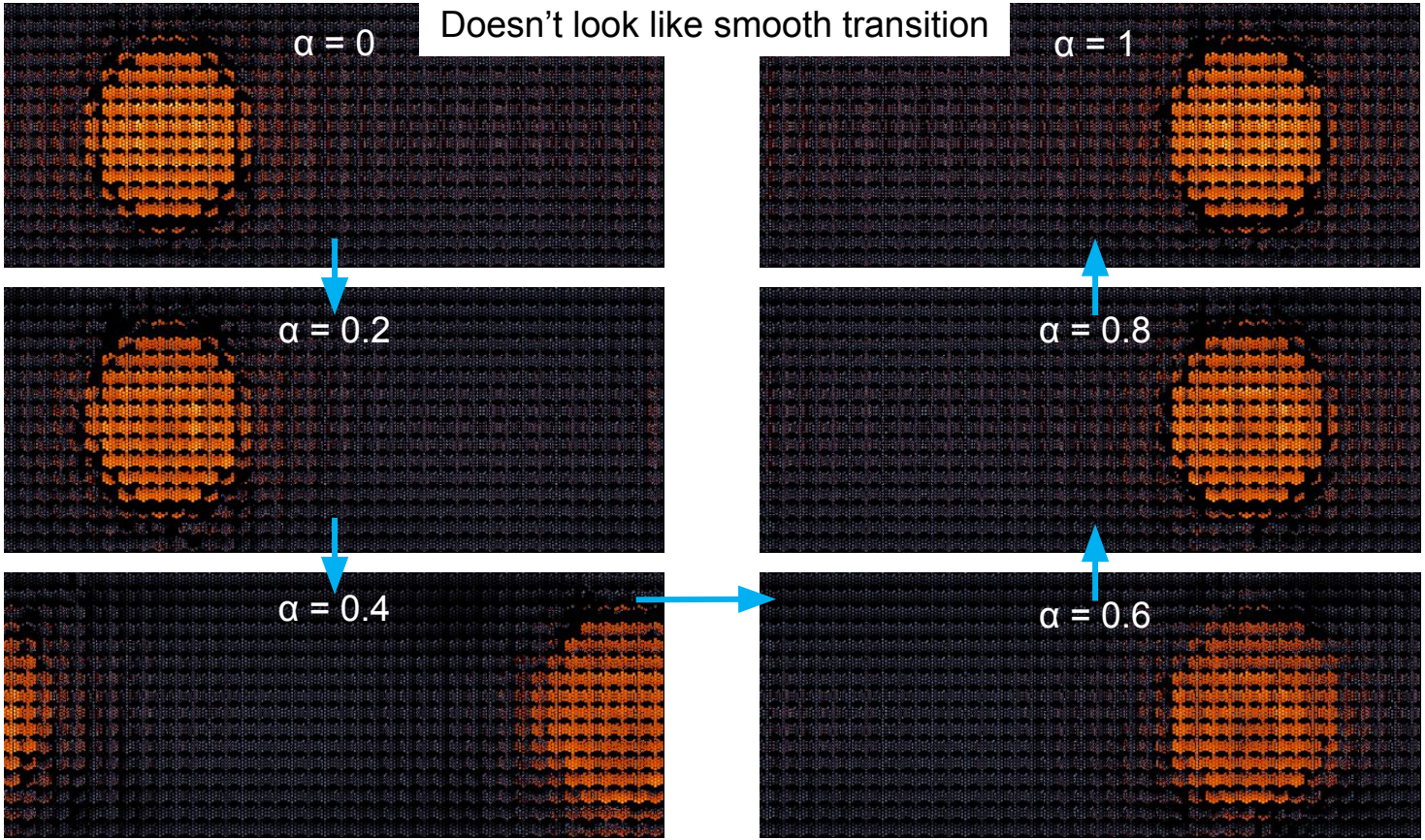


# VAE: Interpolating energy



# VAE: Interpolating tank azimuth



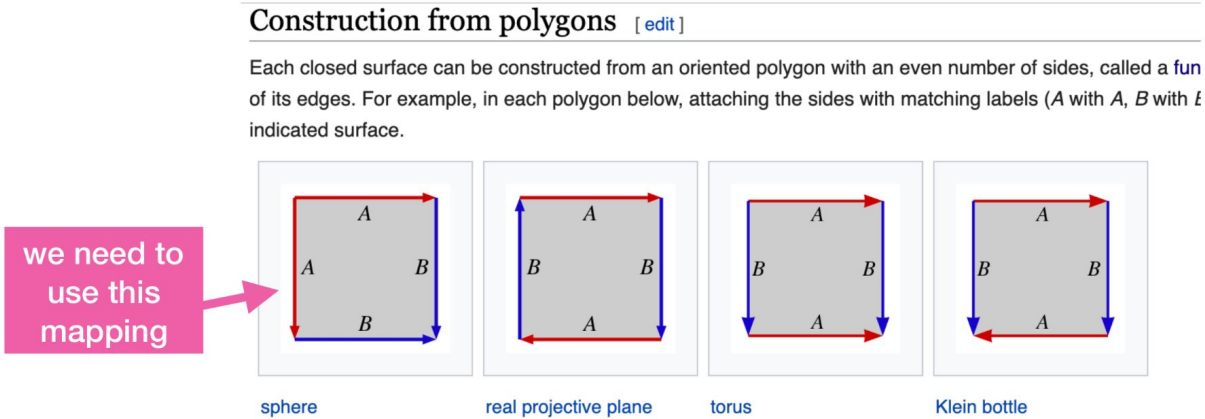
# Embedding a cylinder surface in a plane

Lukas Berns
Tokyo Institute of Technology

- Topologically a cylinder surface is a 2D surface isomorphic to a 2-sphere

- Every closed 2-sphere can be constructed from a 2D plane by identifying the sides appropriately:

we need to use this mapping



# Mapping cylinder surface to a plane

Lukas Berns
Tokyo Institute of Technology

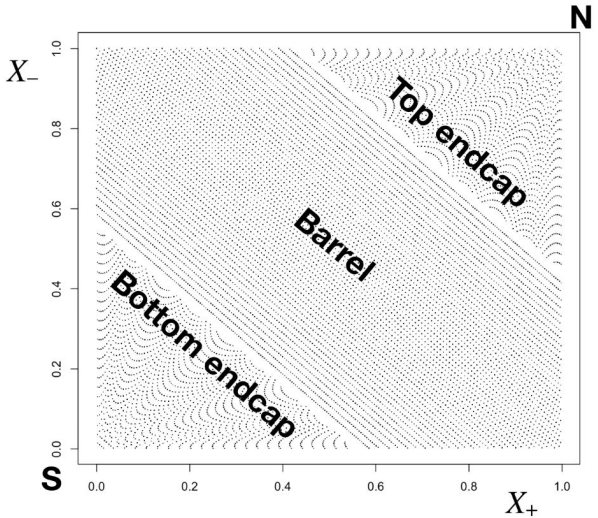
- Map cylinder surface to square (fundamental polygon). $\rho$ and $z$ give (via W) position along (0,0) to (1,1) diagonal, ϕ along perpendicular axis.

$$X_{\pm} = W(\rho, z)\frac{\pi \pm \phi}{2\pi}$$

- Uniform mapping is achieved by demanding uniform density for all ρ,z. By solving differential equation from demanding constant Jacobian we obtain:

$$\mathrm{d}X_{+}\,\mathrm{d}X_{-} = \left|\frac{\partial(X_{+}, X_{-})}{\partial(\rho, \phi)}\right| \mathrm{d}\rho\,\mathrm{d}\phi$$

$$W(\rho, z) = \sqrt{\frac{\rho^2 + 2Rz + RH}{R^2 + RH}}$$



# Putting PMTs on a grid

Lukas Berns
Tokyo Institute of Technology

Prepare square grid with size $\sqrt{N_{PMT}}$ + (0~2)

1. Sort by

   additional space helps better arrangement ↑

$$\alpha\left(\frac{|z|}{H/2}-\frac{\rho}{1\,\mathrm{m}}\right)+\mathrm{rand}()$$

   where $\alpha = 10$ and $H$ is tank height.

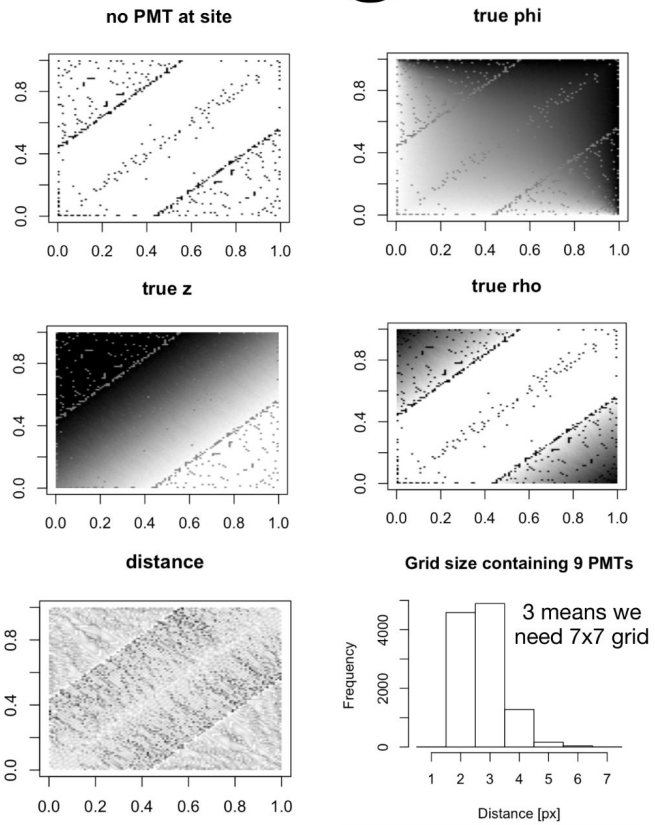
2. Assign to closest empty grid site with distance given by formula:

$$(\Delta W)^2+\beta(\Delta P)^2$$

   where $\beta = 3$ and

$$W := X_+ + X_-$$
$$P := X_+ - X_-$$

Very likely there are better algorithms. Also α,β can be optimized if necessary.



# Geometrical CNNs

Lukas Berns
Tokyo Institute of Technology

- Translation of convolutions across grid not physically meaningful
- Local convolution kernels can be thought of as acting on the function value and discrete derivatives at each point
- Instead of learning convolution kernels on grid, learn coefficients applied to Taylor expansion with respect to spatial dimensions

$$N_i^{(n+1)} = \sum_j K_{j-i} N_j^{(n)}$$

output layer ← input layer

Standard CNN: $K$ depends on relative position of input and output sites $j$-$i$

$$N^{(n+1)}(x) = \left( K + K^I \partial_I + \frac{1}{2} K^{IJ} \partial_I \partial_J \right) N^{(n)}(x)$$

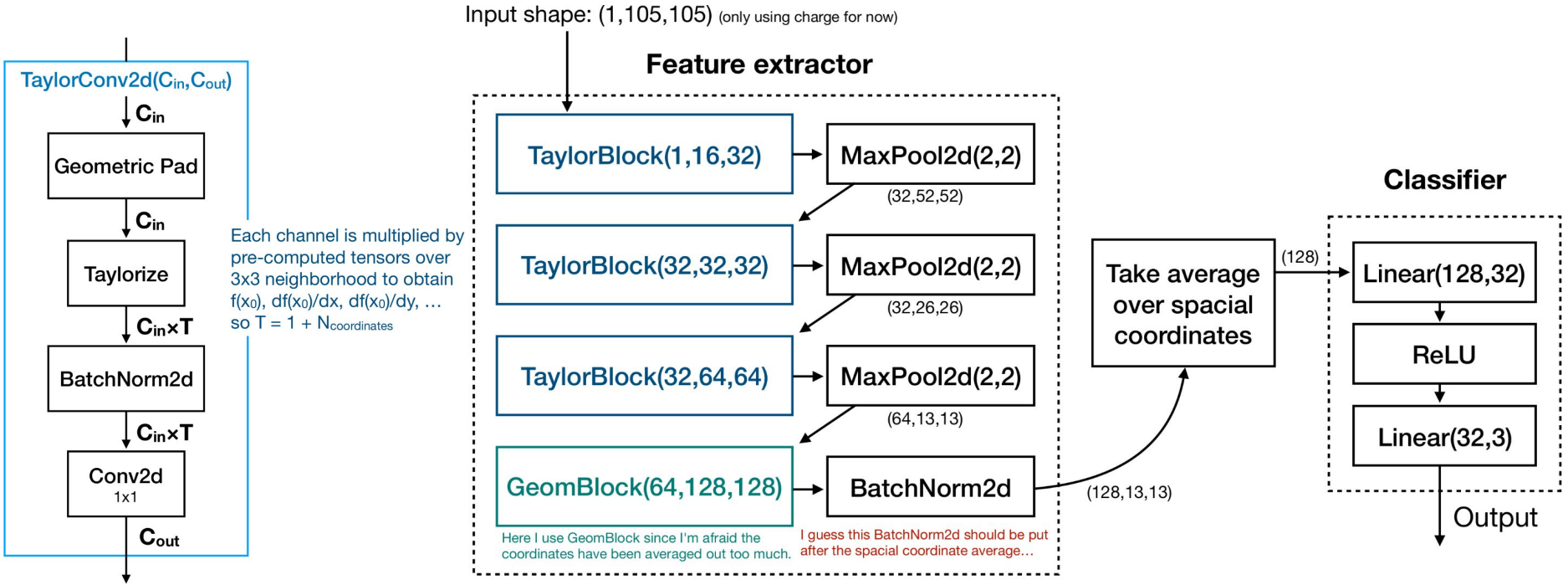
Geometrical CNN: $K$, $K^I$, $K^{IJ}$ are Taylor expansion coefficients as scalars, vectors, tensors in physical space

$$N_i^{(n+1)} = \sum_j \left( K w_{ij} + K^I w_{I,ij} + \frac{1}{2} K^{IJ} w_{IJ,ij} \right) N_j^{(n)}$$

Use precomputed weights $w$, $w_{I,ij}$, $w_{IJ,ij}$ to parametrise how learned coefficients $K$, $K^I$, $K^{IJ}$ act on the 2-D grid

# Network architecture

Lukas Berns
Tokyo Institute
of Technology




For the Taylor blocks after the max pooling I use the coordinates passed through AvgPool2d to derive the weights.