CZECH TECHNICAL UNIVERSITY IN PRAGUE

CERN openlab

Transformer-based Models for Detector Simulations

Stochastic and Physical Monitoring Systems 2023
Sloup v Čechách

Kristina Jaruskova (FNSPE CTU in Prague and CERN)

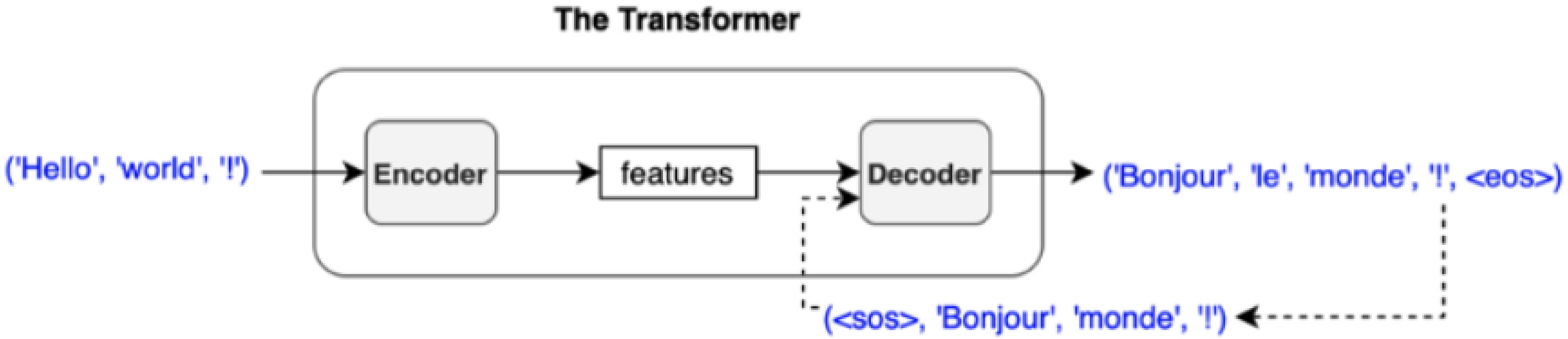
Transformers

- Transformer – neural network based on attention mechanism
- Typical use case – processing sequential data
  - Natural language processing (NLP) – translation, text generation – state-of-the-art
- Other models used for sequences – recurrent networks (RNNs), convolutional networks (CNNs)
- Advantages:
  - no inductive bias → better for modelling long-range dependencies
  - fast to compute (great parallelization)
- Cons: in general, needs more training data to outperform RNNs

- First introduced in "Attention Is All You Need" by A. Vaswani et al. (Google, 2017)
  - English-to-French translation
- Nowadays, many modifications, applied to images and other types of data

Transformers: Original Model

- Encoder-Decoder architecture



Kikaben.com

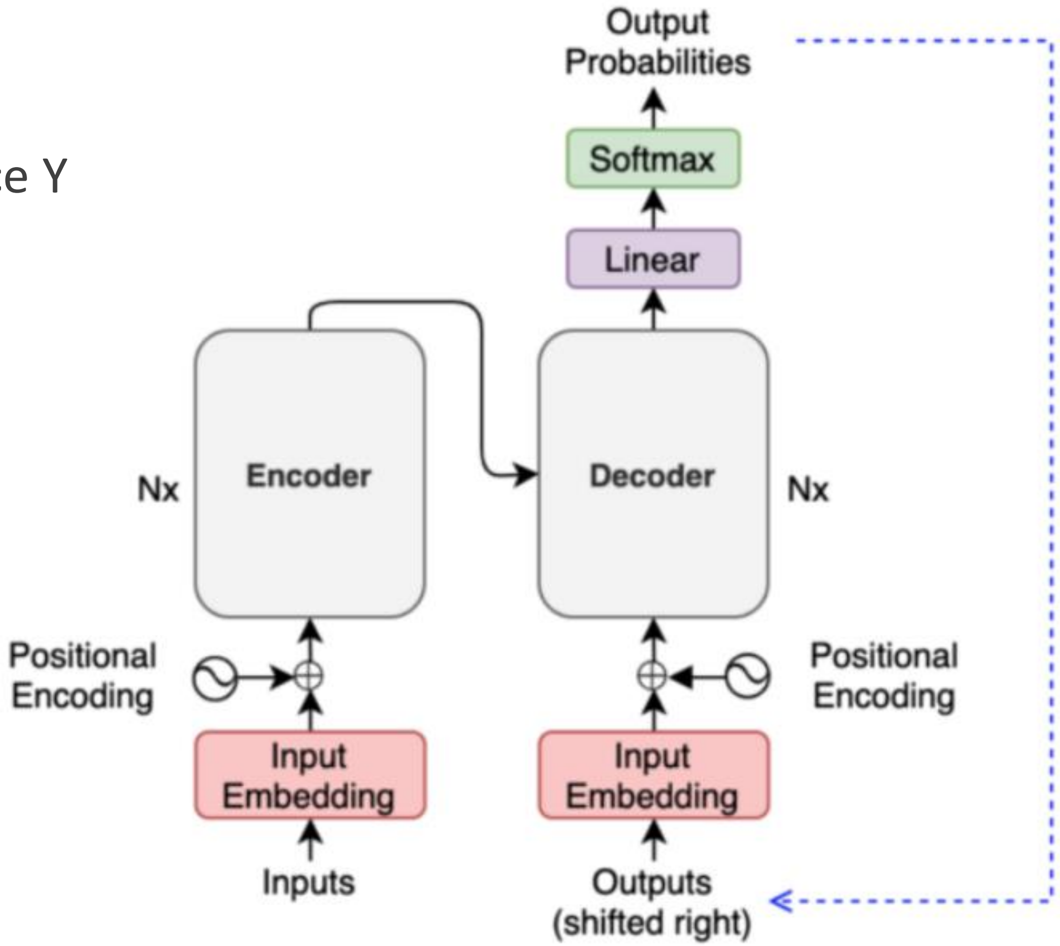
Transformers: Original Model

- Encoder-Decoder architecture

- Assume translation from sentence X to sentence Y

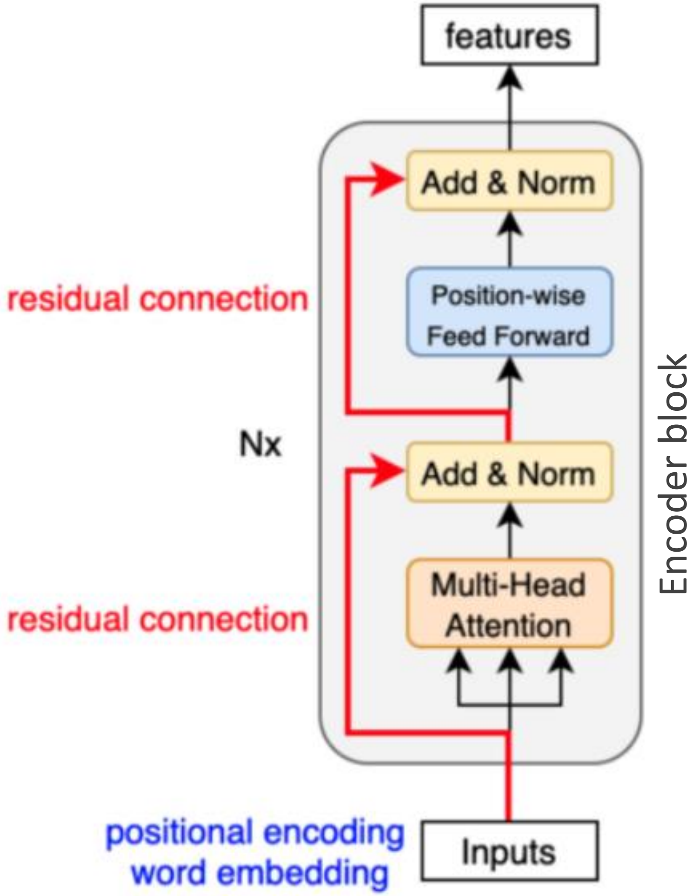
1. Embedding
   - Pre-trained, dictionary
   - Learnt on the go (dense layers or other)
2. Positional encoding
   - Fixed or learnt
3. Encoder – features
   - Embeddings enriched with a context
4. Decoder – probabilities over dictionary
   - Choose the best next word

Repeat until <EOS> predicted.



Kikaben.com

Transformers: Original Model



Kikaben.com

Encoder block

- Input: embedded word or output from previous block
  + positional encoding
- Output: feature vector

Encoder block internals

- Multi-head attention
- Residual connections (addition and layer norm)
- Position-wise Feed Forward
  - Linear layer ⟶ ReLU ⟶ linear layer

Transformer: Attention mechanism

- Translation from sentence *X* to sentence *Y*
- After embedding and encoding:

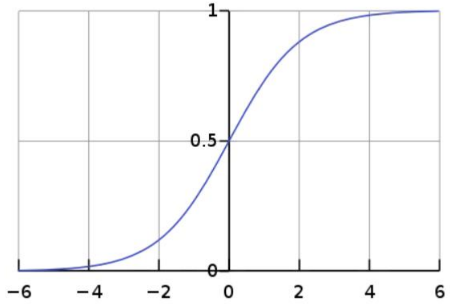
$$\mathbb{X} = (\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n), \quad \mathbb{Y} = (\vec{y}_1, \vec{y}_2, \dots, \vec{y}_m)$$
$$\forall i, j : \ \vec{x}_i, \vec{y}_j \in \mathbb{R}^{d_\mathrm{m}}$$

$$\vec{k}_1 = \mathbb{W}^K \vec{x}_1 \ \dots \ \text{key}$$
$$\vec{q}_1 = \mathbb{W}^Q \vec{y}_1 \ \dots \ \text{query}$$
$$\mathbb{W}^K, \mathbb{W}^Q \ \dots \ d_\mathrm{k} \times d_\mathrm{m}$$

$$s_{11} = \vec{k}_1^T \vec{q}_1 \ \dots \ \text{score}$$

Transformer: Attention mechanism

$$\mathrm{softmax}\left(\vec{z}_i\right) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$



- Translation from sentence X to sentence Y
- After embedding and encoding:

$$\mathbb{X} = (\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n), \quad \mathbb{Y} = (\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m)$$
$$\forall i, j: \ \vec{x}_i, \vec{y}_j \in \mathbb{R}^{d_\mathrm{m}}$$

$$\vec{k}_1 = \mathbb{W}^K \vec{x}_1 \ \ldots \ \text{key}$$
$$\vec{q}_1 = \mathbb{W}^Q \vec{y}_1 \ \ldots \ \text{query}$$
$$\mathbb{W}^K, \mathbb{W}^Q \ \ldots \ d_\mathrm{k} \times d_\mathrm{m}$$

$$s_{11} = \vec{k}_1^T \vec{q}_1 \ \ldots \ \text{score}$$

- In a matrix form:

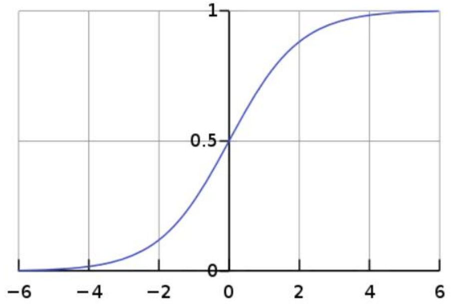
$$\mathbb{K} = \mathbb{W}^K \mathbb{X}$$
$$\mathbb{Q} = \mathbb{W}^Q \mathbb{Y}$$

$$\mathbb{K}^T \vec{q}_1 = \vec{s}_1 \in \mathbb{R}^n$$

$$\mathrm{softmax}\left(\mathbb{K} \vec{q}_1\right) = \vec{w}_1 \ \ldots \ \text{attention weights}$$

Transformer: Attention mechanism

$$\text{softmax}\left(\vec{z}_i\right) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$



- Translation from sentence *X* to sentence *Y*
- After embedding and encoding:

$$\mathbb{X} = (\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n), \quad \mathbb{Y} = (\vec{y}_1, \vec{y}_2, \dots, \vec{y}_m)$$
$$\forall i, j : \ \vec{x}_i, \vec{y}_j \in \mathbb{R}^{d_\mathrm{m}}$$

$$\vec{k}_1 = \mathbb{W}^K \vec{x}_1 \ \dots \ \text{key}$$
$$\vec{q}_1 = \mathbb{W}^Q \vec{y}_1 \ \dots \ \text{query}$$
$$\mathbb{W}^K, \mathbb{W}^Q \ \dots \ d_\mathrm{k} \times d_\mathrm{m}$$

$$s_{11} = \vec{k}_1^T \vec{q}_1 \ \dots \ \text{score}$$

- In a matrix form:

$$\mathbb{K} = \mathbb{W}^K \mathbb{X}$$
$$\mathbb{Q} = \mathbb{W}^Q \mathbb{Y}$$

$$\mathbb{K}^T \vec{q}_1 = \vec{s}_1 \in \mathbb{R}^n$$

$$\text{softmax}\left(\mathbb{K}\vec{q}_1\right) = \vec{w}_1 \ \dots \ \text{attention weights}$$

$$\mathbb{V} = \mathbb{W}^V \mathbb{X} \ \dots \ \text{value matrix}$$
$$\mathbb{V} \ \dots \ d_\mathrm{v} \times n$$

$$\mathbb{V}\vec{w}_1 = \sum_{i=1}^{n} w_{1i} \vec{v}_i \ \dots \ \text{attention vector}$$

context of X regarding query $\vec{q}_1$ (of $\vec{y}_1$)

Transformer: Attention mechanism

- In matrix form – context of X for all words in Y:

$$\mathbb{K}^{\mathsf{T}}\mathbb{Q} = \mathbb{S} = (\vec{s}_1, \ldots, \vec{s}_m) \quad \ldots \quad \text{score matrix}$$

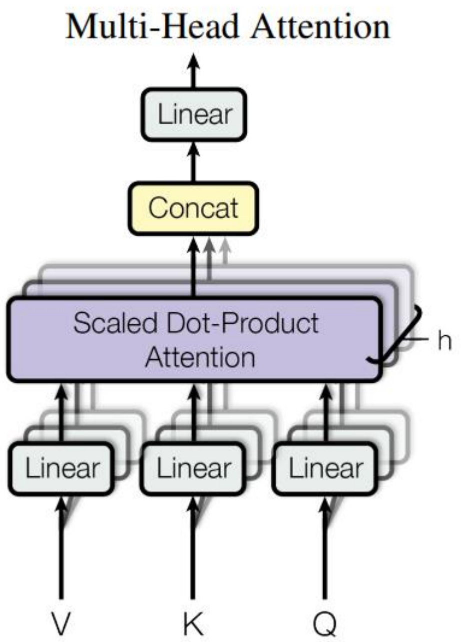
$$\text{Attention}\,(\mathbb{Q}, \mathbb{K}, \mathbb{V}) = \mathbb{V} \cdot \text{softmax}\left(\frac{\mathbb{K}^{\mathsf{T}}\mathbb{Q}}{\sqrt{d_k}}\right) \quad \ldots \quad d_{\mathrm{v}} \times m$$

Transformer: Attention mechanism

- In matrix form – context of *X* for all words in *Y*:

$$\mathbb{K}^{\mathsf{T}}\mathbb{Q} = \mathbb{S} = (\vec{s}_1, \dots, \vec{s}_m) \quad \dots \text{ score matrix}$$

$$\text{Attention}\,(\mathbb{Q}, \mathbb{K}, \mathbb{V}) = \mathbb{V} \cdot \text{softmax}\left(\frac{\mathbb{K}^{\mathsf{T}}\mathbb{Q}}{\sqrt{d_k}}\right) \quad \dots \quad d_{\mathrm{v}} \times m$$



Attention Is All You Need

- Multi-head attention:

$$\text{head}_i = \text{Attention}\,(\mathbb{Q}_i, \mathbb{K}_i, \mathbb{V}_i)$$

$$\mathbb{Q}_i = \mathbb{W}_i^{\mathrm{Q}}\mathrm{Y}$$
$$\mathbb{K}_i = \mathbb{W}_i^{\mathrm{K}}\mathrm{X}$$
$$\mathbb{V}_i = \mathbb{W}_i^{\mathrm{V}}\mathrm{X}$$

$$\forall i \in \{1, \dots, h\}$$

$$\text{MultiHead}\,(Y, X) = \mathbb{W}^{\mathrm{O}} \cdot \text{Concat}\,(\text{head}_1, \dots, \text{head}_h)$$

$$\text{Output dim: } d_{\mathrm{m}} \times m$$

$\longrightarrow$ i.e. we mix the contextual information gained from the individual heads

Transformer: Attention mechanism

- In matrix form – context of *X* for all words in *Y*:

$$\mathbb{K}^{\mathsf{T}}\mathbb{Q} = \mathbb{S} = (\vec{s}_1, \dots, \vec{s}_m) \quad \dots \quad \text{score matrix}$$

$$\text{Attention}\,(\mathbb{Q}, \mathbb{K}, \mathbb{V}) = \mathbb{V} \cdot \text{softmax}\left(\frac{\mathbb{K}^{\mathsf{T}}\mathbb{Q}}{\sqrt{d_k}}\right) \quad \dots \quad d_{\mathrm{v}} \times m$$

Multi-head **self**-attention:

Y = *X*

↓

MultiHead*(X, X)*

- Multi-head attention:

$$\text{head}_i = \text{Attention}\,(\mathbb{Q}_i, \mathbb{K}_i, \mathbb{V}_i)$$

$$\mathbb{Q}_i = \mathbb{W}_i^{\mathrm{Q}}\mathrm{Y}$$
$$\mathbb{K}_i = \mathbb{W}_i^{\mathrm{K}}\mathrm{X}$$
$$\mathbb{V}_i = \mathbb{W}_i^{\mathrm{V}}\mathrm{X}$$

$$\forall i \in \{1, \dots, h\}$$

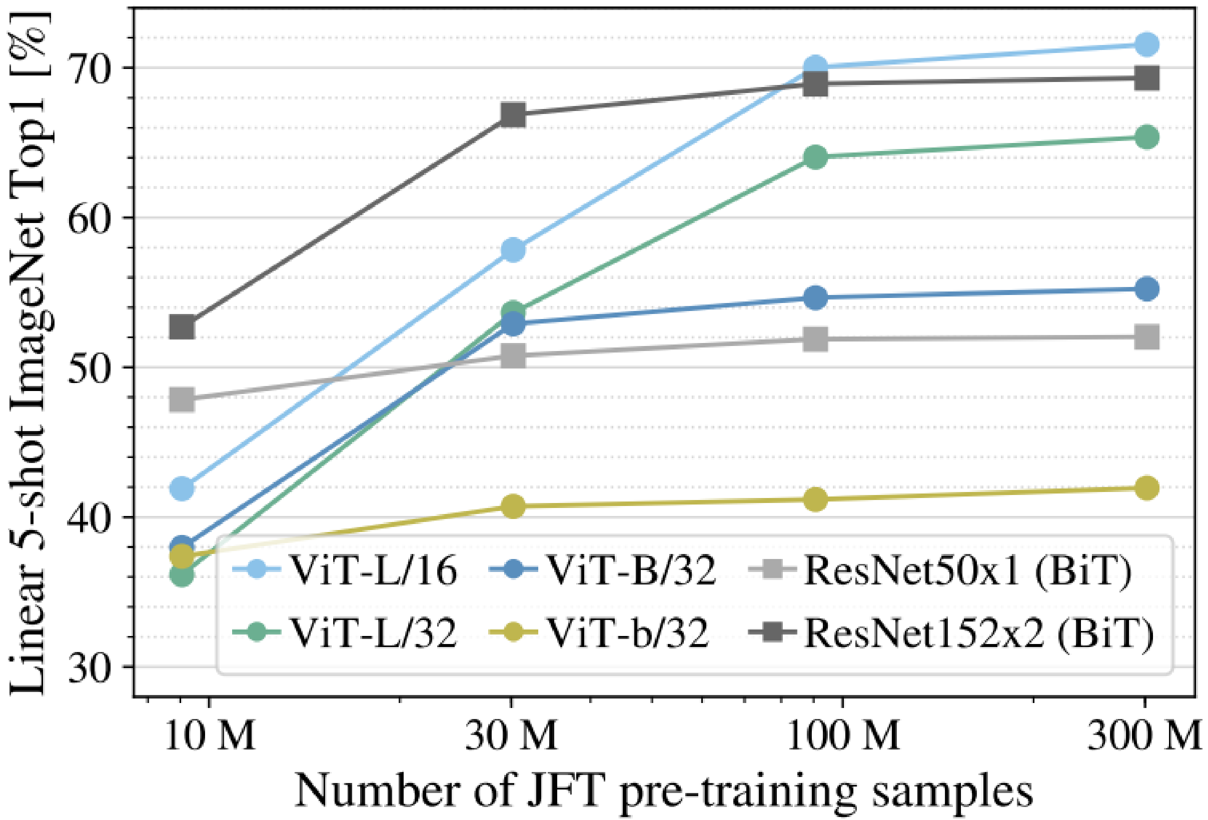
$$\text{MultiHead}\,(Y, X) = \mathbb{W}^{\mathrm{O}} \cdot \text{Concat}\,(\text{head}_1, \dots, \text{head}_h)$$

$$\text{Output dim:}\ \ d_{\mathrm{m}} \times m$$

$\longrightarrow$ i.e. we mix the contextual information gained from the individual heads

Transformer: Performance

- Vision Transformer (ViT) model
- Only encoder part trained for classification task

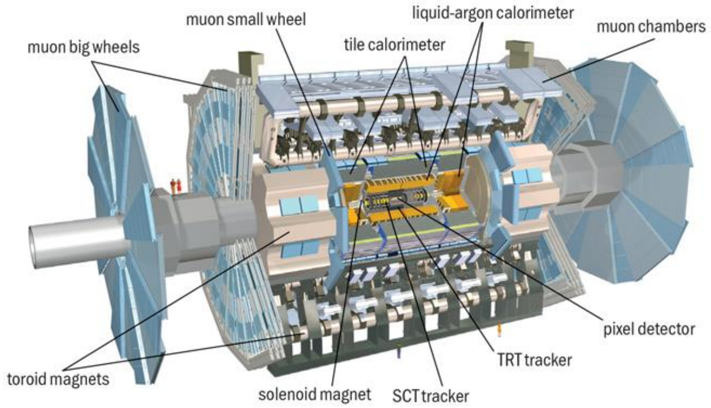


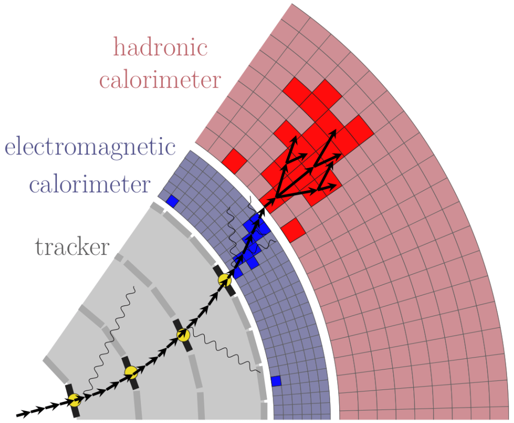
"An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale" by A. Dosovistkiy et al. (2020)

Calorimeter simulations

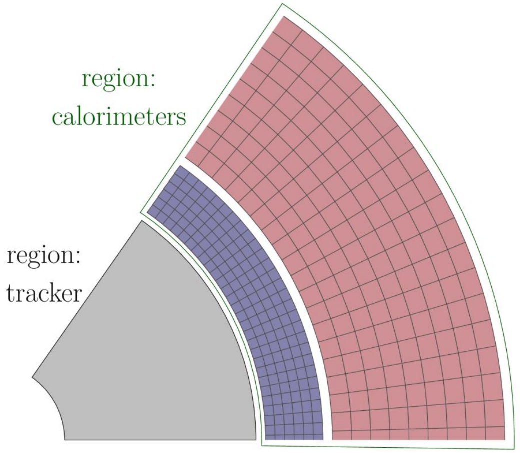
- Simulating passage of a particle through a detector
- Calorimeter – high granularity, large volume

- Monte Carlo simulations – computationally intensive
- Fast simulation – based on DL models
  - GANs, VAEs, INNs (Normalizing Flows)
  - MLP, CNN, GNNs

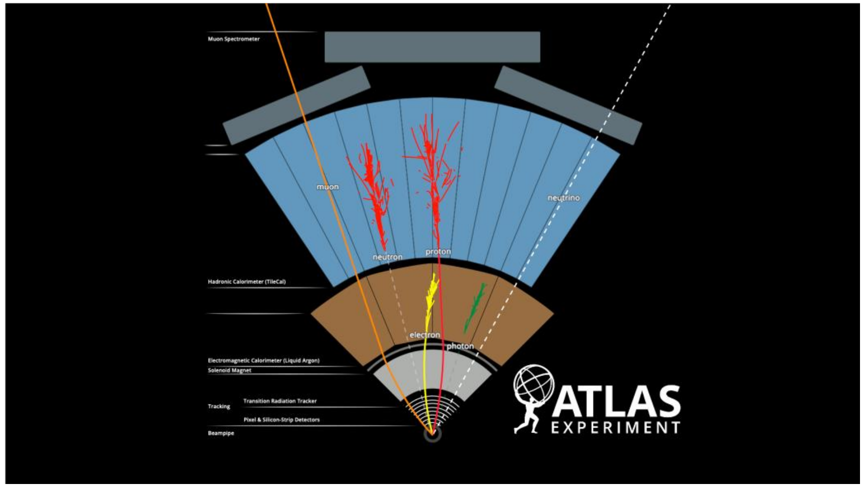




Full Sim

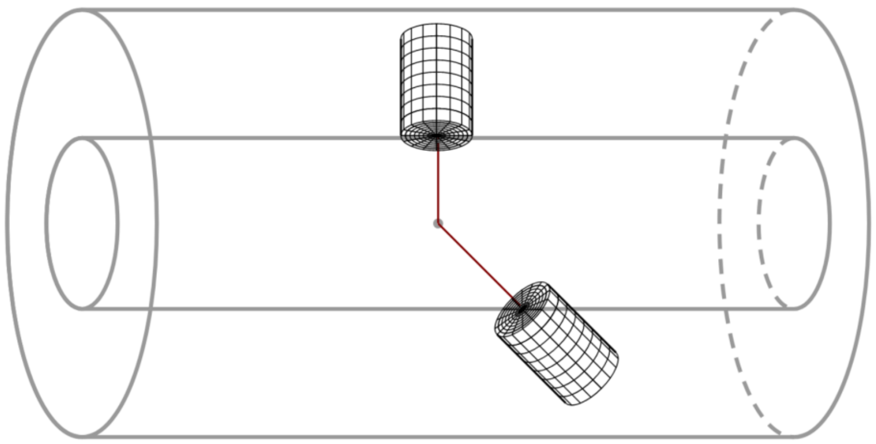


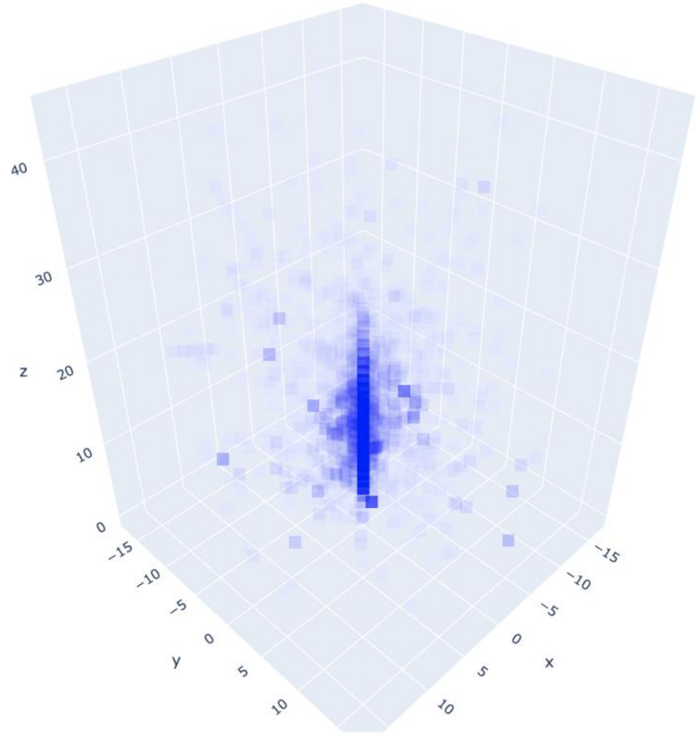
Fast Sim



General calorimeter dataset

- RxPhixZ = 10x50x24
- Primary particle energy: 64, 128, 256 GeV
- Entering angle: 70°
- 10 000 samples per energy
- Sparsity: 60 % (mostly empty image)
- Public dataset (Zenodo.org)



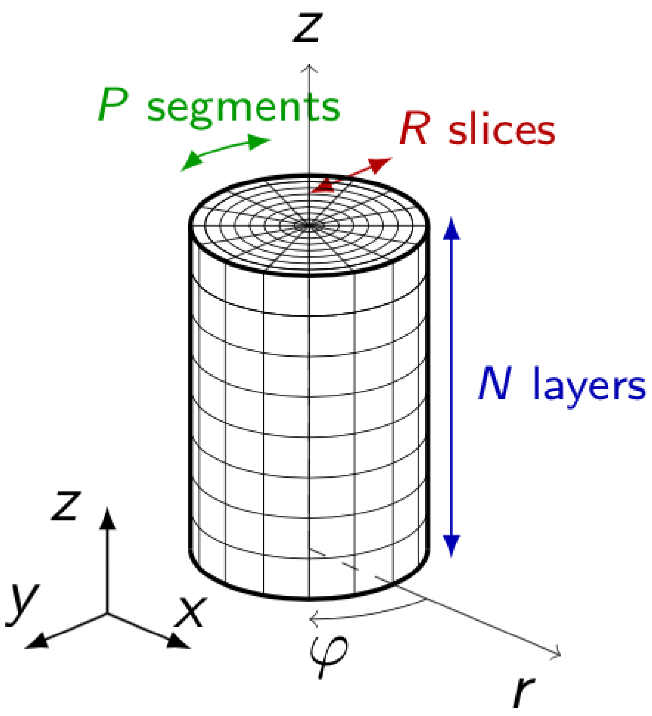


Pre-processing and training for image completion

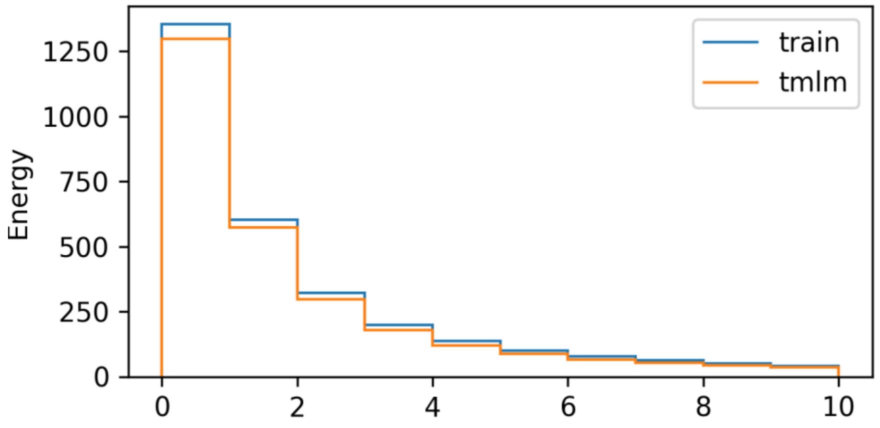
Pre-processing (creating a sequence + embedding)

- Splitting the cylinder into patches
  - Slicing in Phi and Z directions

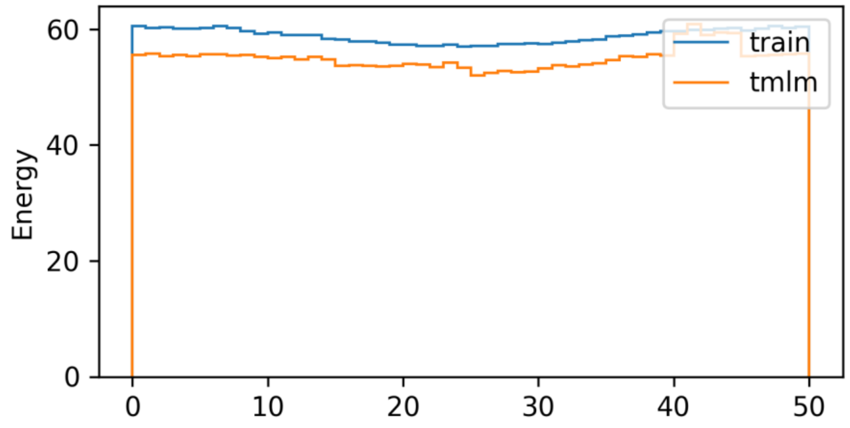
  → **sequence of 20 patches** (600 cells each)
- Flattening of the patch
- Embedding each flattened patch using GCN (Graph Convolutional Network)
- Adding positional encoding to the embedding
  - 2-layer MLP network
- Masking 70 % of the patches (replacing all values with 0s)
- Training transformer-based model to complete the masked patches
  - Encoder part of the transformer
- Reshape to original dimensions



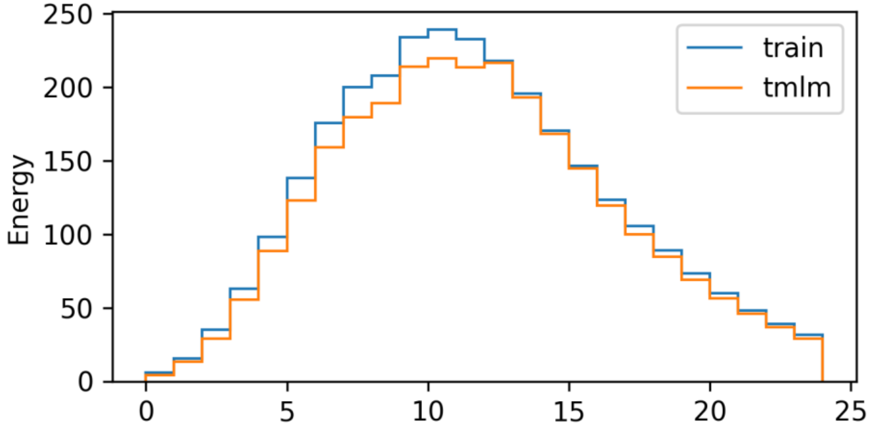
Transformer for image completion



R



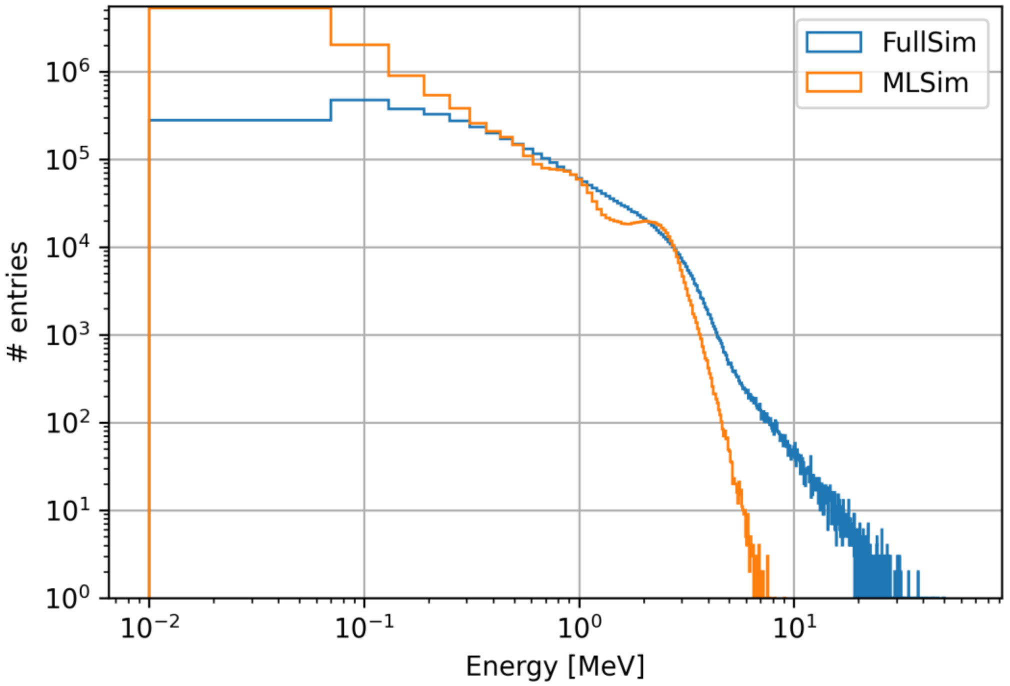
Phi

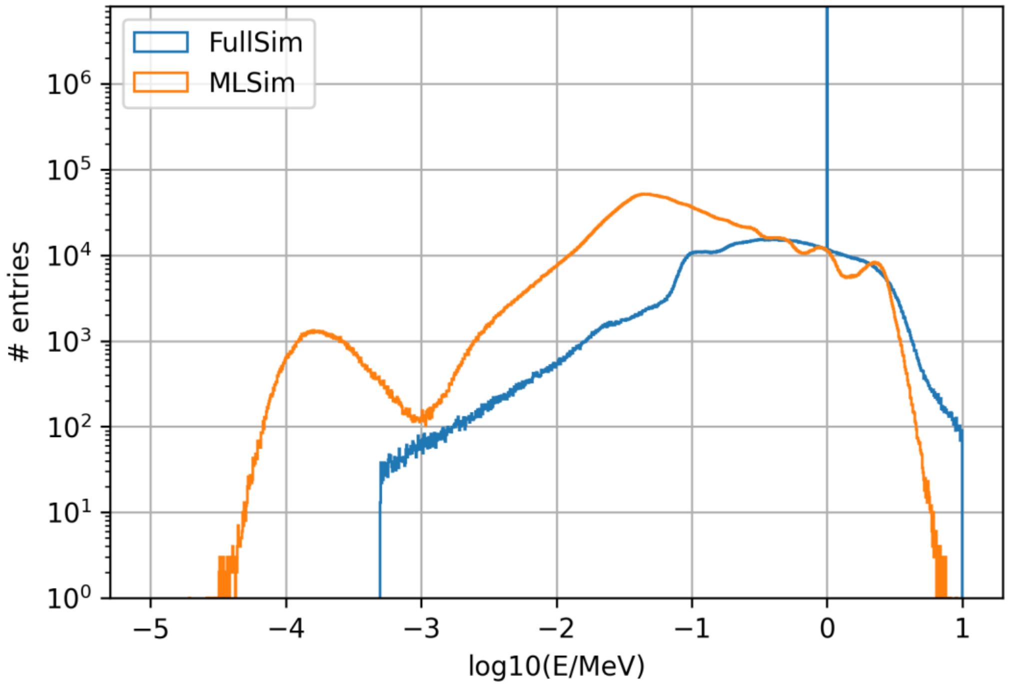


Z

Transformer for image completion

- Cell energy distribution





Thank you for your attention.

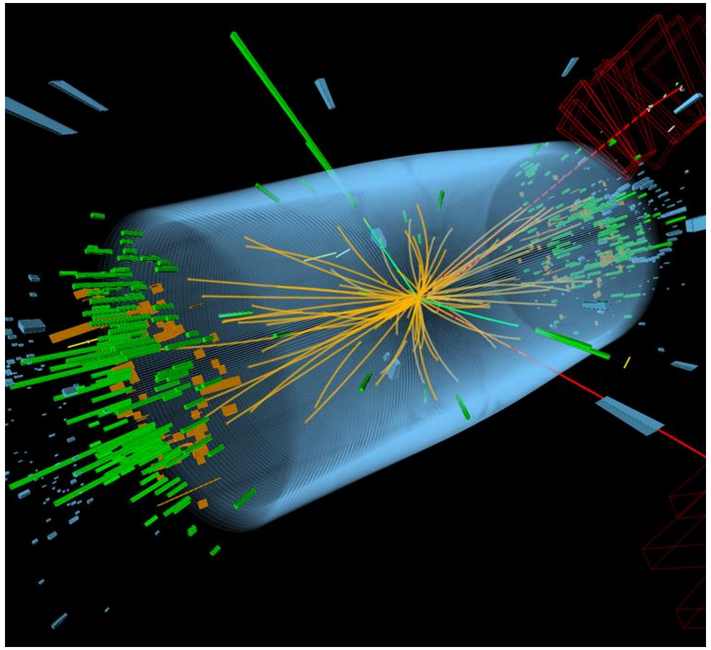
References:

- A. Vaswani et al. Attention Is All You Need (2017)
- A. Dosovistkiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (2020)
- Kikaben.com. Transformer Encoder-Decoder: Let's Understand Then Model Architecture (2021) [online]
- D. Salamani and A. Zaborowska. High Granularity Electromagnetic Calorimeter Shower Images (2022) [online]

Detector simulations

What?

- Simulating passage of a particle through a detector
- Modelling particle interactions with matter
  - Specialized software – Geant4
  - Mote Carlo method used to solve the particle transport equations

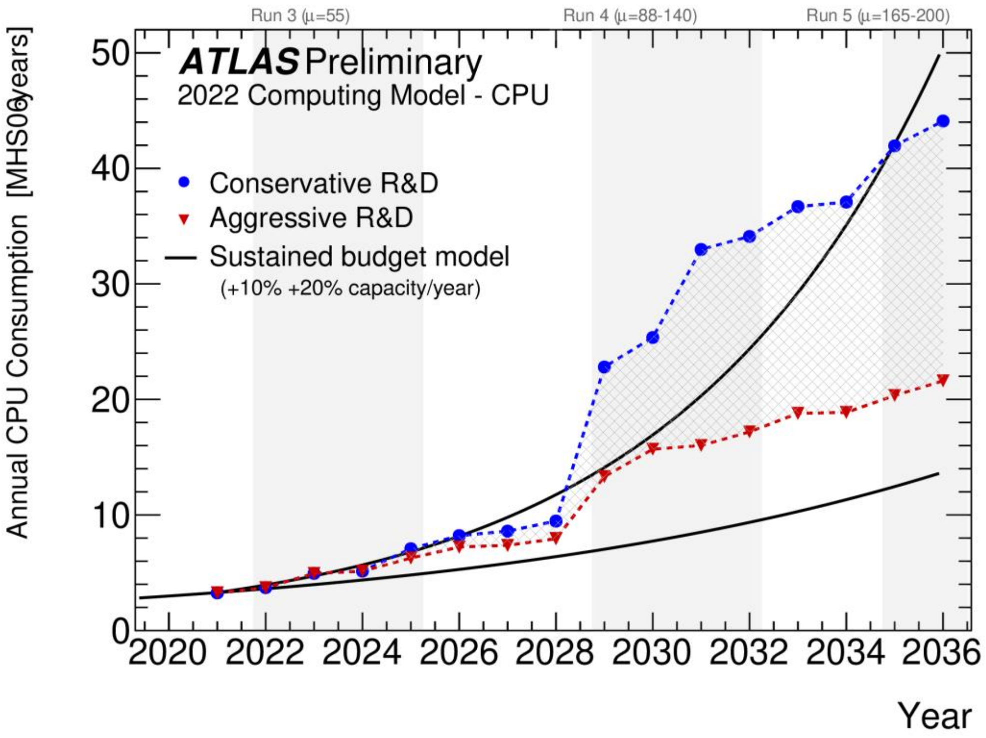


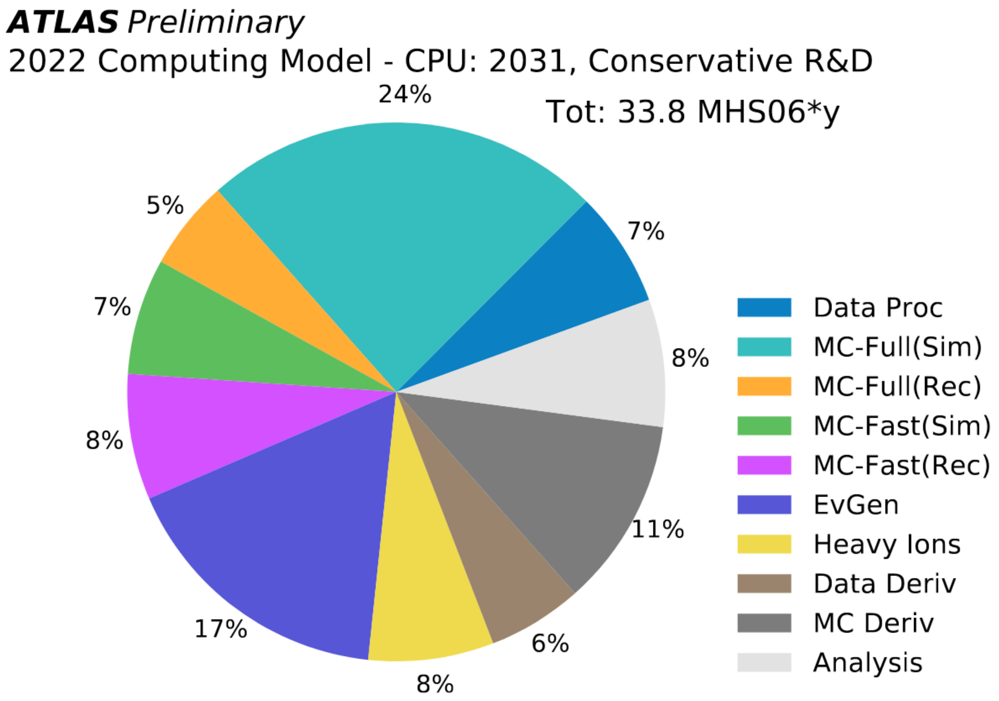
Why?

- Detector design
- Physics analysis purposes
  - Reference to be compared with the experimental measurements
  - Can simulate known and "new" physics
- Tuning of reconstruction algorithms

Detector simulations

- Monte Carlo simulations are slow!
  - ATLAS experiment – simulations take substantial part of resources
  - 1 full event ~ 1 minute (reference)

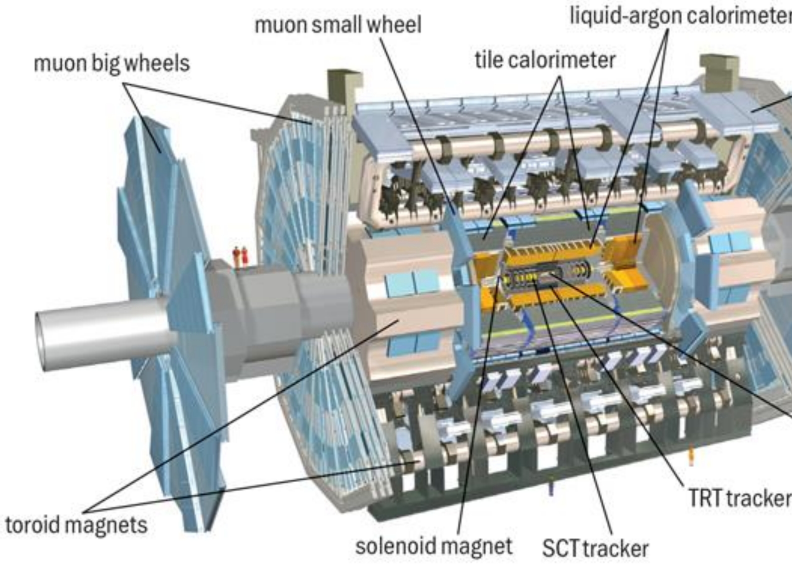


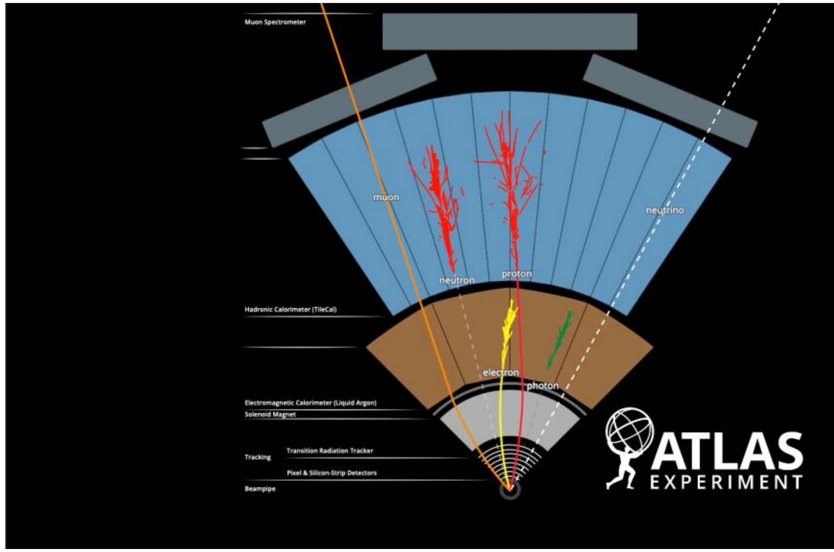


- Conservative R&D – maintain person power, already includes ML and DL development
- Aggressive R&D – more people on computing development

Detector simulations

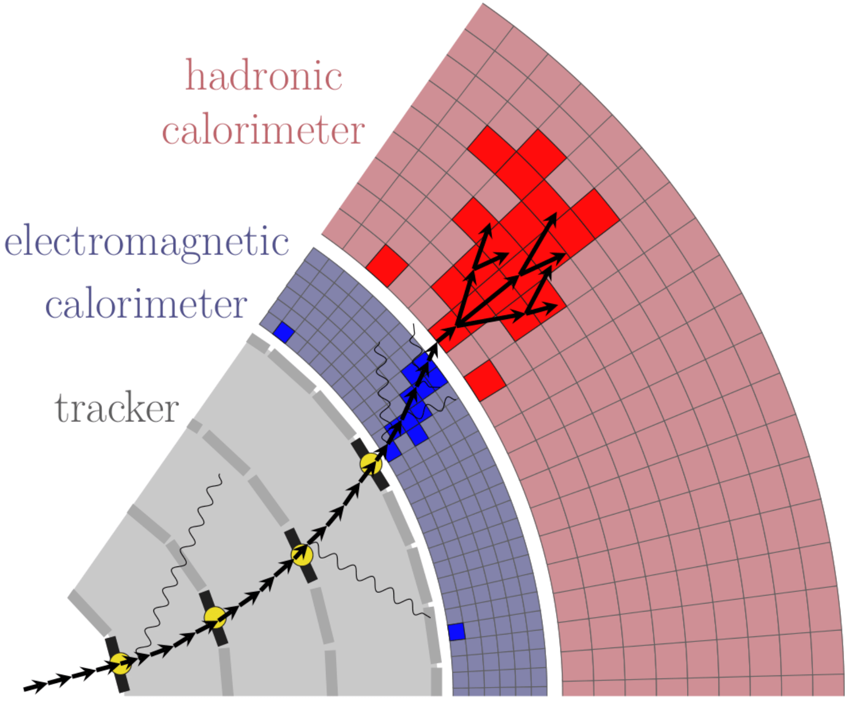
- Calorimeters are the most time consuming part – 53 % (1)
  - Large volume, high granularity
  - Electromagnetic and hadronic calorimeters
- Search for fast simulation methods ongoing in all experiments
- Deep learning – up to 10^6 x speed up (2)
- Current approaches
  - GANs (ATLAS) – already in use – 100 GANs ? (reference)
  - VAEs (Geant4)
  - Normalizing Flows (DESY?) – use links form the calochallenge workshop





Detector simulation

- Full Sim



- Fast Sim