
CERN
openlab

Foundation Models

Renato Cardoso(openlab), Nadya Chernyavskaya(EP-CMG-DS), Kristina Jaruskova(openlab), Piyush Raikwar(EP-SFT), Dalila Salamani(EP-SFT), Kalliopi Tsolaki(openlab), Sofia Vallecorsa(openlab), Anna Zaborowska(EP-SFT)

Special thanks to Mudhakhar Srivatsa from IBM

Foundation Models

- A model trained on broad data and adaptable to a range of different downstream tasks, zero-shot, few-shot learning.
- Foundation Models concepts:
  - self/semi-supervised learning + transfer learning but at scale:
    - Billions of parameters and gigabytes of data
    - Large and diverse datasets → powerful representations
- Examples:
  - BERT (340M params.), GPT-2, GPT-3 (175B params.) – Generative language models
  - CLIP – Language-Image pre-training
  - DALL-E, DALL-E 2, Imagen – Text to Image models
  - GATO – Sequence to sequence model

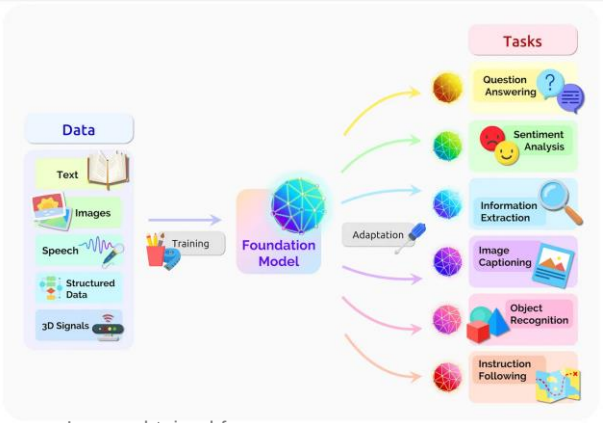


Image obtained from:
On the Opportunities and Risks of Foundation Models

- Stanford CRFM (2021) : On the Opportunities and Risks of Foundation Models [arxiv.2108.07258]

Introduction

Why use Foundation Models:

- ML is computational expensive with large datasets and models
  - Train once. Then, adapt to new detector geometries, quickly.
- Transformers as building block in foundation models:
  - A generalized architecture without any inductive bias
  - Model long-range dependencies (Attention mechanism)
  - Permutation invariant
  - Initially proposed for sequence-to-sequence tasks

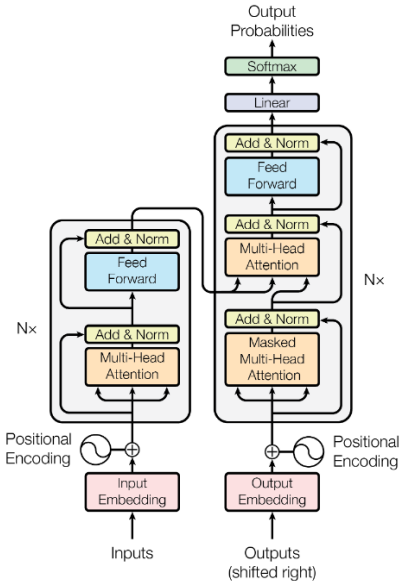


Figure 1: The Transformer - model architecture.

Our Objective:

- Foundation model trained on MC data to perform different physics related tasks

  Simulations - one lengthy training, then fast adaptation to different detector geometries

  Reconstruction - one base model adaptable to different tasks (particle identification, regression on phys. variables, etc.
- Understand how foundation model concept apply to our use case:
  - Understand the minimal scale of the model for reaching meaningful results (No need to reach BERT / GPT-3 scale)

First Phase: Checking learned shower representation with transformers

Second Phase: Generative Foundation model for fast and accurate calorimetry simulation

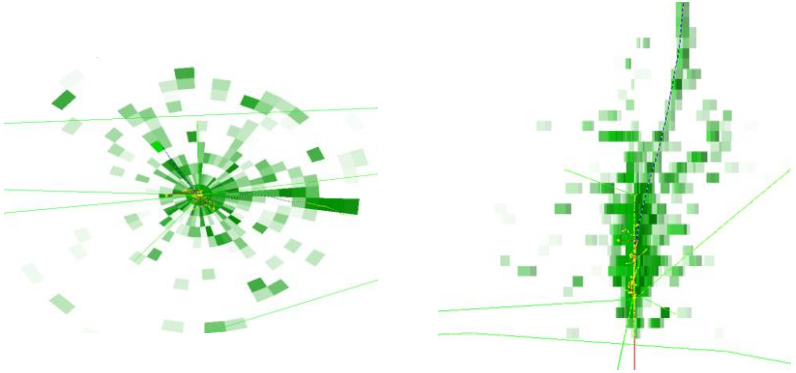
Third Phase: Large scale training

Generative Foundation Model for FastSim

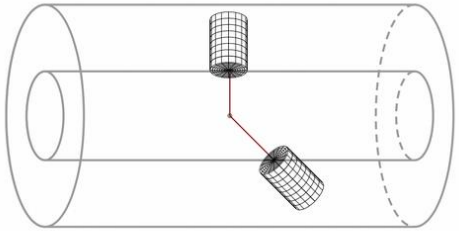
Dataset:

- High Granularity Electromagnetic Calorimeter Shower Images [zenodo]

	Original	Subset
Energy	1 GeV - 1 TeV	64, 128 & 256 GeV
Angle	50º - 90º	70º
Detector Materials	SiW, SciPb	SiW
Granularity	40k	12k / 40k



Electromagnetic Calorimeter Shower Image



Cylindrical read-out along particle direction

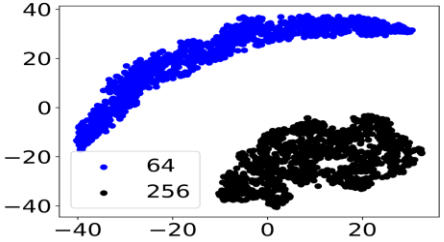
First Phase

Using a ViT model to learn the shower representation

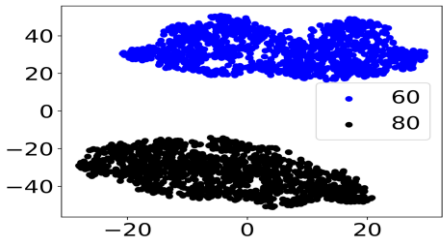
- Using a masked Language model (MLM)
- MLM is learning representation by trying to predict hidden information
- An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [arXiv:2010.11929]

Classification:

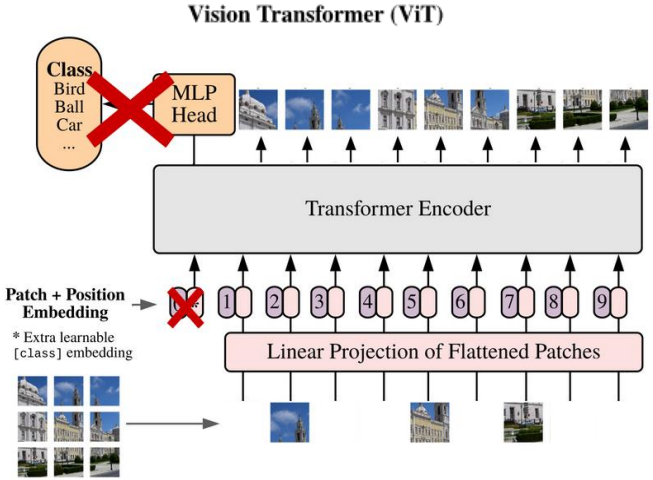
- Downstream classifier (Downstream task)
- Try to predict the energy of the incident particle from transformer embeddings.



Energy prediction



Angle prediction



More work done…

A lot of other tasks:

- VAE like-learning with transformers
- Graph Neural Network
- Preprocessing
- Sinkhorn Loss
- Secondary Loss for regression task

Current focus on Generative Task (Second Task):

- **Diffusion**
  - Denoising Diffusion Probabilistic Models (DDPM)
  - Used for most of the foundation models for image generation
- **Auto-regressive**
  - Vector Quantised-Variational Auto Encoder (VQ-VAE)
  - Commun on NLP tasks while also showing good results on images tasks

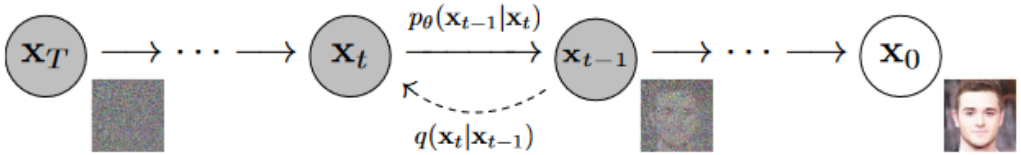
Diffusion Model

What are diffusion Models:

- Idea:
  - Gradually adding Gaussian noise to an image and then use a model to reverse the process
- What is Diffusion:
  - random motion of the particles or molecules, described by the laws of thermodynamics and statistical mechanics
- Diffusion Models are a class of probabilistic generative models that turn noise to a representative data sample.
- Examples:
  - DALL-E 2 (Open-AI, 12 billion parameters)
  - Imagen (Google)
- Denoising Diffusion Probabilistic Models [arXiv:2006.11239]

Why use Diffusion Models:

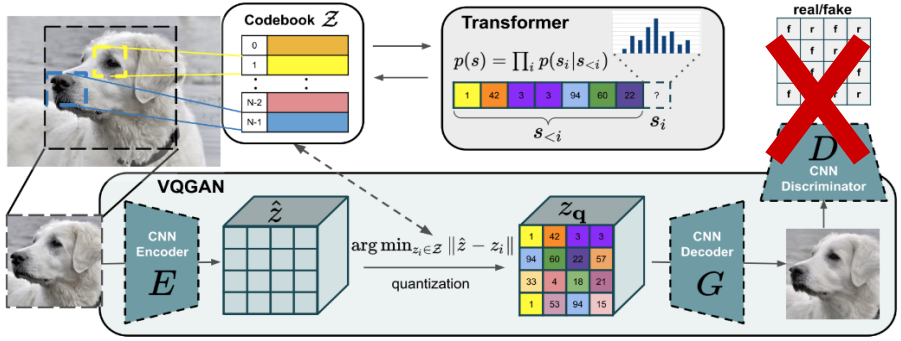
- High sample generation quality.
- Diverse sample generation.
- Able to do multiple tasks:
  - Text to image generation
  - Image Inpainting
  - etc.

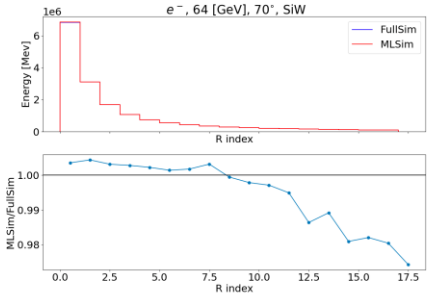


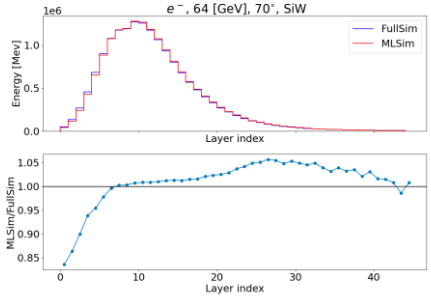
Autoregressive model

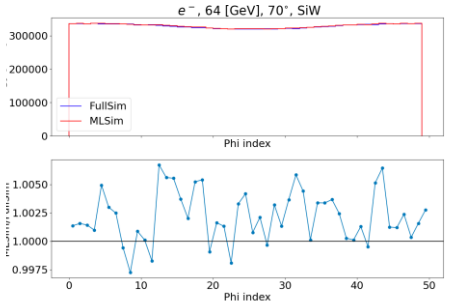
VQ-VAE + Transformers:

- VQ-VAE to build a codebook (dictionary) of shower features.
- Transformer to predict those codebook vectors (shower features) autoregressively, starting from Layer 0.
  - VQVAE sees whole shower. Decodes it into 64* tokens.
  - Transformer sees previous tokens, outputs probabilities over the next one.
- Advantages:
  - 64 forward passes needed.
  - Shorter sequence.
- ~10-20 mins per epoch.









Conclusion and Future Work

Conclusions:

- First phase: Masked Language Model
  - Able to achieve good shower representations
  - Able to do downstream classifying task
- Multiple different test are being realized at the same time
- Currently finished with First Phase focusing on the work for the second phase
  - 2 different generative models applied to our physics use case, still in proof of concept

Future Work:

- Finish the proof of concept
- Third phase: Scale up the model and the dataset for better representations
  - Understand the minimal scale of the model for reaching meaningful results

Paper submission on CHEP 2023: **Transformers for Generalized Fast Shower Simulation**


CERN
openlab

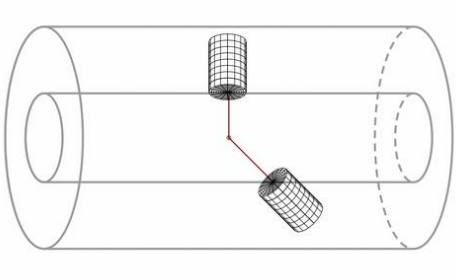
Backup

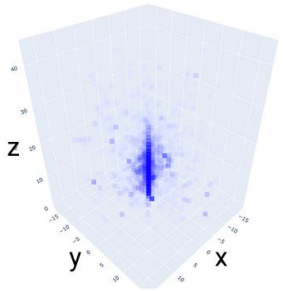
Dataset and Patches

Dataset:

- High Granularity Electromagnetic Calorimeter Shower Images

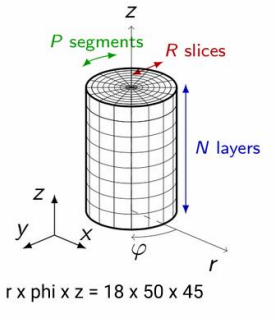
	Original	Subset
Energy	1 GeV - 1 TeV	64, 128 & 256 GeV
Angle	50 - 90	70
Geometries	SiW, SciPb	SiW

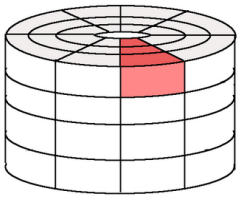




Patch configuration:

- Transformers needs a sequence as input
- Patches are formed by making splits in r, phi and z direction
- More patches -> more computationally expensive
- Current:
  - 1 patch in r, 10 in phi, 15 in z
  - Patch size = 18 x 5 x 3





Eg. 3 x 1 x 1

Positional embeddings and Masking

Positional embeddings:

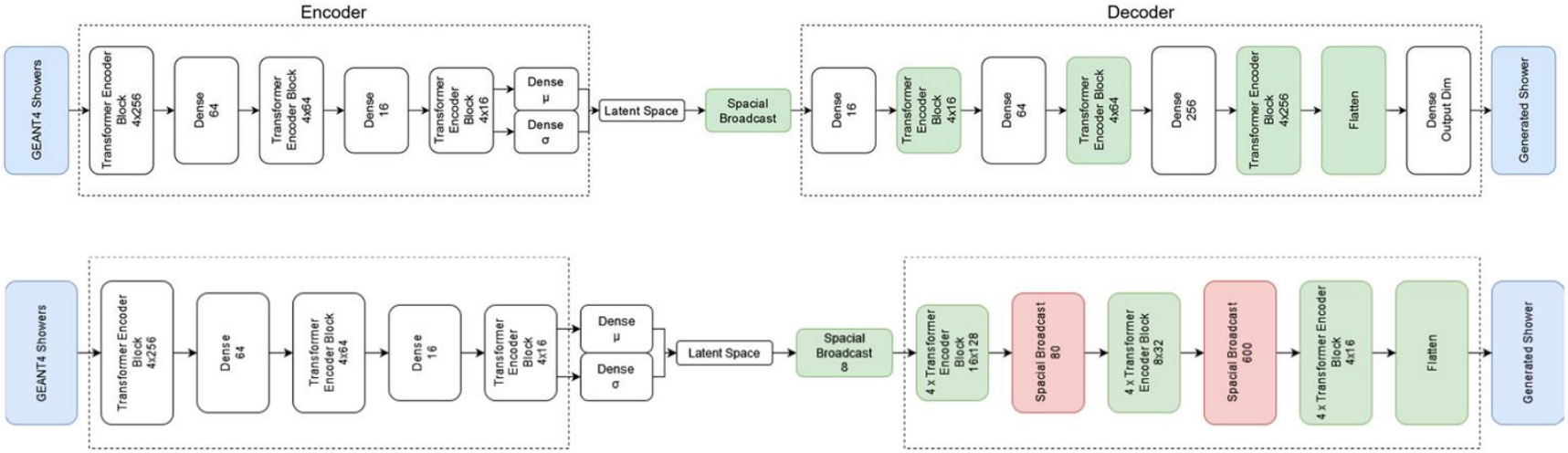
- Transformers are permutation invariant. Positional embeddings gives an understanding of position to the model
- Explored:
  - 1D learnable keras embedding layer.
  - Fixed 3D positional embeddings
    - Alternate sine-cosine
    - Each direction takes 1/3rd of the embedding dimension
    - Phi-rollover
- Observation
  - Fixed 3D positional embeddings perform better

Masking:

- Implementation:
  - Randomly choose given percentage of patches to mask
  - Set all elements in that patch to zero
  - Feed to transformer
  - Try to predict the whole input
- Observations
  - Better results with higher percentage
  - 10% to 90%
  - Bellow 60% , no meaningful information learned

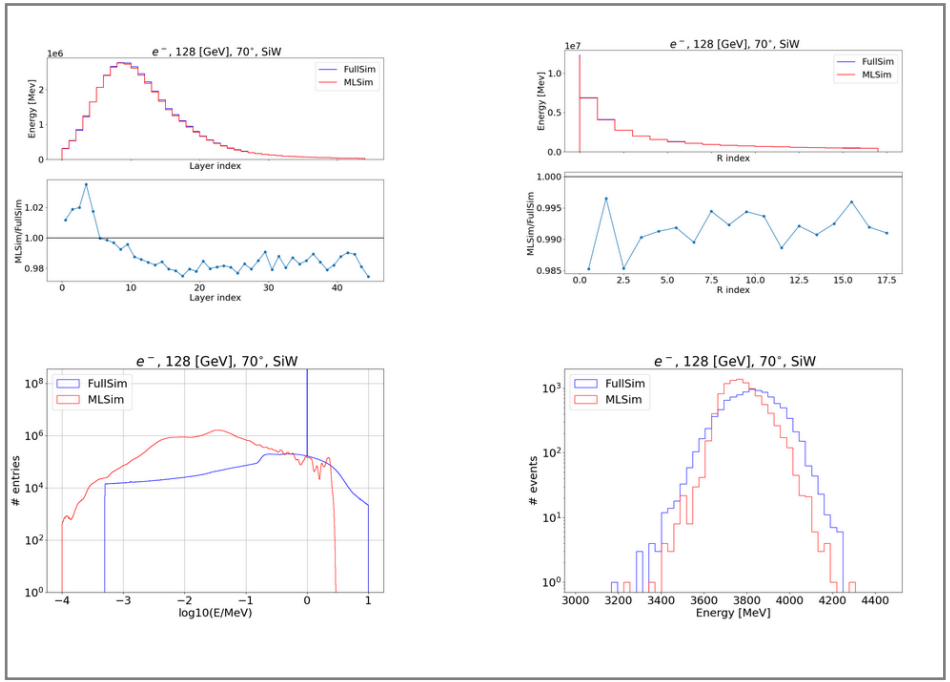
VAE-like learning

- Use of a Dense VAE model architecture with transformer encoder in between
  - Have a model that already works on this specific task and change to include the attention mechanism
  - Substitute the Dense layers, proving that the Attention mechanism is working for this task
    - Use Spatial broadcast instead [arXiv:1901.07017]



VAE-like learning

Results with Dense Layers



Transformers

- Proposed for sequence-to-sequence tasks
- I/O is any type of sequences.
- Encoder-Decoder blocks
- Positional embeddings
- Attention: Dynamically focus on important parts in the input.
- Multi-headed attention.

Attention Is All You Need [arXiv:1706.03762]

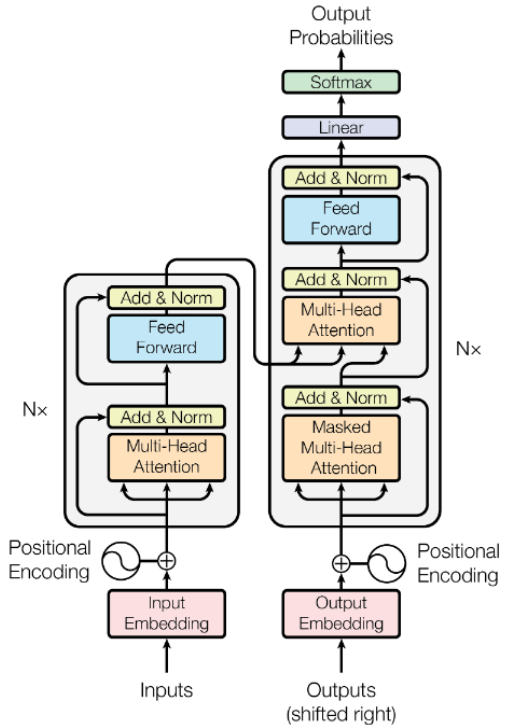


Figure 1: The Transformer - model architecture.

Preprocess and Loss Function

Preprocessing

Motivation:
- Exploiting methods developed for computer vision comes with the challenging aspect of the dynamic ranges of pixel intensities (image:0-255 vs energy depositions: >10 orders of magnitude)
- Improve per cell energy generation distribution
- Previous efforts shown results of trade-off between faster convergence and retaining image quality*

Preprocessing techniques on shower data (experiments carried out on VAE):
- division by energy value of the incident particle in GeV and MeV
- log transformation
- power-law

WIP:
- division by max shower energy and 99th percentile

Loss function

Motivation:
- Improve per cell energy generation distribution
- Get more feedback from loss function

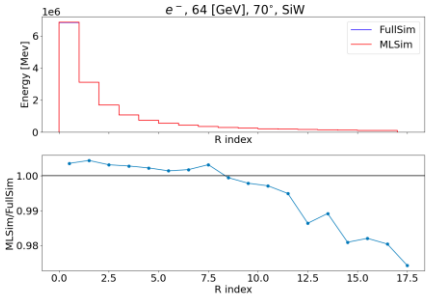
Reconstruction loss function experiments:
- Binary Cross Entropy proved to work with our data
- MSE / MAE (did not work)

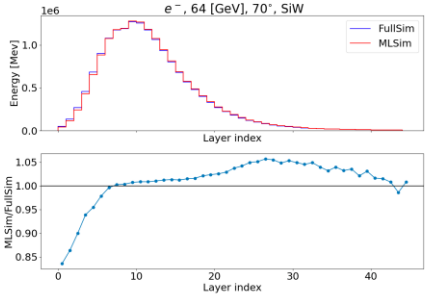
WIP:
- adding components from secondary learning tasks (exps on VAE)
  - regression of the primary particle energy

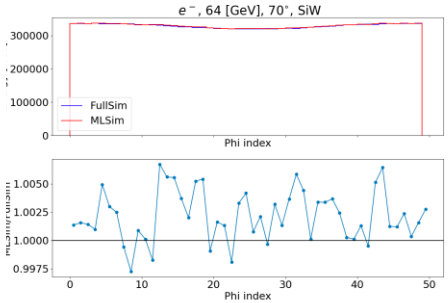
*Khattak, G.R., Vallecorsa, S., Carminati, F. et al. Fast simulation of a high granularity calorimeter by generative adversarial networks. Eur. Phys. J. C 82, 386 (2022). https://doi.org/10.1140/epjc/s10052-022-10258-4

Autoregressive results

VQVAE







AR Prior