Co-financed by the Connecting Europe
Facility of the European Union

INTERACTIONS

UNIVERSITY OF
COPENHAGEN

PolarBERT, a foundation model for the IceCube Neutrino Observatory

Inar Timiryasov, Jean-Loup Tastet, Oleg Ruchayskiy
Niels Bohr Institute and DIKU, University of Copenhagen

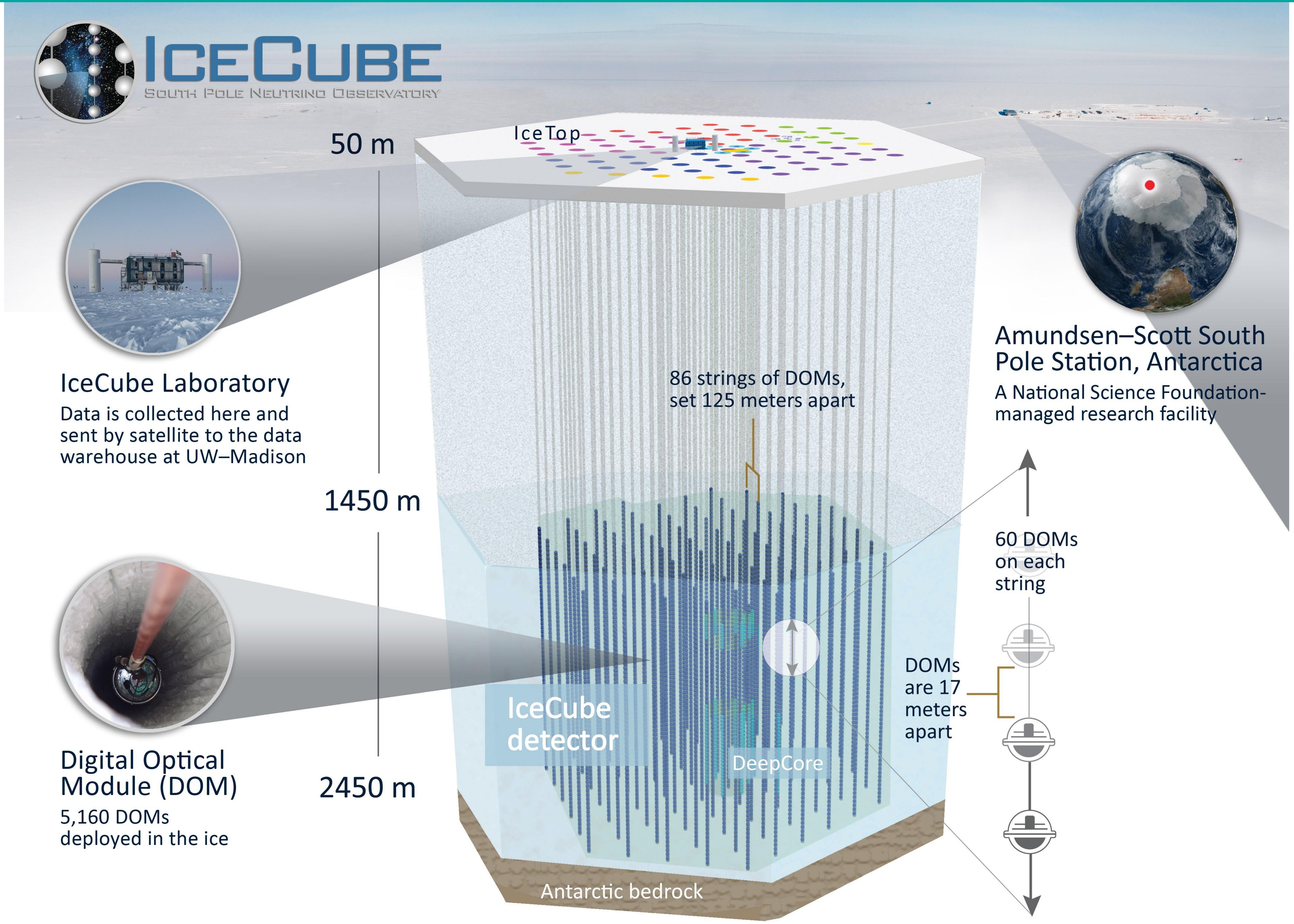
PolarBert paper:
https://ml4physicalsciences.github.io/2024/files/NeurIPS_ML4PS_2024_259.pdf

repo: https://github.com/timinar/PolarBERT/tree/main

HAMLET - Physics
2024-08-20, Copenhagen

IceCube

- Neutrino telescope
- Located at the South Pole
- Detector volume: 1 cubic kilometer
- Oftentimes observes through Earth
- 5160 optical modules (DOMs)
- Public dataset from Kaggle Competition 130 million events

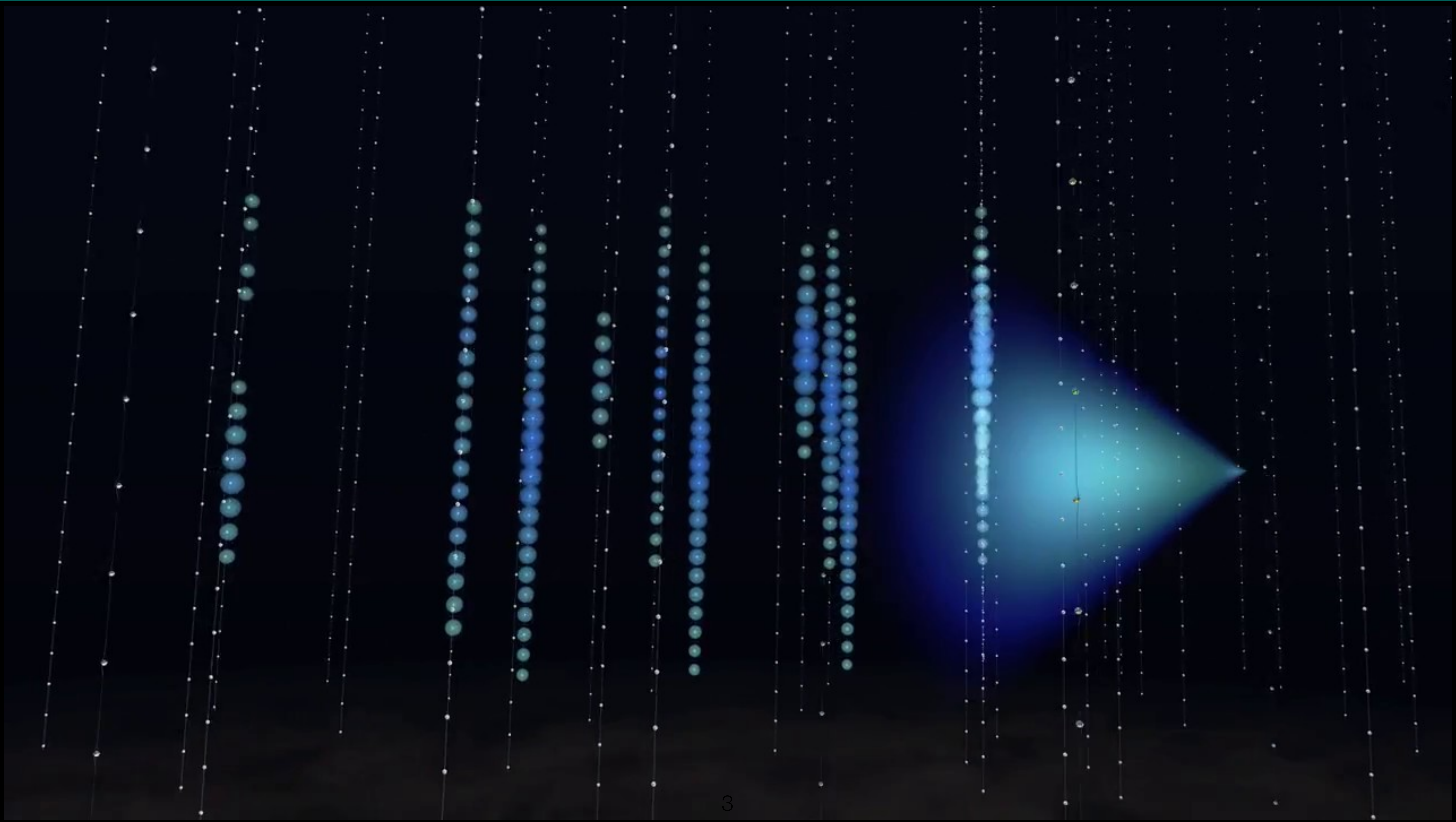


KM3NeT module: 

* I am not a member of IceCube

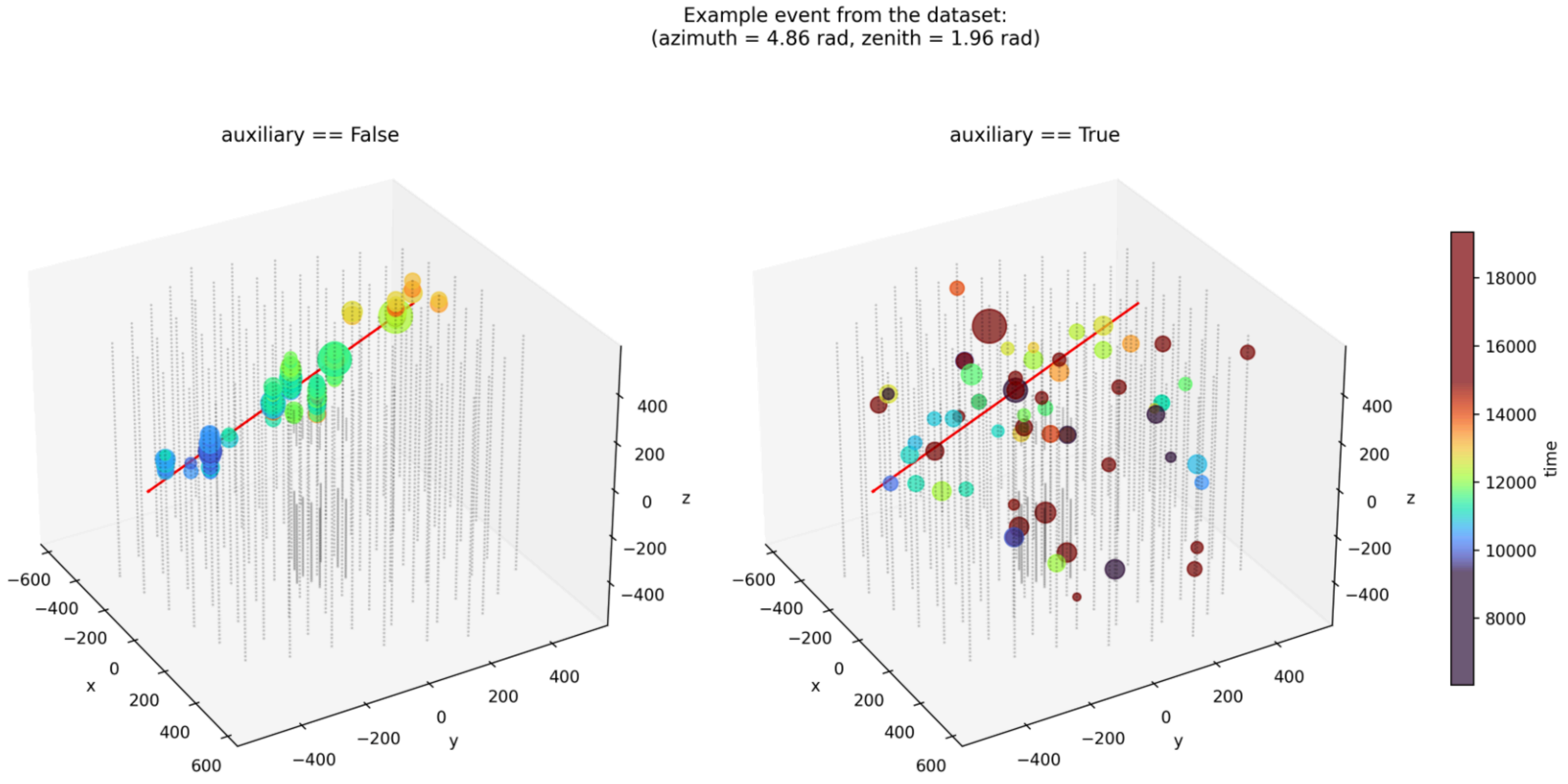
IceCube event

https://youtu.be/OXSqiPLn9CM?si=nnvKH0WpJgEWRn56



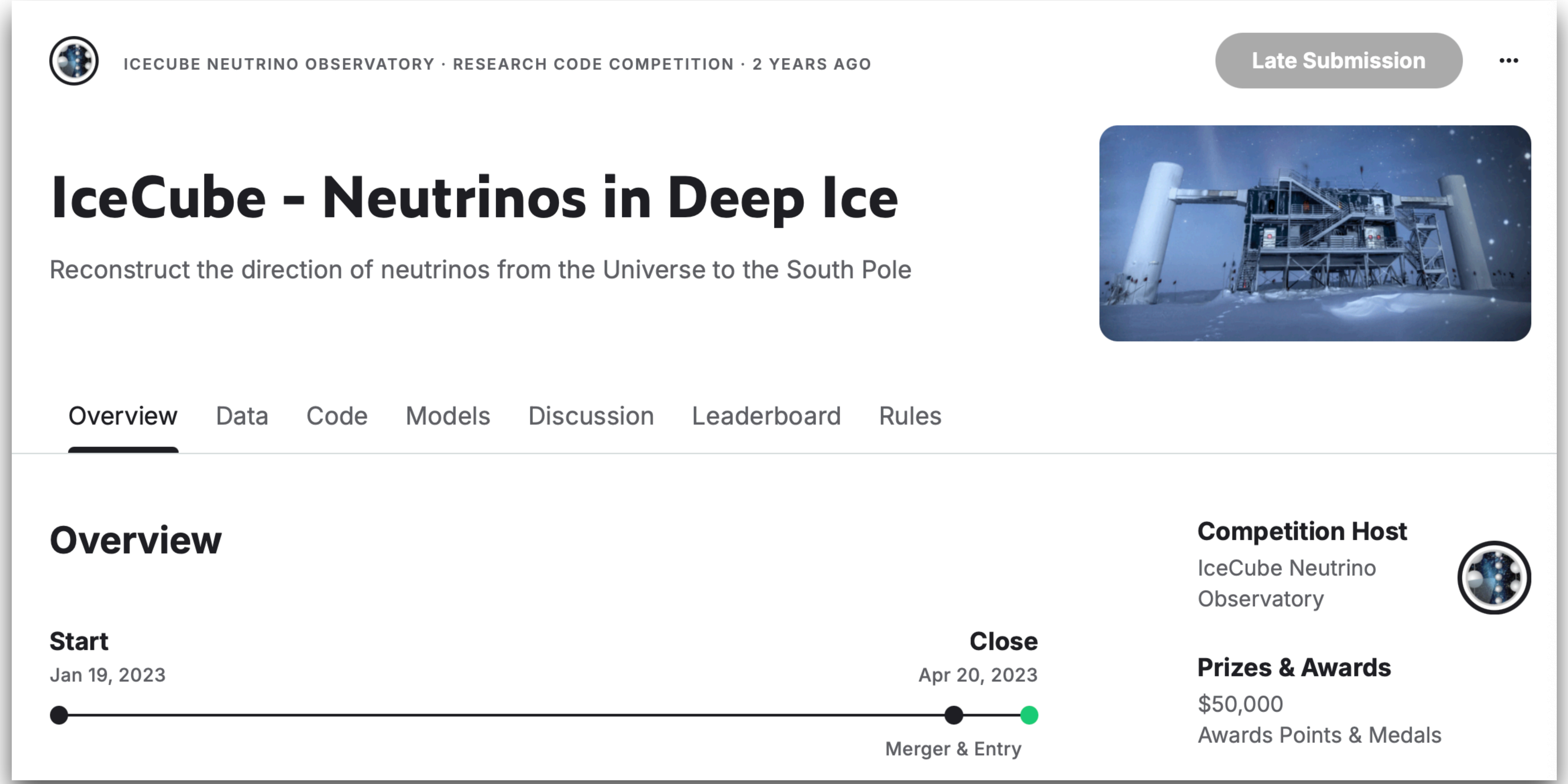
Inverse problem: reconstruct the neutrino direction

- Neutrino energy
- Neutrino direction
(astrophysical sources; identification with galactic plane)
- Traditional methods: likelihood based
- $L(x, y, z, t, \theta, \phi) = p(\text{data} \,|\, x, y, z, t, \theta, \phi)$
- $$L(x, y, z, t, \theta, \phi) = \prod_{j=1}^{N_{DOM}} \prod_{i=1}^{N_{hit}} [p_j(t_i)]^{q_i}$$
t_i - pulse time, q_i - charge
- To maximize the likelihood one has to simulate light propagation through Ice (currently used: arxiv.org/abs/2103.16931)



Machine Learning in IceCube

- Graph Neural Networks for Low-Energy Event Classification & Reconstruction in IceCube
  https://arxiv.org/abs/2209.03042
- A Kaggle competition in 2023
  (901 Participants)
- Kaggle is a specialized platform
  for ML competitions
- Still not better than
  traditional methods at high energies



Can we learn something from LLM progress?

- LLMs benefit from internet-scale datasets.
- Physics also has a lot of data.
    - Both labeled (MC) and unlabeled.
- Can we benefit from unlabeled data?

An exabyte of disk storage at CERN

CERN disk storage capacity passes the threshold of one million terabytes of disk space

29 SEPTEMBER, 2023 | By Tim Smith

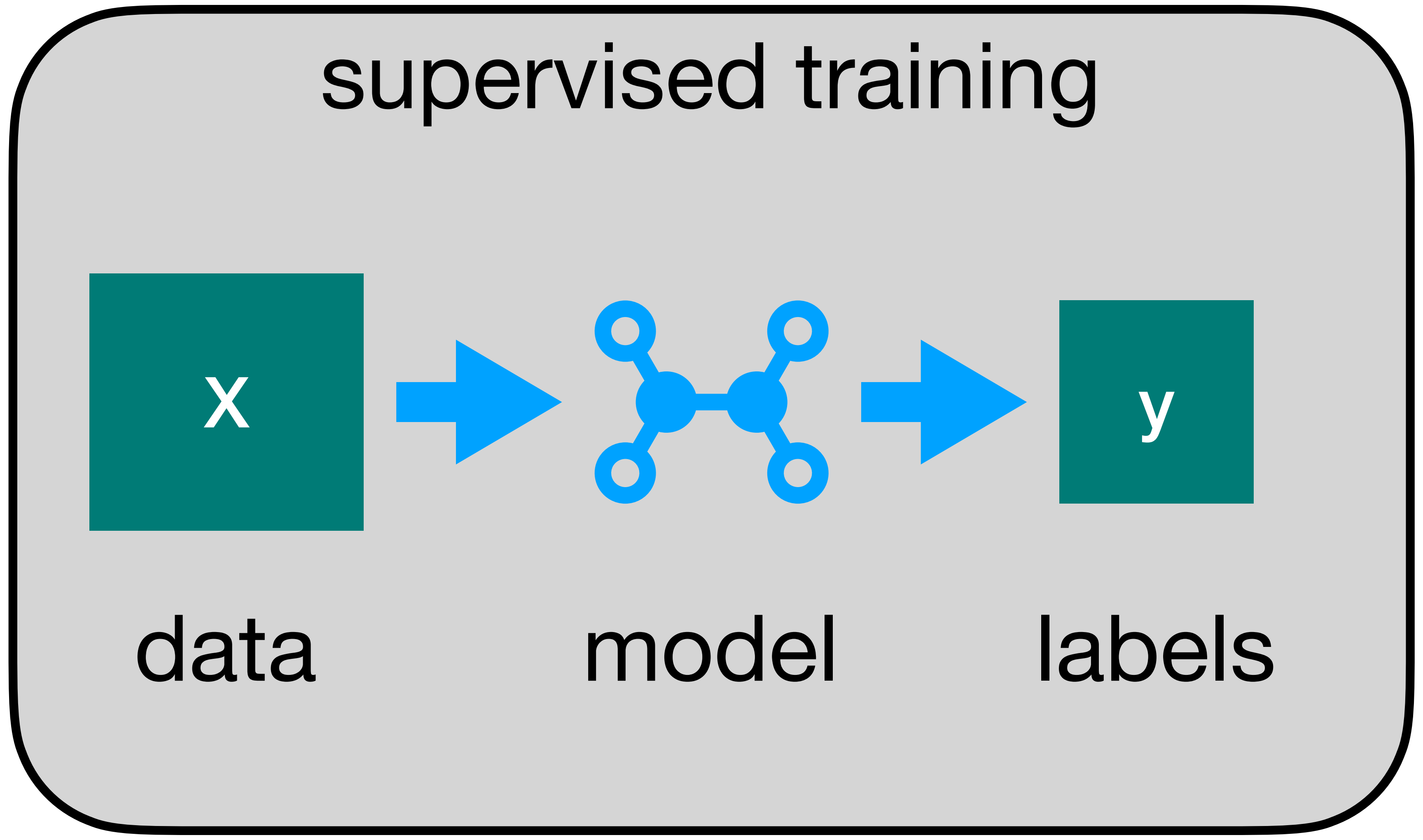
A fraction of the 111 000 devices that form CERN's data storage capacity. (Image: CERN)

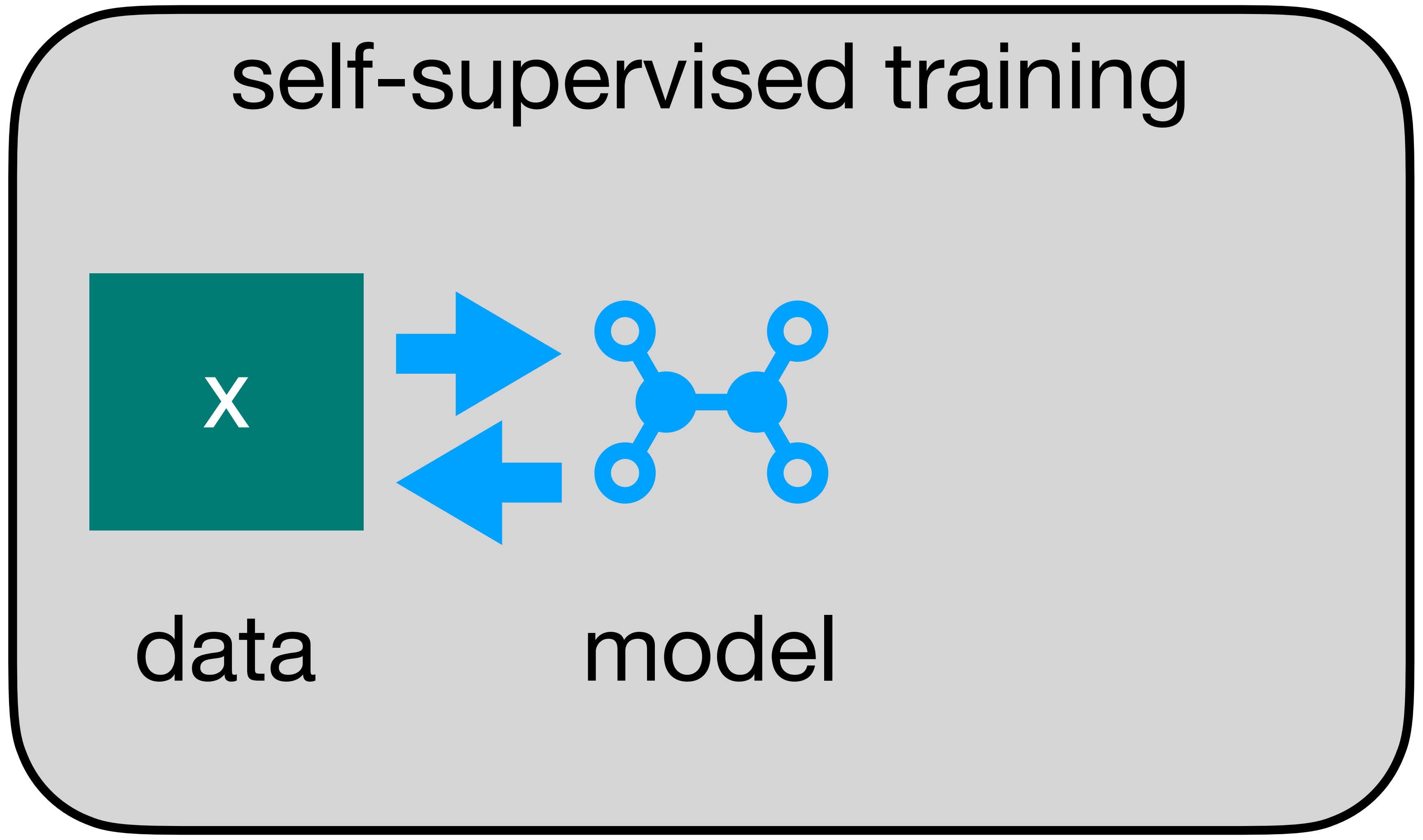
source:
https://home.cern/news/news/computing/exabyte-disk-storage-cern

What do we mean by “foundation models”?

- Initially, the term has been coined for models like BERT and GPT-3
  [2108.07258](https://arxiv.org/abs/2108.07258) “On the Opportunities and Risks of Foundation Models”

- Here, by foundational models we mean the models that are pretrained in a self-supervised way and can be fine-tuned for downstream tasks.

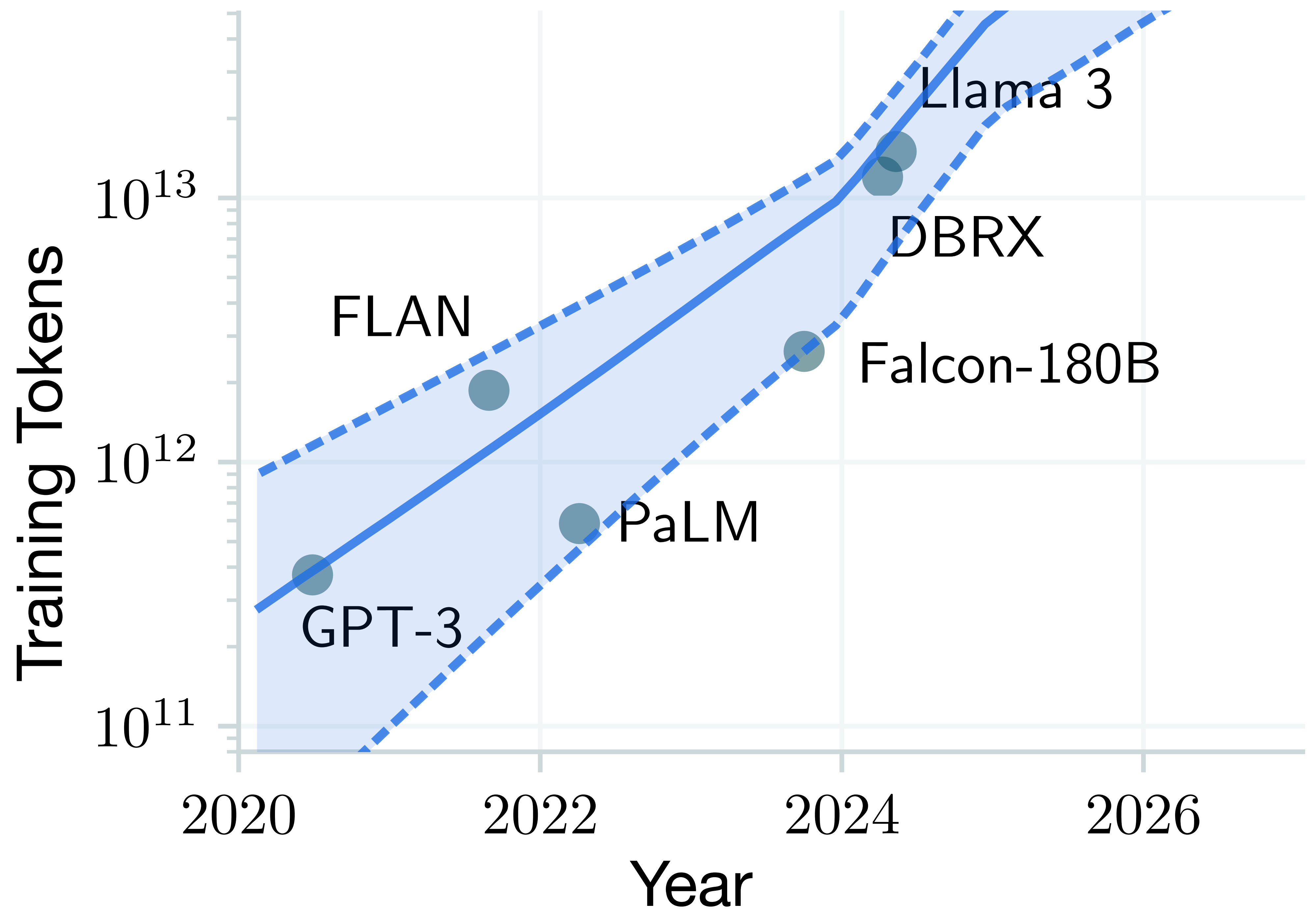




Success of self-supervise training

Outside physics:

- Labeled data is limited
- Unlabeled data is abundant (text, image, video)
- Led to GenAI revolution



source:
2211.04325 "Will we run out of data? Limits of LLM scaling based on human-generated data"

- BERT - 3.3B tokens

1810.04805 "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"

Self-supervise training: Scaling Laws

Performance predictably improves with scale

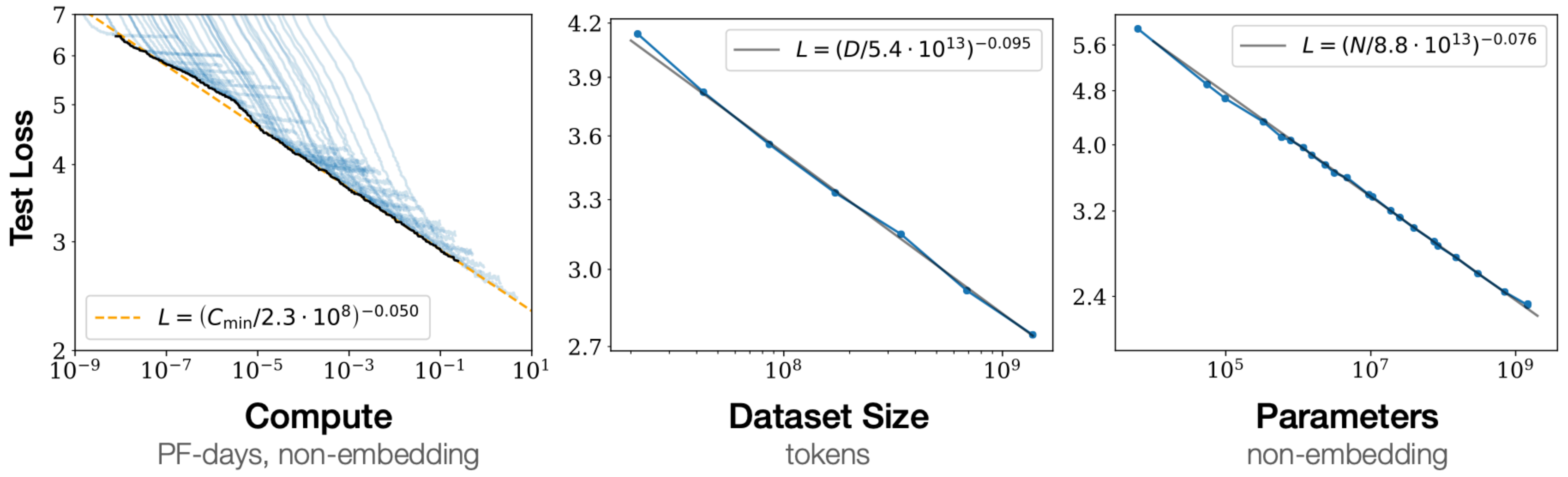


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

https://arxiv.org/pdf/2001.08361
Scaling Laws for Neural Language Models
Jared Kaplan et al

Foundation models in particle physics

(a very incomplete list)

- **Pre-training strategy using real particle collision data for event classification in collider physics**
  https://arxiv.org/abs/2312.06909
  Tomoe Kishimoto, Masahiro Morinaga, Masahiko Saito, Junichi Tanaka

- **Finetuning Foundation Models for Joint Analysis Optimization**
  https://arxiv.org/abs/2401.13536
  Matthias Vigl, Nicole Hartman, Lukas Heinrich

- **Masked Particle Modeling on Sets: Towards Self-Supervised High Energy Physics Foundation Models**
  https://arxiv.org/abs/2401.13537
  Lukas Heinrich, Tobias Golling, Michael Kagan, Samuel Klein, Matthew Leigh, Margarita Osadchy, John Andrew Raine

- **A Language Model for Particle Tracking**
  https://arxiv.org/abs/2402.10239
  Andris Huang, Yash Melkani, Paolo Calafiura, Alina Lazar, Daniel Thomas Murnane, Minh-Tuan Pham, Xiangyang Ju

- **OmniJet-α: The first cross-task foundation model for particle physics**
  https://arxiv.org/abs/2403.05618
  Joschka Birk, Anna Hallin, Gregor Kasieczka

- **Re-Simulation-based Self-Supervised Learning for Pre-Training Foundation Models**
  https://arxiv.org/abs/2403.07066
  Philip Harris, Michael Kagan, Jeffrey Krupa, Benedikt Maier, Nathaniel Woodward

- **OmniLearn: A Method to Simultaneously Facilitate All Jet Physics Tasks**
  https://arxiv.org/abs/2404.16091
  Vinicius Mikuni, Benjamin Nachman

Foundation models in astro and particle physics

- Bumblebee: A Foundation Model for Particle Physics Discovery
  https://ml4physicalsciences.github.io/2024/files/NeurIPS_ML4PS_2024_191.pdf
  (Authors not fully listed in snippet)

- Towards a collaborative approach with Large Language Models and Foundation Models for scientific understanding in fundamental physics
  https://arxiv.org/abs/2501.05382
  (Authors not fully listed in snippet)

- Bridging the Gap: Examining Vision Foundation Models for Optical and Radio Astronomy Applications
  https://arxiv.org/abs/2409.11175
  E. Lastufka, O. Bait, M. Drozdova, V. Kinakh, D. Piras, M. Audard, M. Dessauges-Zavadsky, T. Holotyak, D. Schaerer, S. Voloshynovskiy

- AstroCLIP: A Cross-Modal Foundation Model for Galaxies
  https://arxiv.org/abs/2310.03024
  Liam Parker, Francois Lanusse, Siavash Golkar, Leopoldo Sarra, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Geraud Krawezik, Michael McCabe, Ruben Ohana, Mariel Pettee, Bruno Regaldo-Saint Blancard, Tiberiu Tesileanu, Kyunghyun Cho, Shirley Ho

- Towards an astronomical foundation model for stars with a Transformer-based model
  https://arxiv.org/abs/2308.10944
  Henry W. Leung, S. G. Djorgovski

- Self-Supervised Learning Strategies for Jet Physics
  https://arxiv.org/abs/2503.11632
  Patrick Rieck, Kyle Cranmer, Etienne Dreyer, Eilam Gross, Nilotpal Kakati, Dmitrii Kobylanskii, Garrett W. Merz, Nathalie Soybelman

- HEP-JEPA: A Joint Embedding Predictive Architecture for a Foundation Model in High Energy Physics
  https://arxiv.org/abs/2502.03933
  (Authors not fully listed in snippet)

- Enhancing Masked Particle Modeling for High Energy Physics Foundation Models
  https://arxiv.org/abs/2409.12589
  (Authors not fully listed in snippet)

- A Foundation Model for Event Classification in High-Energy Physics
  https://arxiv.org/abs/2412.10665
  (Authors not fully listed in snippet)

- Large-scale Pretraining and Finetuning for Efficient Jet Classification in Particle Physics
  https://arxiv.org/abs/2408.09343
  (Authors not fully listed in snippet)

- Enabling Unsupervised Discovery in Astronomical Images through Self-Supervised Representations
  https://arxiv.org/abs/2311.14157
  Koketso Mohale, Michelle Lochner

- Data Compression and Inference in Cosmology with Self-Supervised Machine Learning
  https://arxiv.org/abs/2308.09751
  Aizhan Akhmetzhanova, Siddharth Mishra-Sharma, Cora Dvorkin

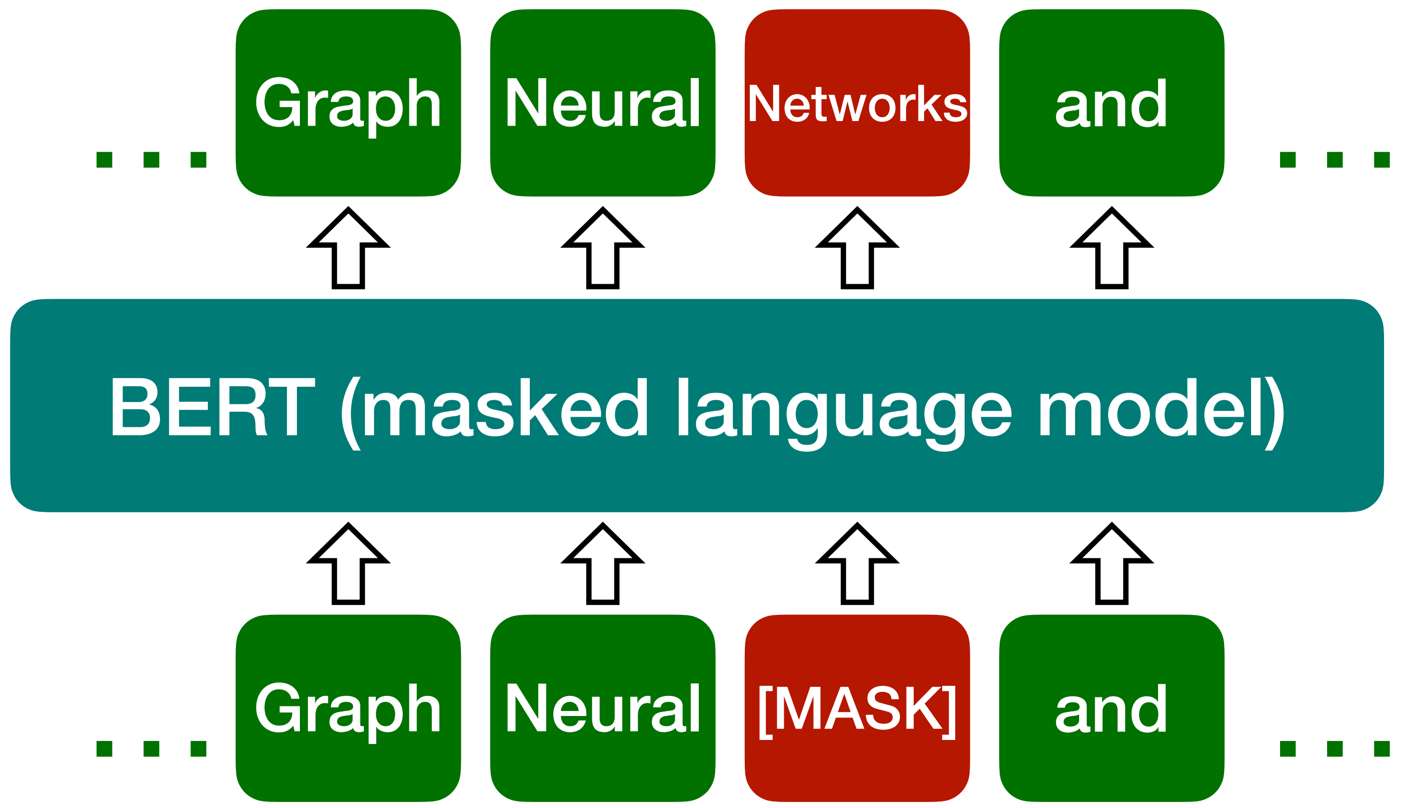
- AstroM^3: A self-supervised multimodal model for astronomy
  https://arxiv.org/abs/2411.08842
  Mariia Rizhko, Joshua S. Bloom

See Gemini Report

Challenges of self-supervise learning in particle physics

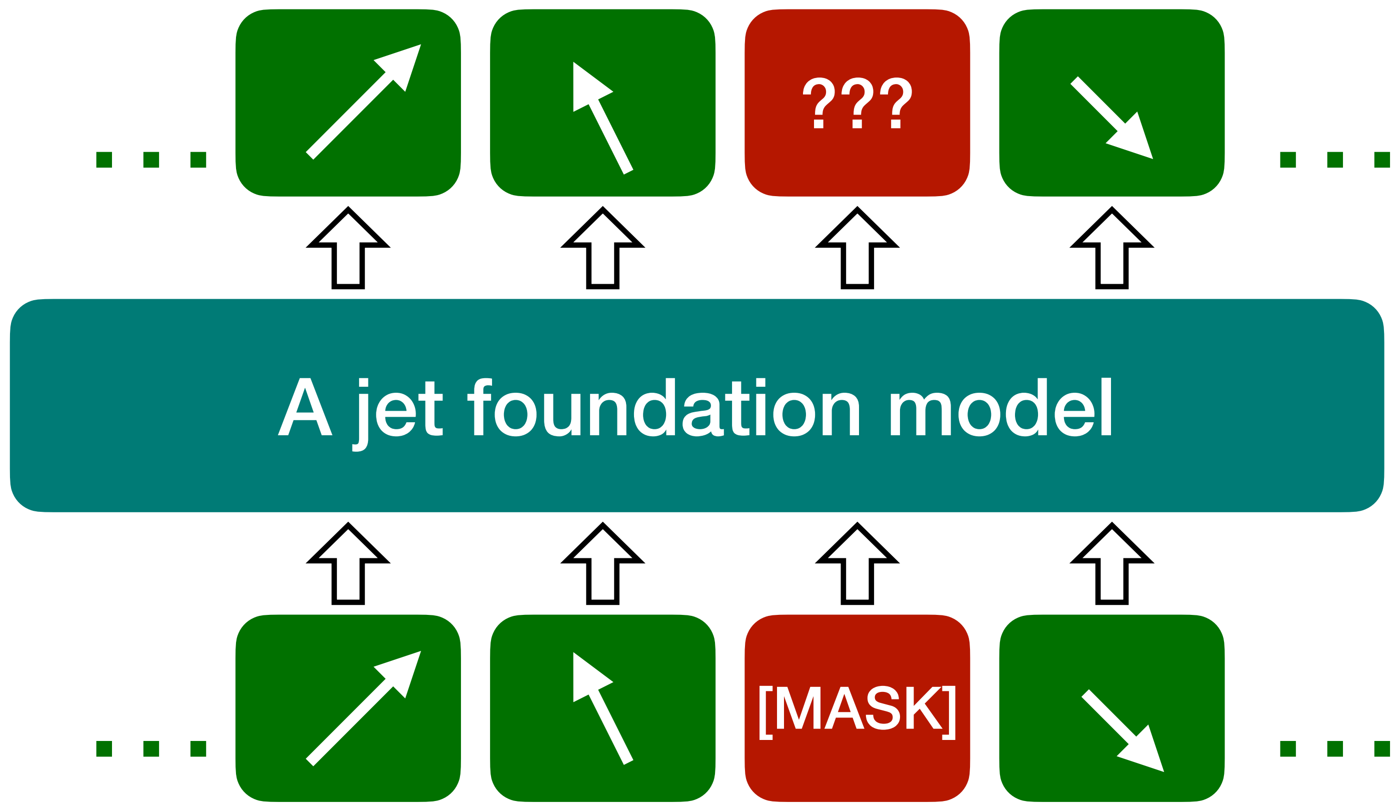
BERT
(Bidirectional Encoder Representations from Transformers)

predict the distribution of a token from a discrete set



A jet foundation model

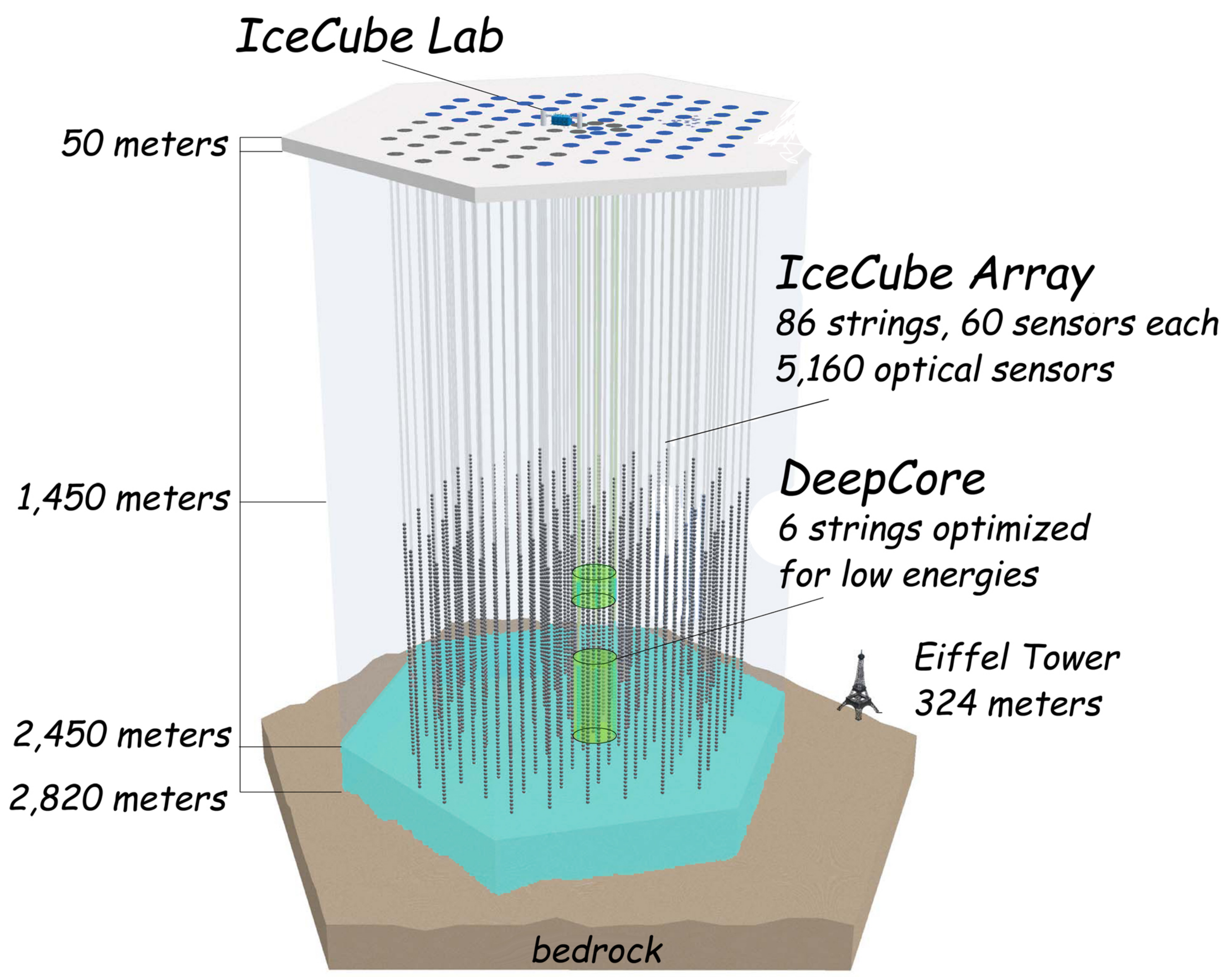
How to predict a continuous 4-vector?



Usually lossy discretization:
- VQ-VAE (2401.13537, 2403.05618)
- pixelization (2402.10239)

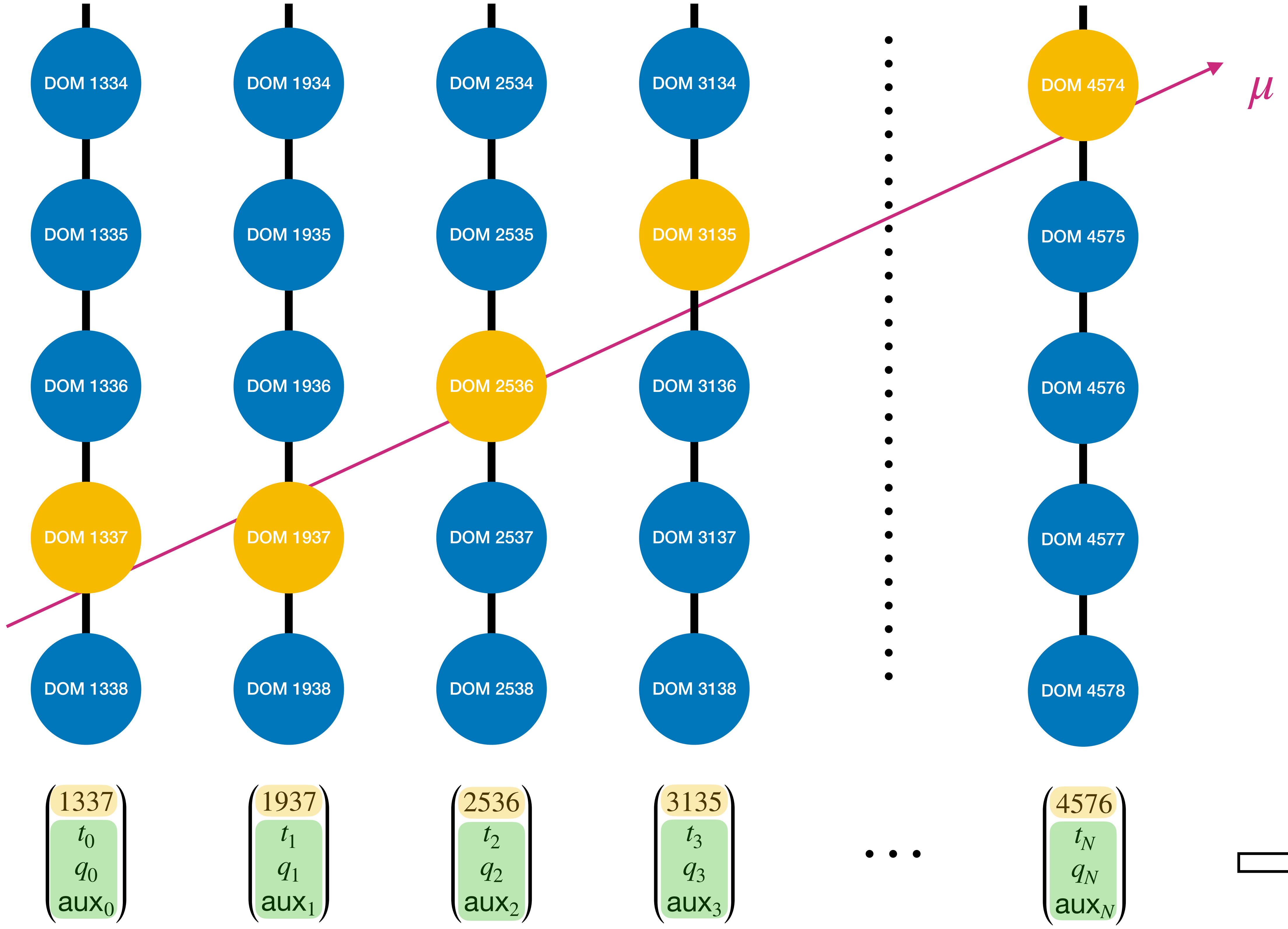
Challenges of self-supervise learning in particle physics

- How to predict a continuous 4-vector?
- Usually lossy discretization:
  - VQ-VAE (2401.13537, 2403.05618)
  - pixelization (2402.10239)
- How to sort 4-vectors?
- IceCube
  - 5160 DOMs — natural "tokenization"
  - Pulses have timestamps



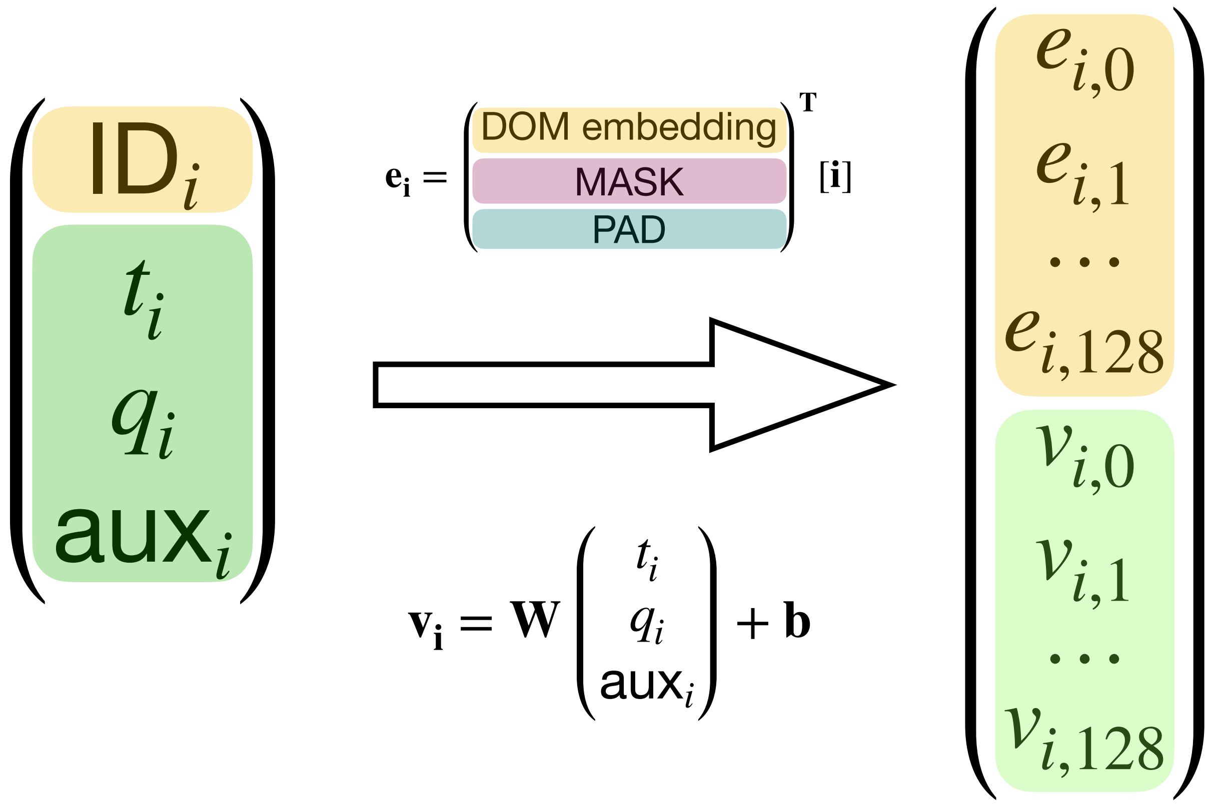
IceCube Embedding

linear layer transforming DOM x,y,z coordinates
works better for directional reconstruction

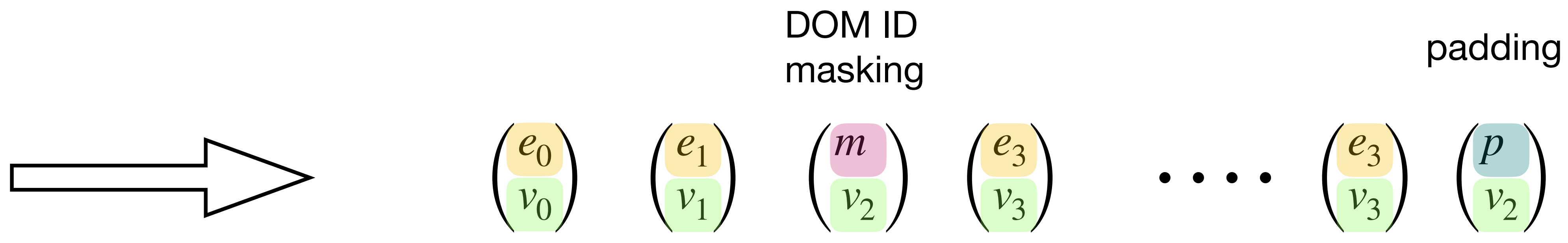


pulses (arranged by time)

pulse embedding

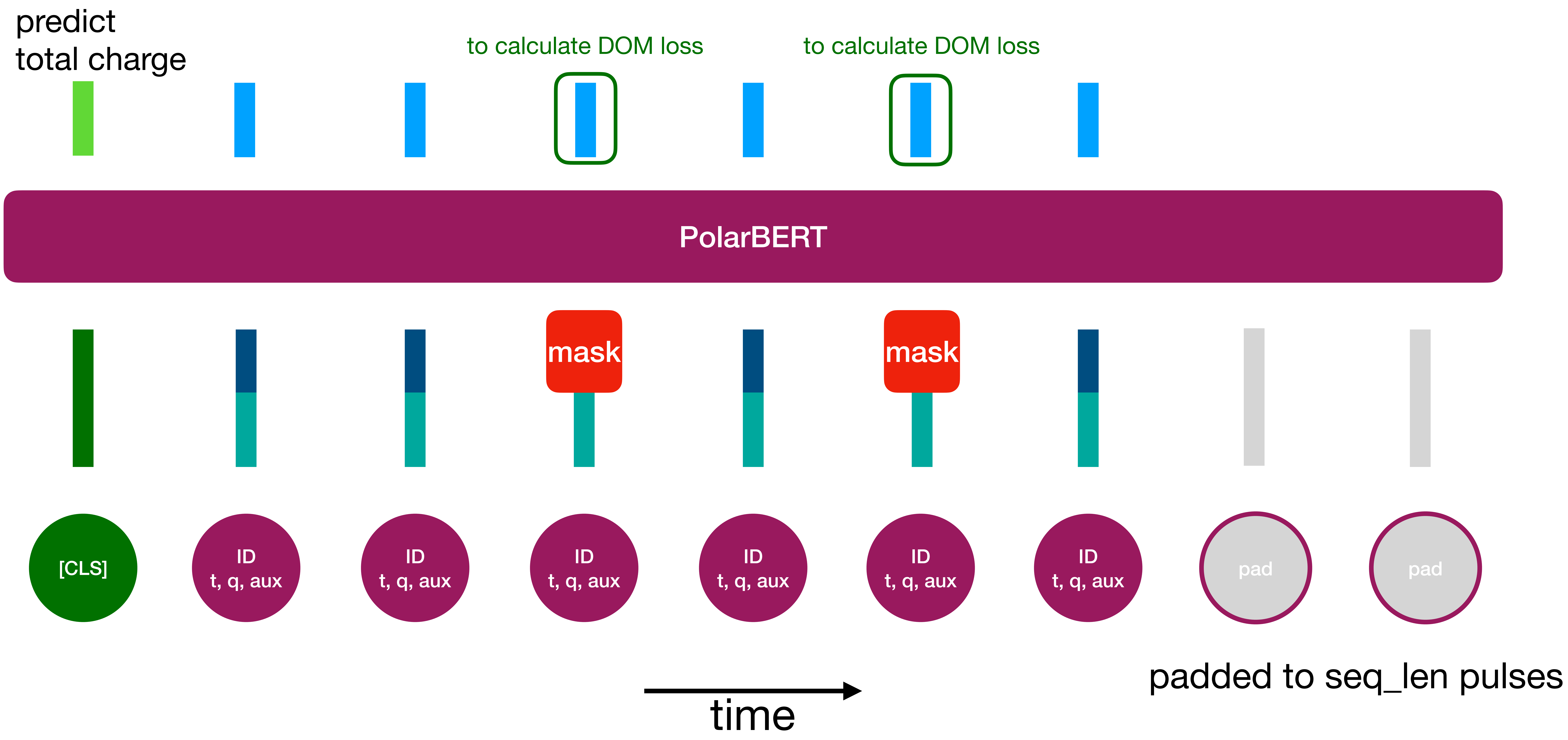


No position data!



time-series (padded to fixed length)

Pretraining

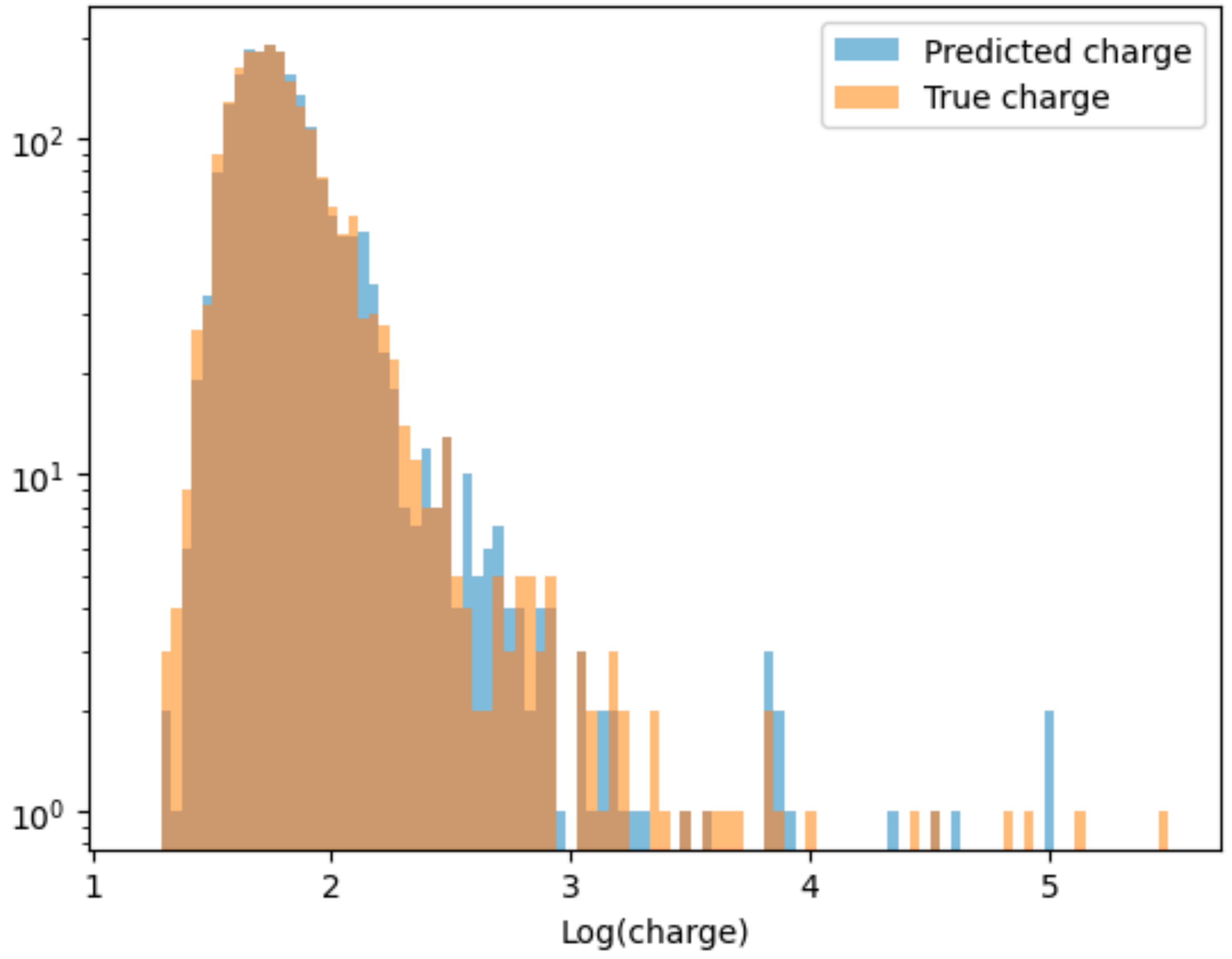


Pretraining: DOM loss

- The detection process is inherently stochastic
- We cannot predict the next DOM with certainty
- Similarly to LLMs, we use cross-entropy
  (but other option are possible: Earth Mover's Distance, Chamfer distance)
- DOM-loss: $L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\log(p_i)$, the sum over N masked doms
- Use only aux=false (HLC) pulses! aux=true pulses are impossible to predict.

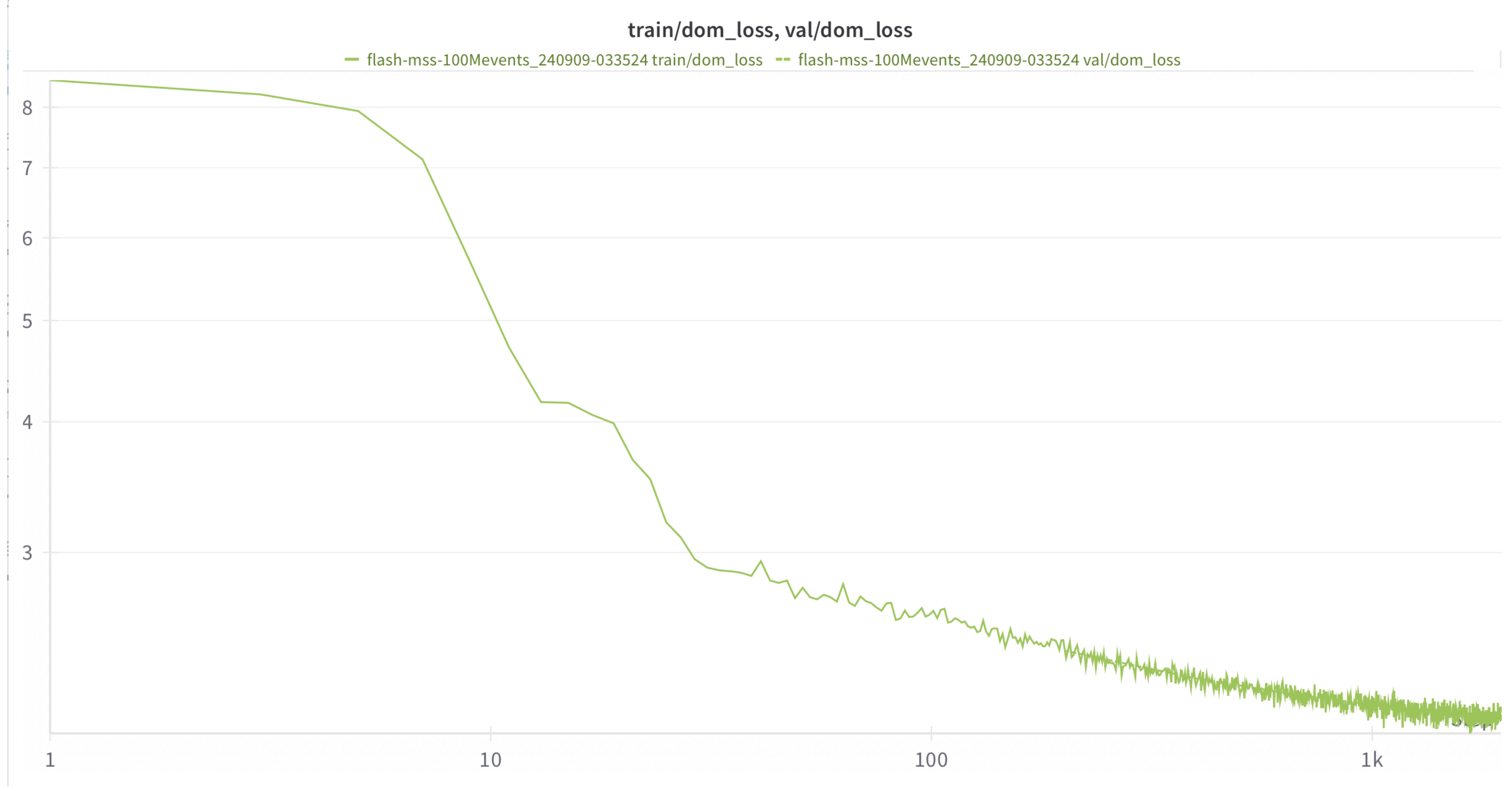
Pretraining: regression loss

- The model has to learn how to collect useful information in [CLS] embedding for the future use on downstream tasks.
- We need some feature that is not directly accessible to the model, but can be obtained from the data (no labels)
- Candidates: the total charge of the event, center of charge
- We subsample the events, and the charge is provided as a log
- Charge prediction loss: MSE($\log$(total charge))



PolarBERT: Foundation Model For IceCube

- Backbone: transformer (could be GRU, Mamba)
- Pretraining:
    - Subsample events to seq_len (currently 128)
    - input: (DOM projections) $\oplus$ (projection of features)
    - loss function = DOM-loss + $\lambda \times$ charge-prediction-loss
- Fine-tuning for downstream tasks
- IceCube kaggle MC data for both pretraining and finetuning (studies using real data can be only published by the collaboration)



100 million events

BERT: 3,300M tokens
PolarBERT: 12,700M “tokens”
(100M events x 127 pulses)

PolarBERT: Foundation Model For IceCube

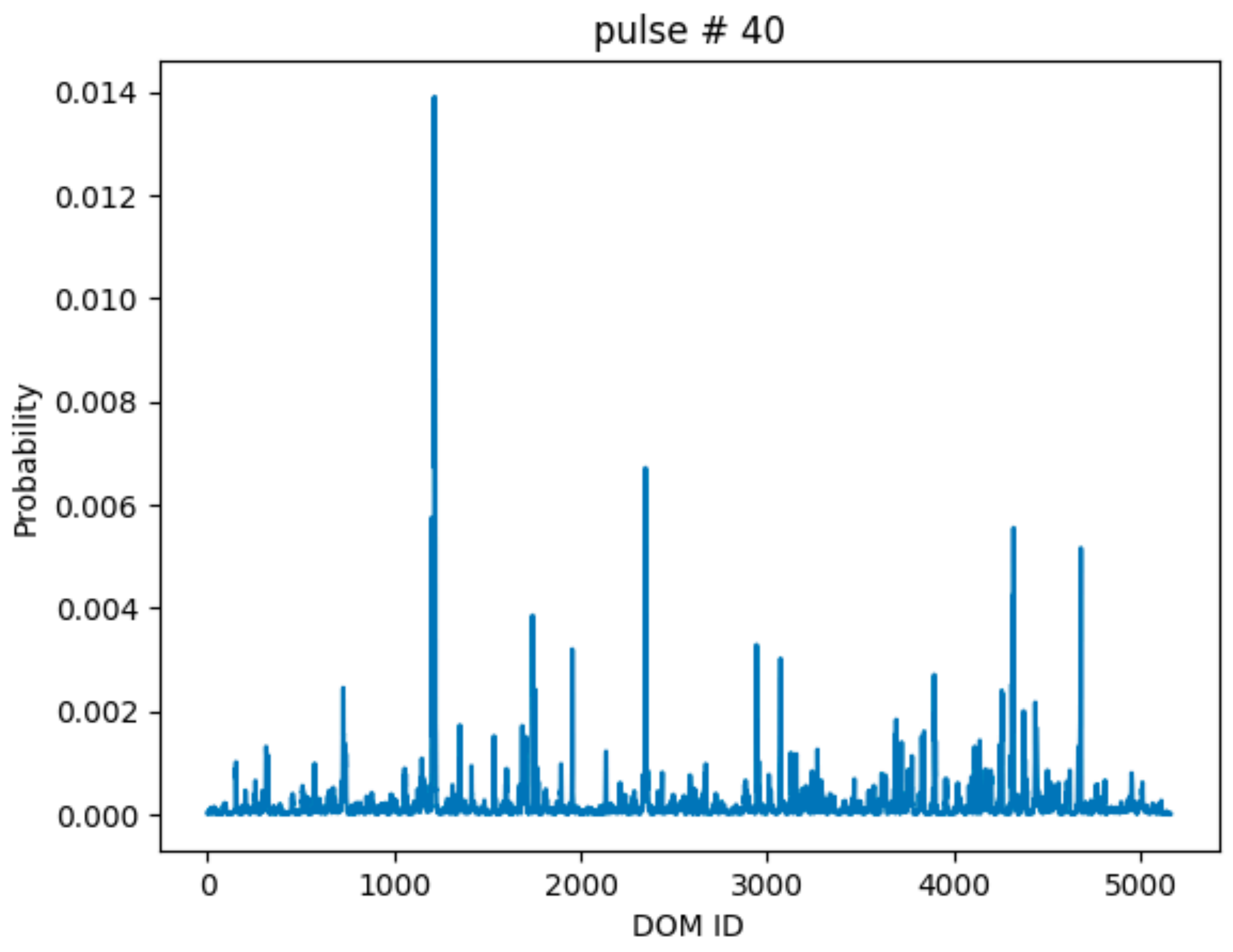
- Backbone: transformer (could be GRU, Mamba)
- Pretraining:
  - Subsample events to seq_len (currently 128)
  - input: (DOM projections) $\oplus$ (projection of features)
  - loss function = DOM-loss + $\lambda \times$ charge-prediction-loss
- Fine-tuning for downstream tasks
- IceCube kaggle MC data for both pretraining and finetuning (studies using real data can be only published by the collaboration)

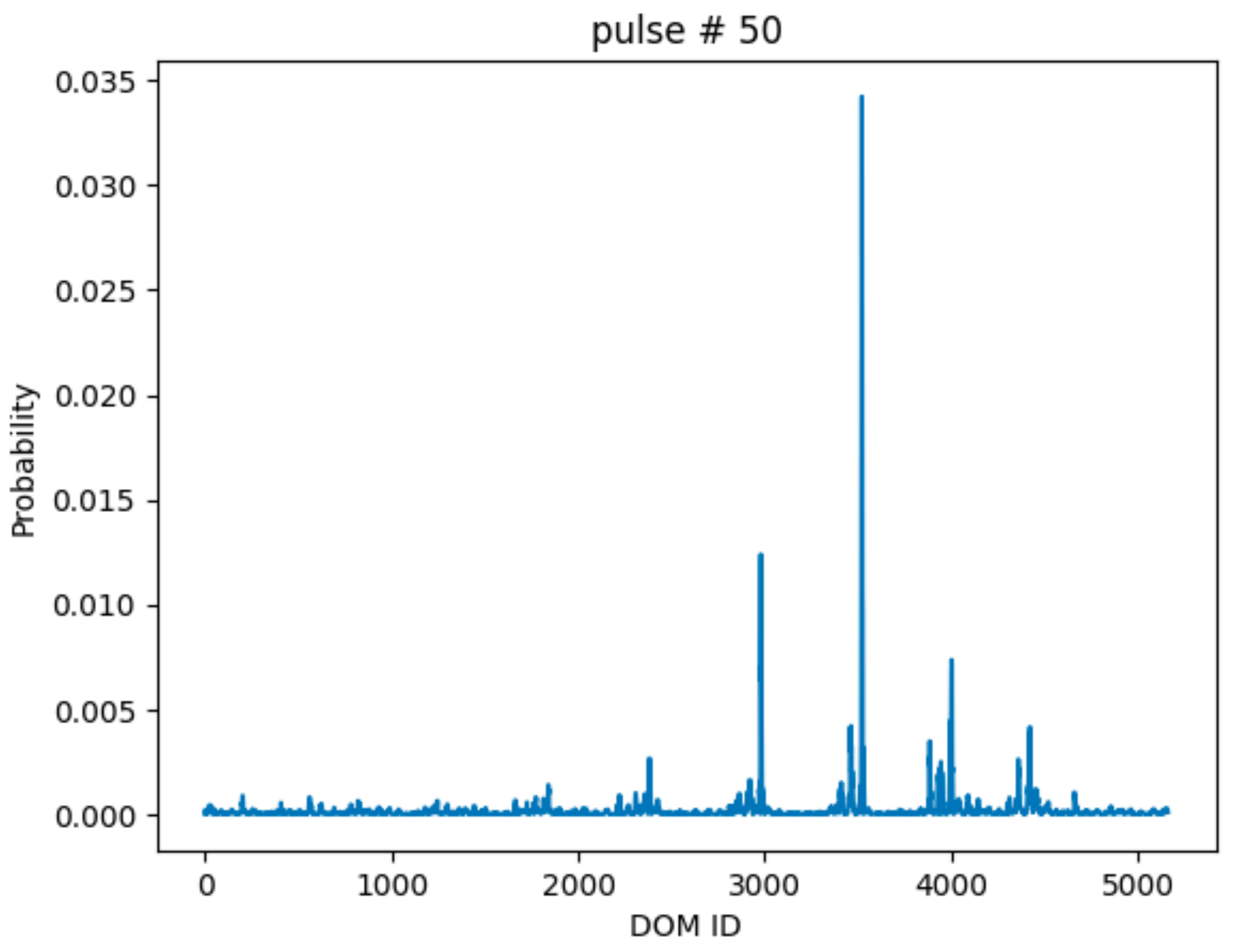
```yaml
model:
  use_dom_positions: true
  embedding_dim: 256
  dom_embed_dim: 128
  num_heads: 8
  hidden_size: 1024
  num_layers: 8
  lambda_charge: 1.0
  directional:
    hidden_size: 1024
```

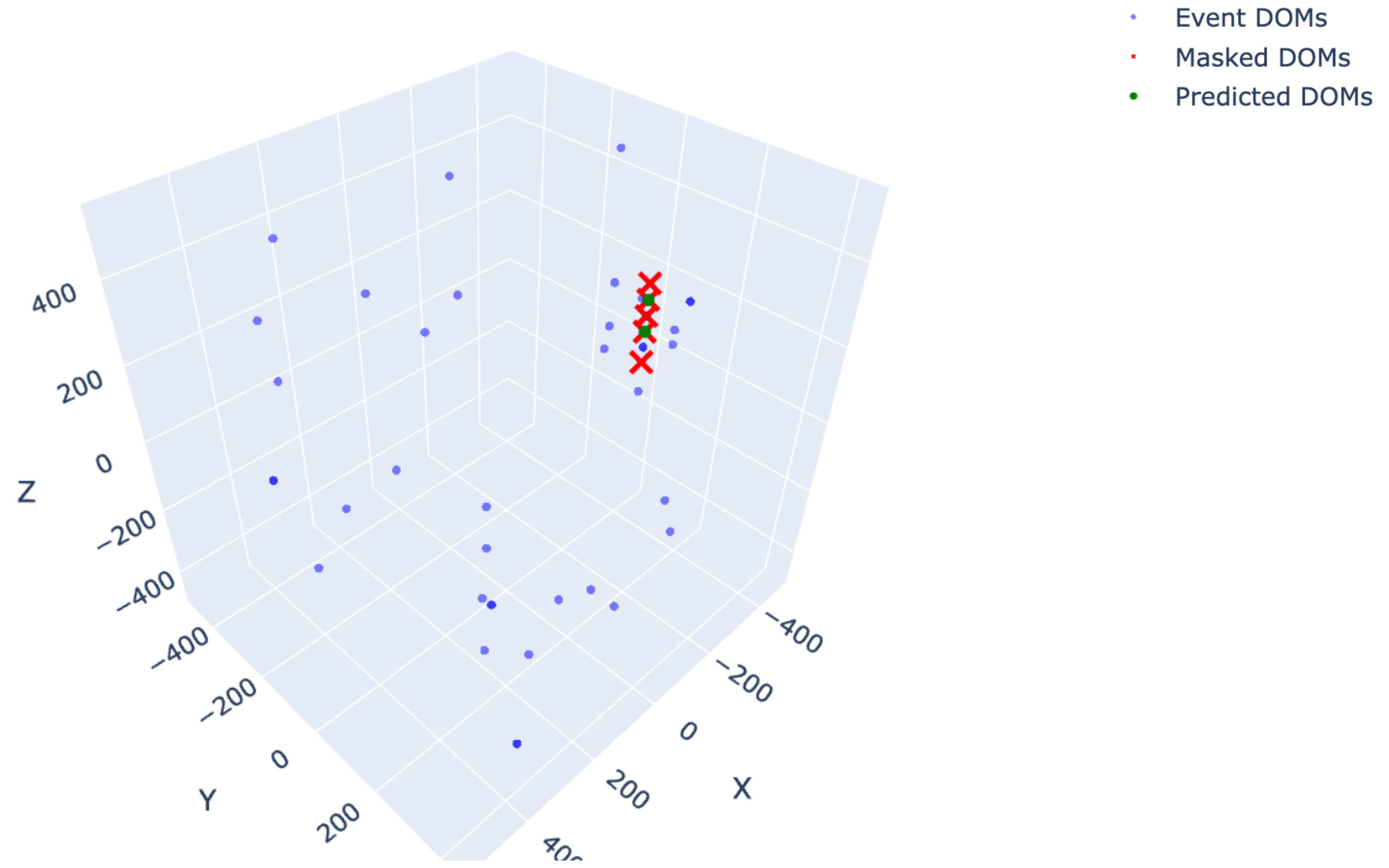
a typical model (7.6M params)
see the Config

Interpreting the DOM Loss

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\log(p_i)$$



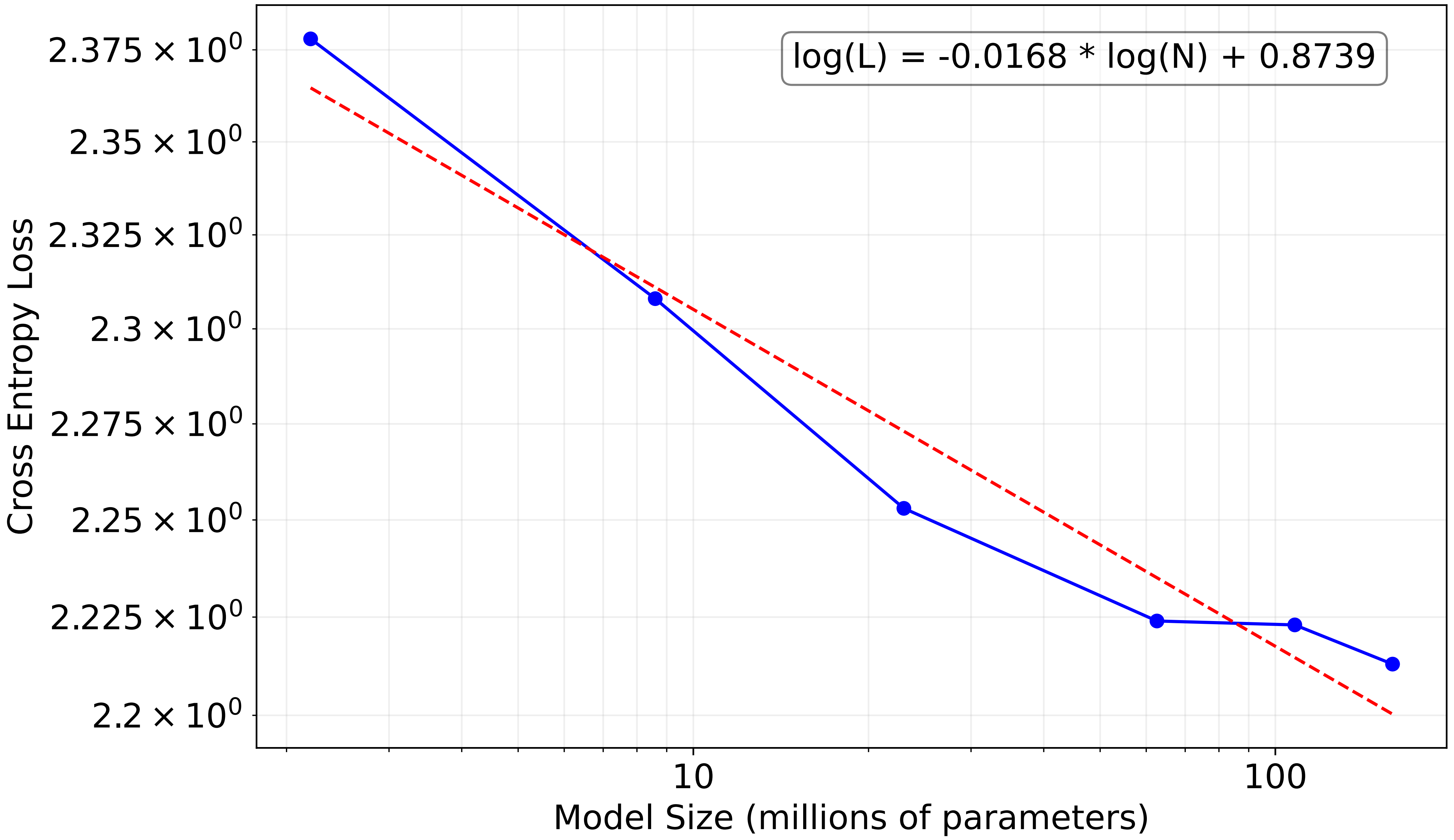




some uncertainty about the string and the DOM

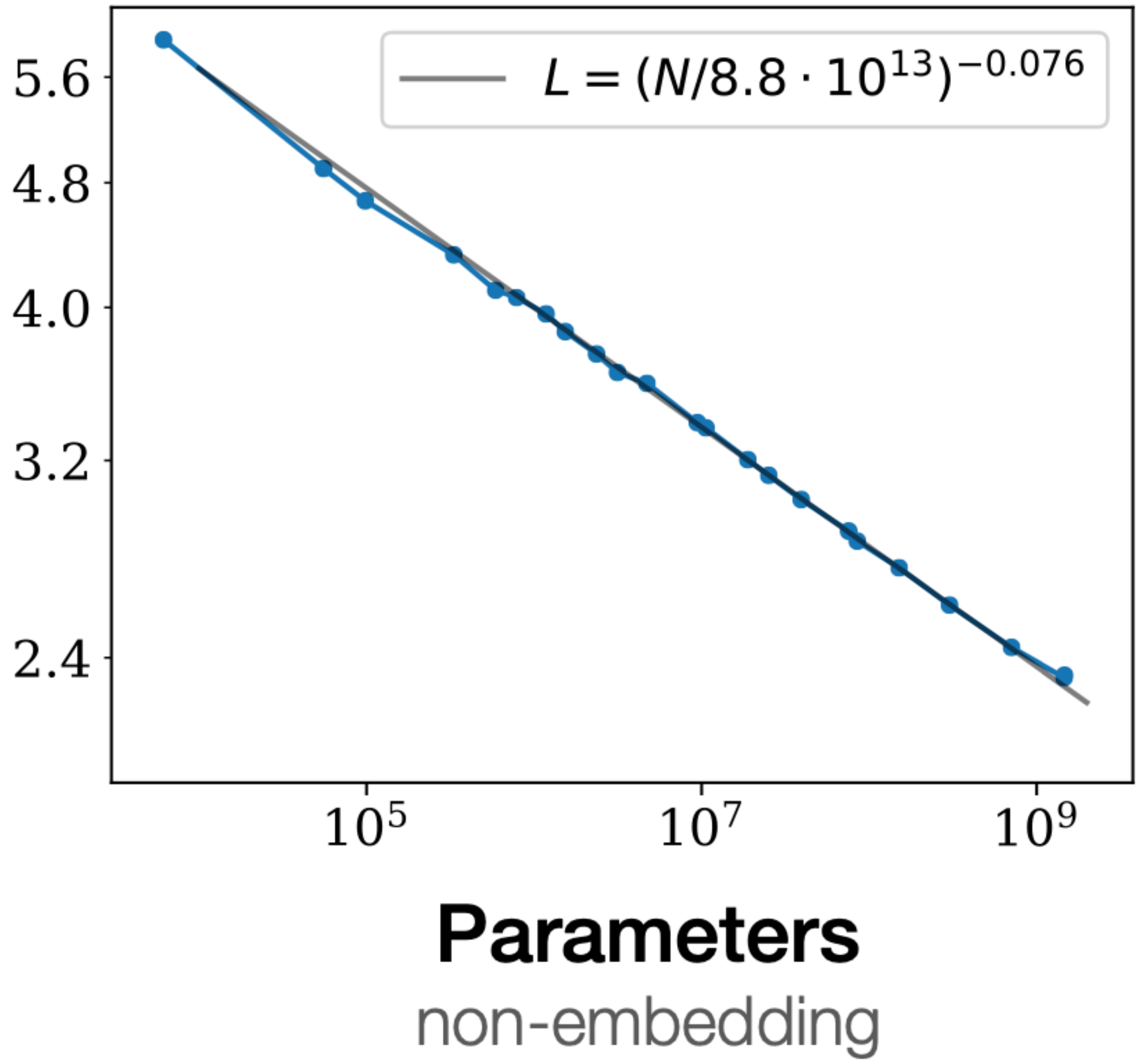
Model Size Scaling

PolarBERT



Models trained on 10M neutrino events

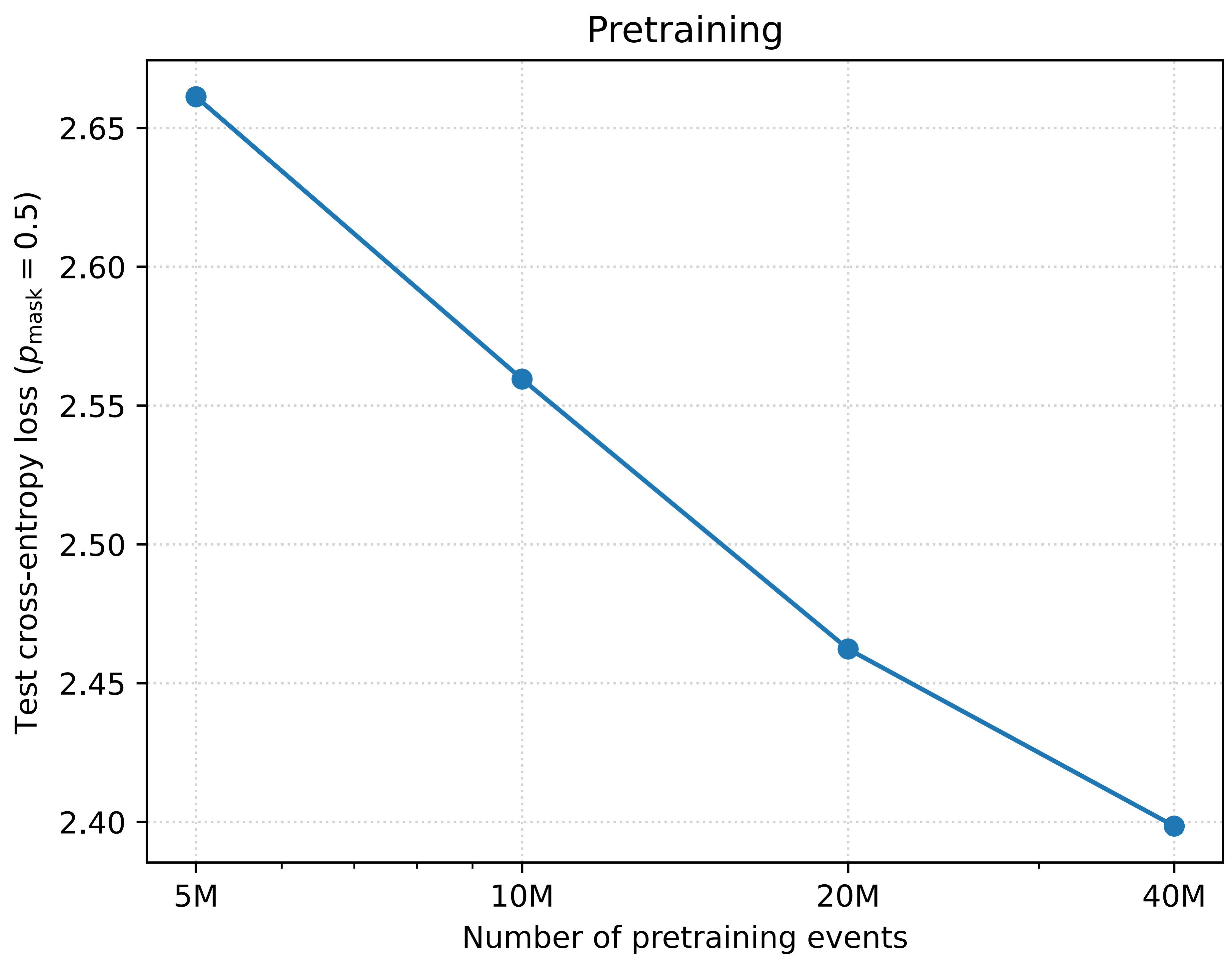
LLMs



Models trained to convergence
Kaplan et all, 2020

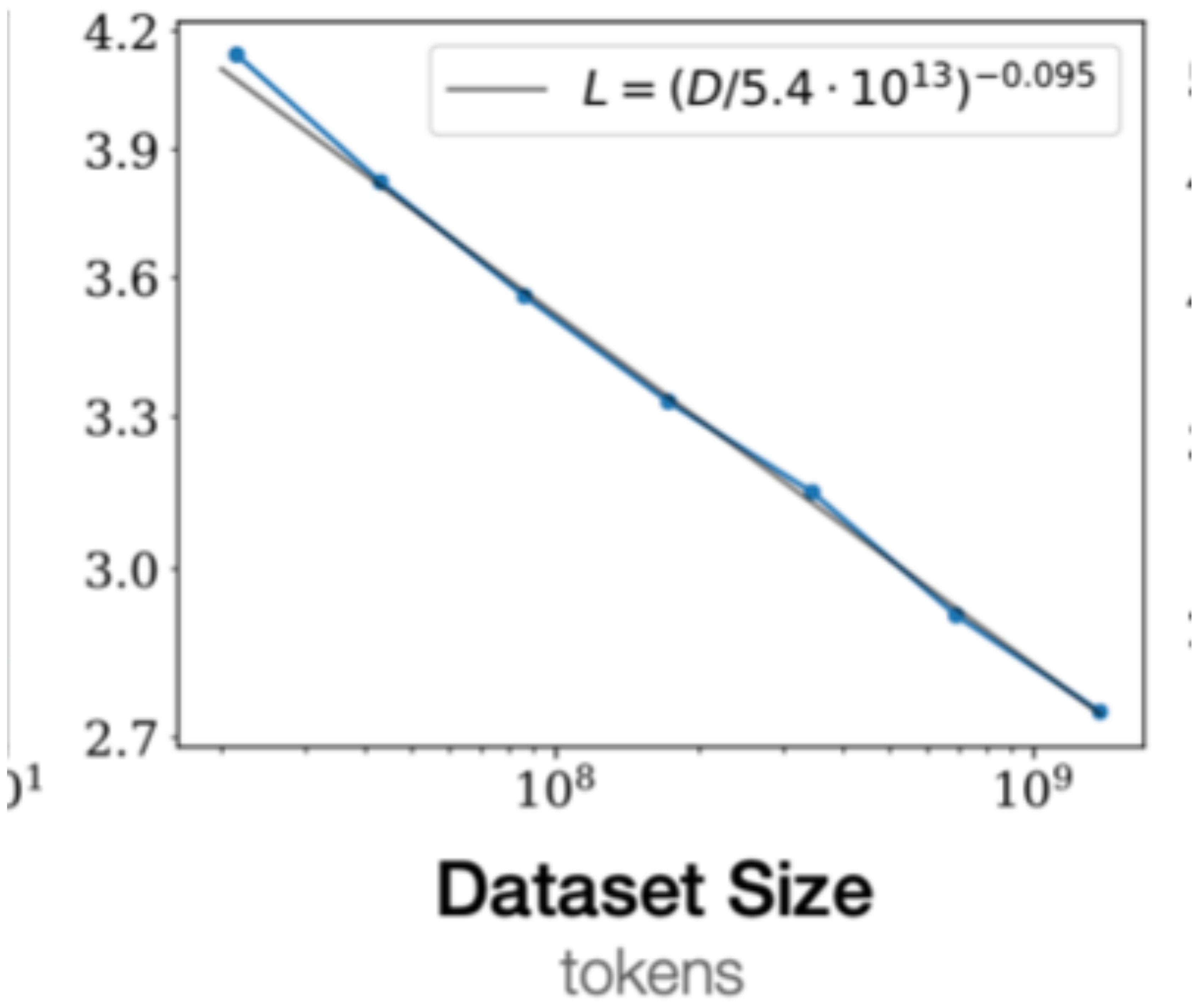
Dataset Size Scaling

PolarBERT



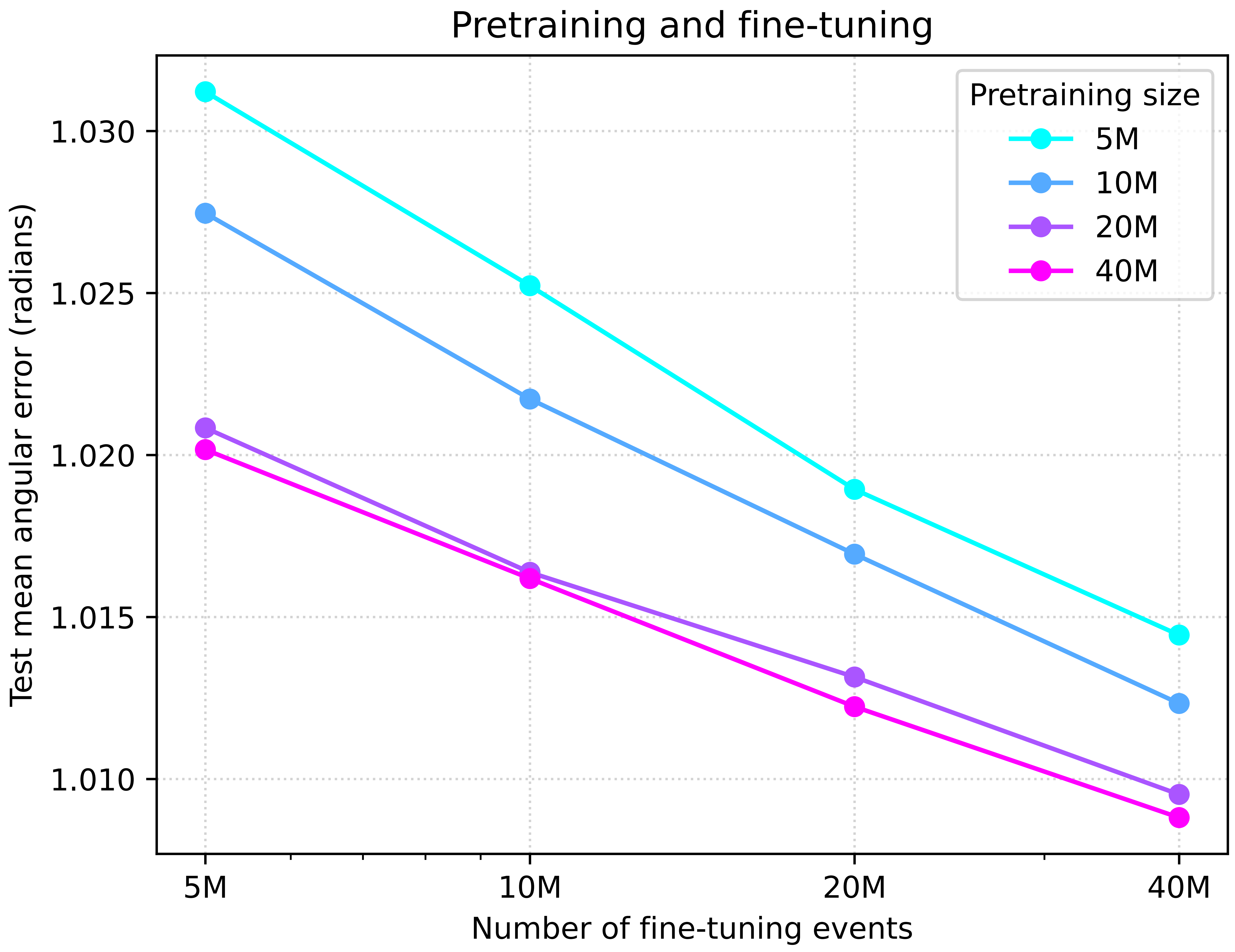
7.6M Models

LLMs



Models trained to convergence
Kaplan et all, 2020

Finetuning (Directional Reconstruction)



- Pretrained model can be successfully fine-tuned on a downstream task.
- We add a “prediction head”: an MLP to the [CLS] embedding output.
- Train the resulting model with direction labels.
- Fine-tuning is sample-efficient.
- When tuned on the full Kaggle dataset, the mean angular error is **0.984.** This corresponds to a Kaggle silver medal.
- Results with a 6.6M model. We expect improvement with the size.

Takeaways

- We can leverage unlabeled data for IceCube direction reconstruction.
- Our foundation model, PolarBert, is competitive with Kaggle models.
- Scaling also works in physics (but with smaller exponents).

Technical Bits

- **Experiments**
  - Comparing apples to apples is hard.
  - One has to tune hyperparameters of all models!
  - Comparing models trained with the same hyperparameters could be misleading!
  - Technically, wandb sweeps are convenient.
    But still hard to interpret, since parameters correlate.
  - See recipes here: https://github.com/google-research/tuning_playbook
  - For scaling μP is useful (see here https://github.com/EleutherAI/nanoGPT-mup)

Technical Bits

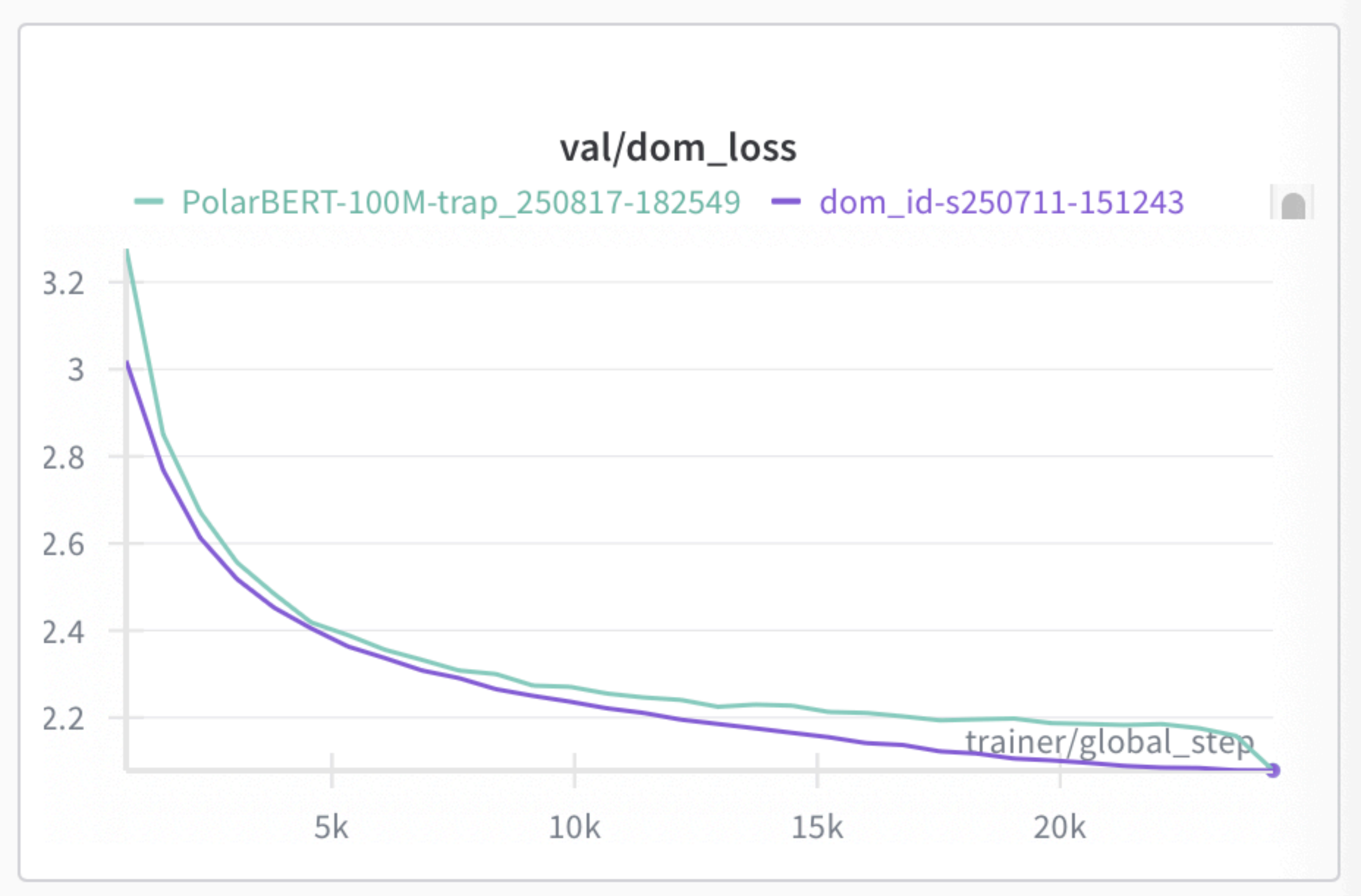
- **LR Schedule**

  - LR Schedule (warmup with ~1/(1-beta) steps, annealing) significantly improves the performance

  - *Cosine schedule* is very popular.
    * Great results
    * Hard to compare different dataset sizes
    * hard to tune the parameters (many correlations)

  - *Trapezoidal schedule*
    * Similar performance (last ~1000 annealing steps are important)
    * Better for parameter tuning and model comparison

Technical Bits



trapezoidal (green) vs cosine (purple) schedule