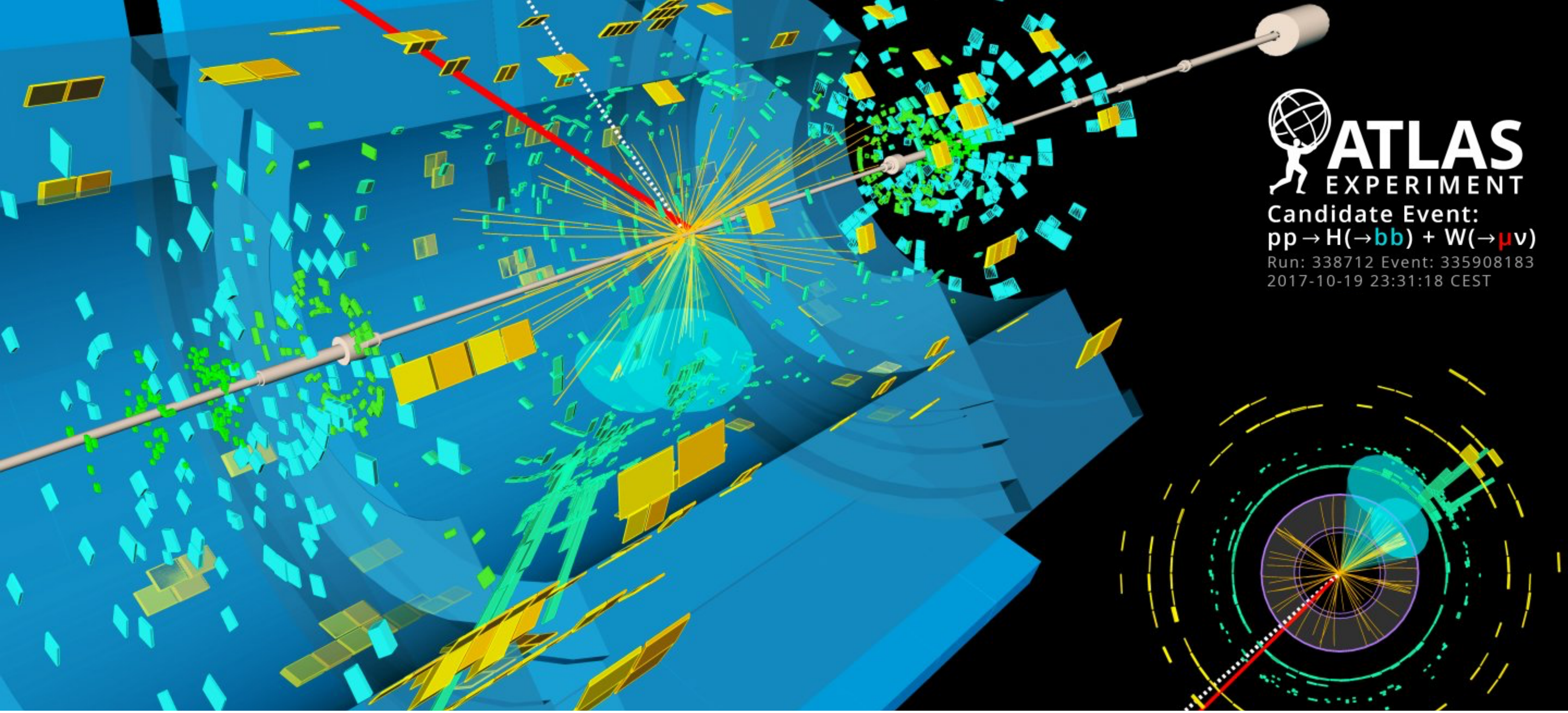


Deep Learning in Data Analysis: Introduction to Deep Learning in HEP

Lecture 1

Sofia Vallecorsa | Ilaria Luise

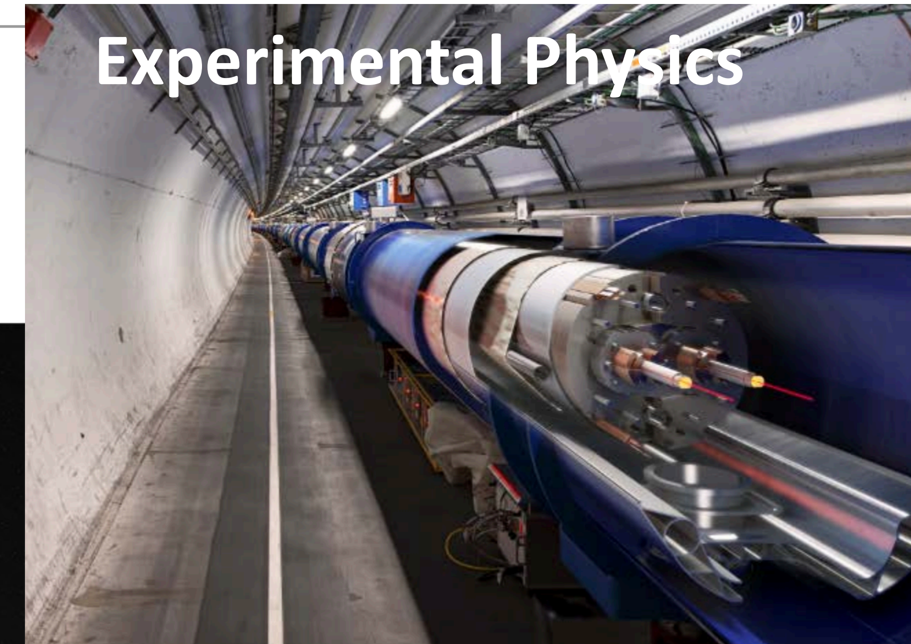
Thematic CERN School of Computing on Machine Learning
17th October 2024

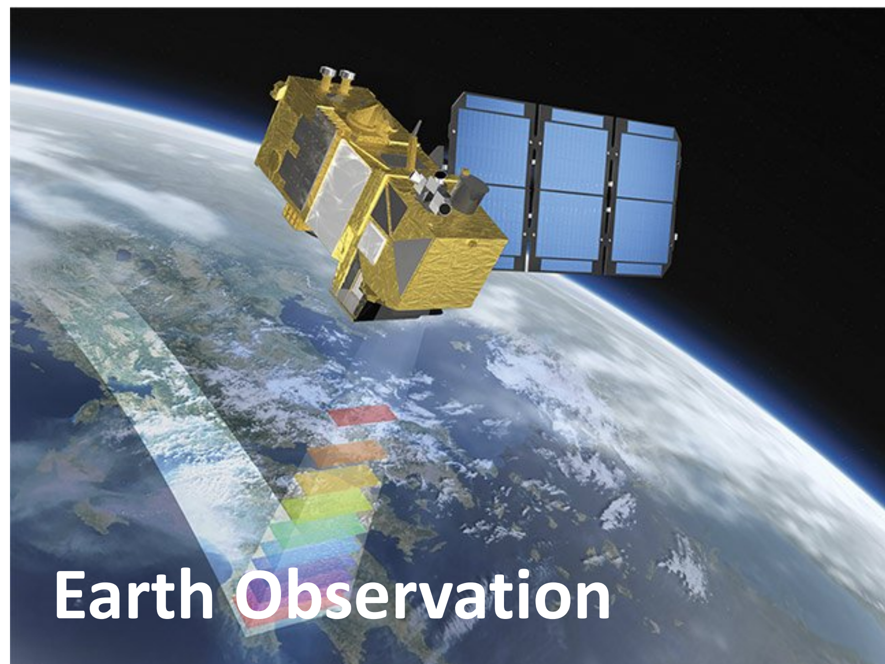
Outline

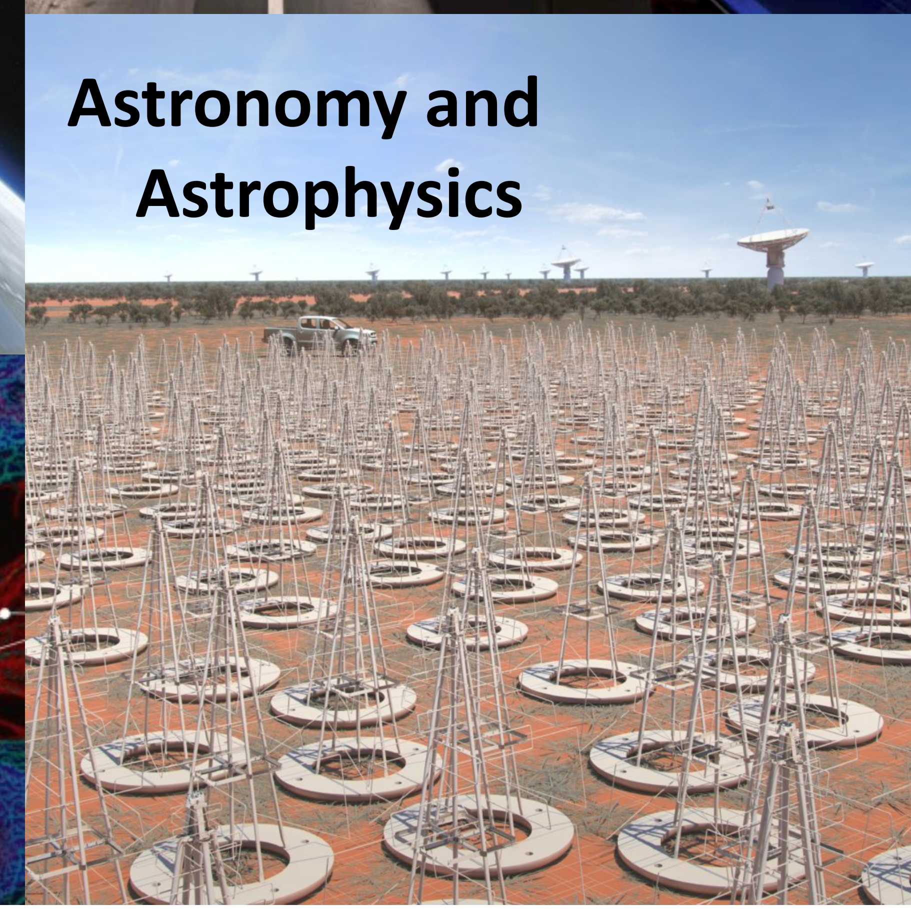
- Introduction
    - The need for depth - graphs complexity
- Computational challenges
- Generative Models

Big Data in Science

Science produces more data than ever before and at an unmatched pace in history



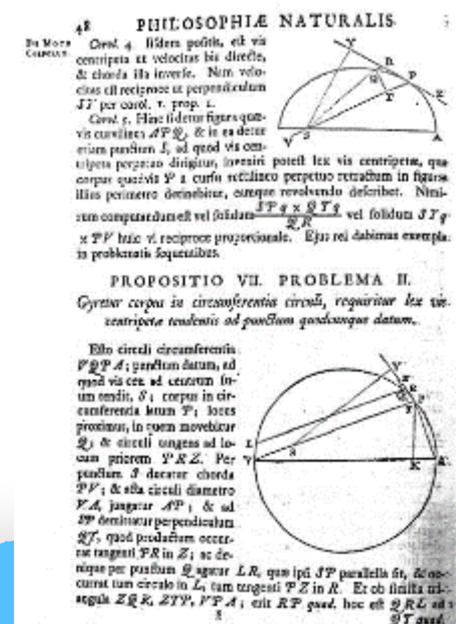




Four Paradigms of Scientific Research

4000 years

Empirical observations

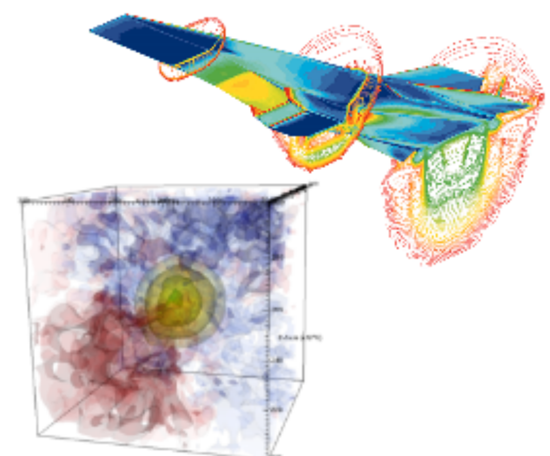


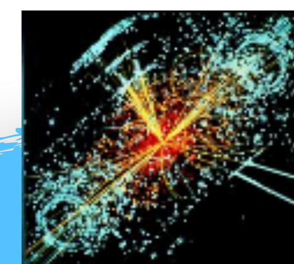
PHILOSOPHIÆ NATURALIS.

PROPOSITIO VII. PROBLEMA II.

500 years

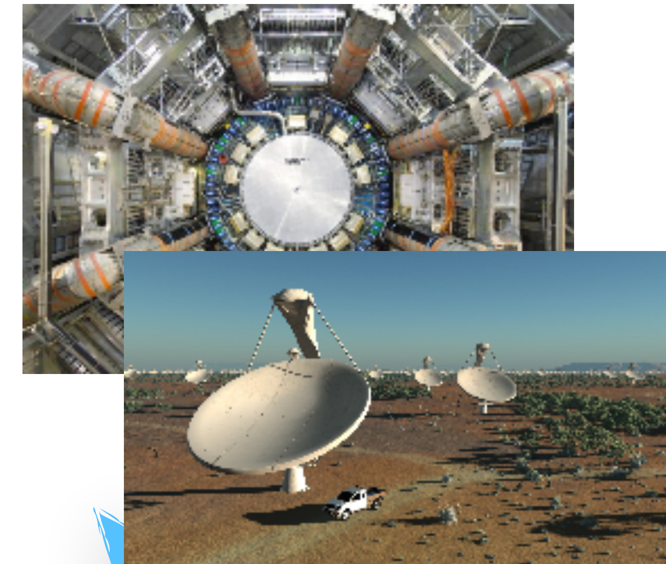
Generalization
Theoretical models





~50 years

Simulations
Computational sciences



Today

Data-driven science

Data-driven science & AI

Is Artificial Intelligence just a **refined, faster** approach to computational science?

Machine Learning

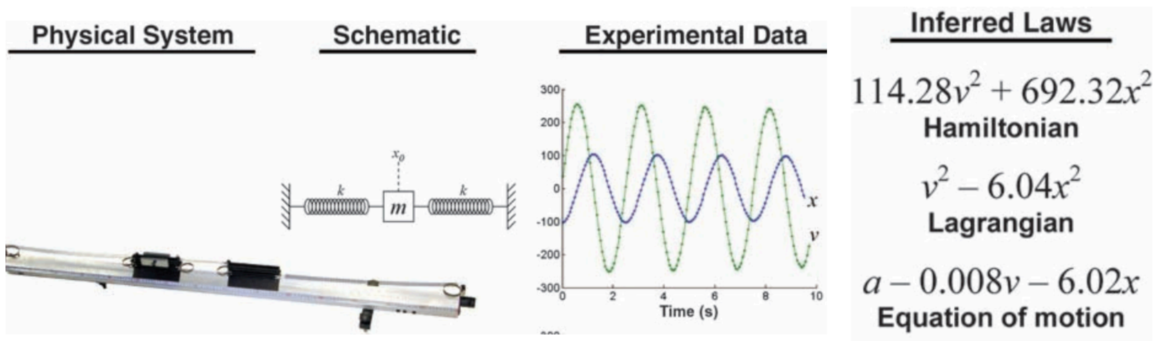
Deep Learning

Artificial Intelligence

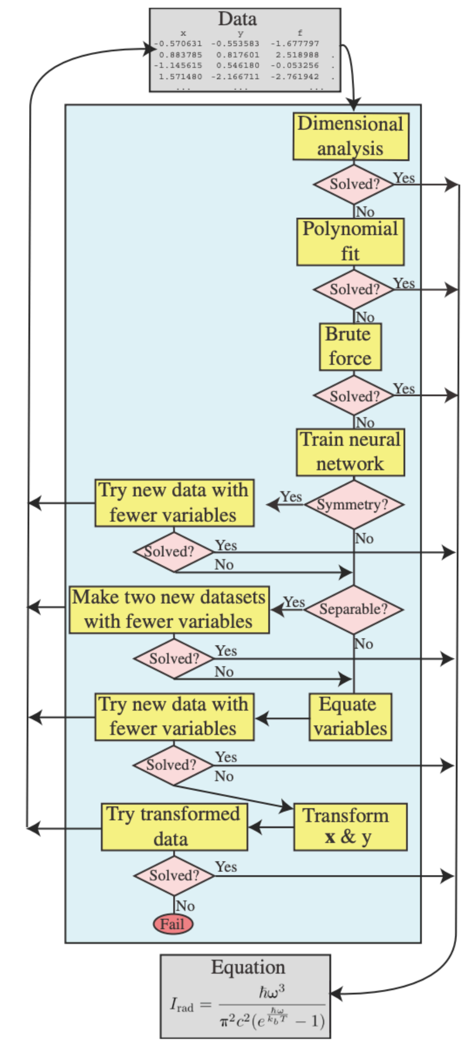
Sofia Vallecorsa, Ilaria Luise CERN - sofia.vallecorsa@cern.ch | ilaria.luise@cern.ch

Rediscovering physics

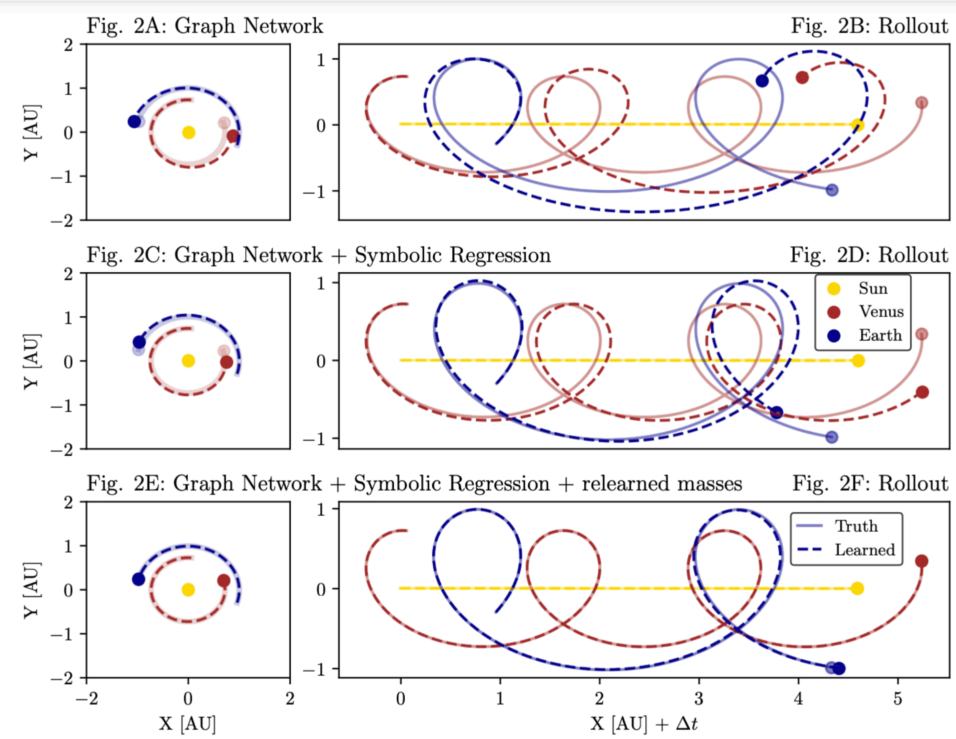
Schmidt, Michael, and Hod Lipson. "**Distilling free-form natural laws from experimental data.**" *science* 324.5923 (2009): 81-85.



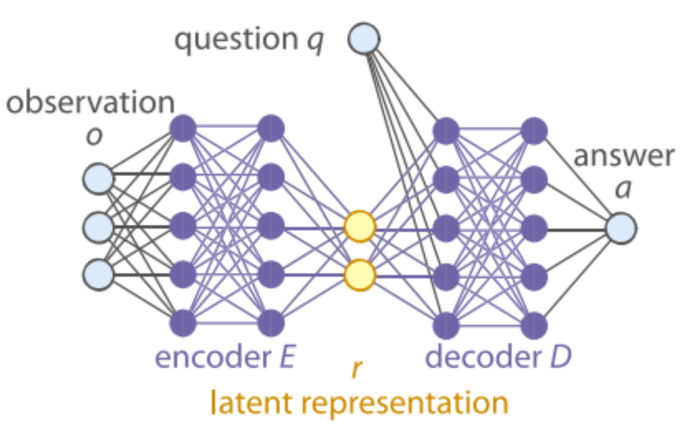
Udrescu, Silviu-Marian, and Max Tegmark. "**AI Feynman: A physics-inspired method for symbolic regression.**" *Science Advances* 6.16 (2020): eaay2631.

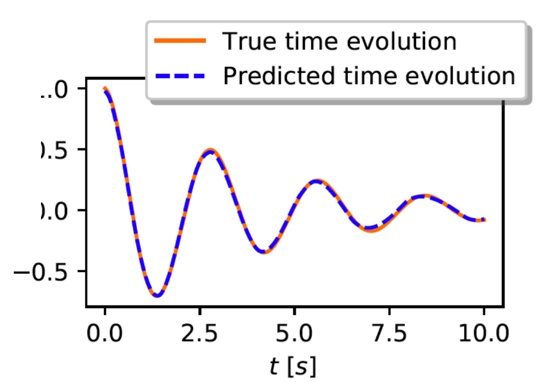


Lemos, Pablo, et al. "**Rediscovering orbital mechanics with machine learning.**" *arXiv:2202.02306* (2022)



Iten, Raban, et al. "**Discovering physical concepts with neural networks.**" *Physical review letters* 124.1 (2020): 010508.





Can we train AI to understand **physics itself** in order to achieve new discoveries ?

Let's start at the beginning

Universal approximator

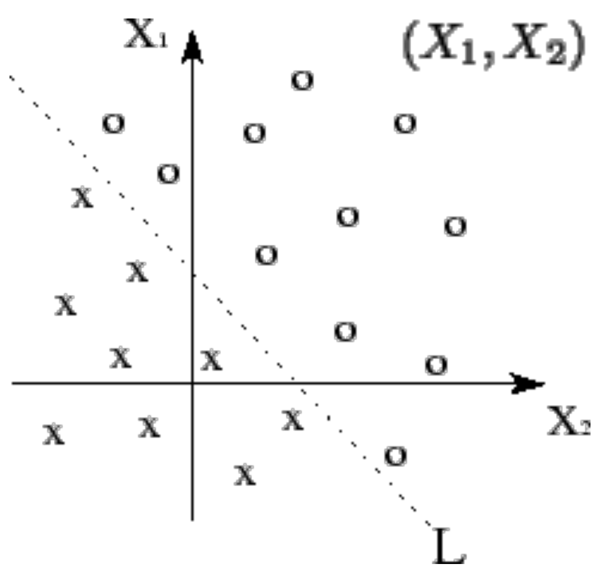
NN with a single hidden layer containing a finite number of non-linear neurons approximate continuous functions to any desired degree of accuracy.

Hornik, Kurt; Tinchcombe, Maxwell; White, Halbert (1989). *Neural Networks*. **2**. Pergamon Press. pp. 359–366.

The need for depth

A single layer perceptron can categorize “linearly separable” patterns (binary classification):

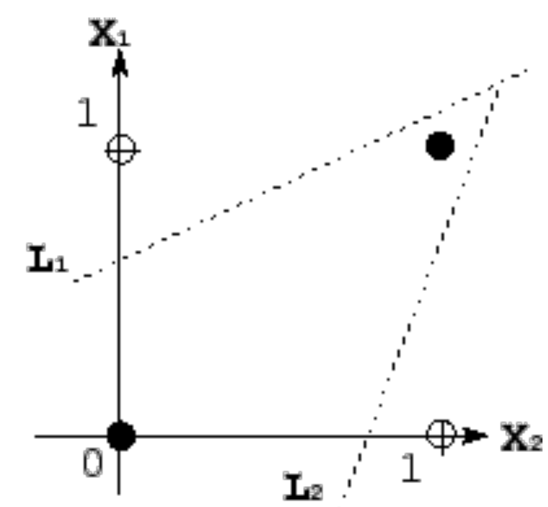
OR function is linearly separable



Exclusive OR is a non linearly separable pattern:

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

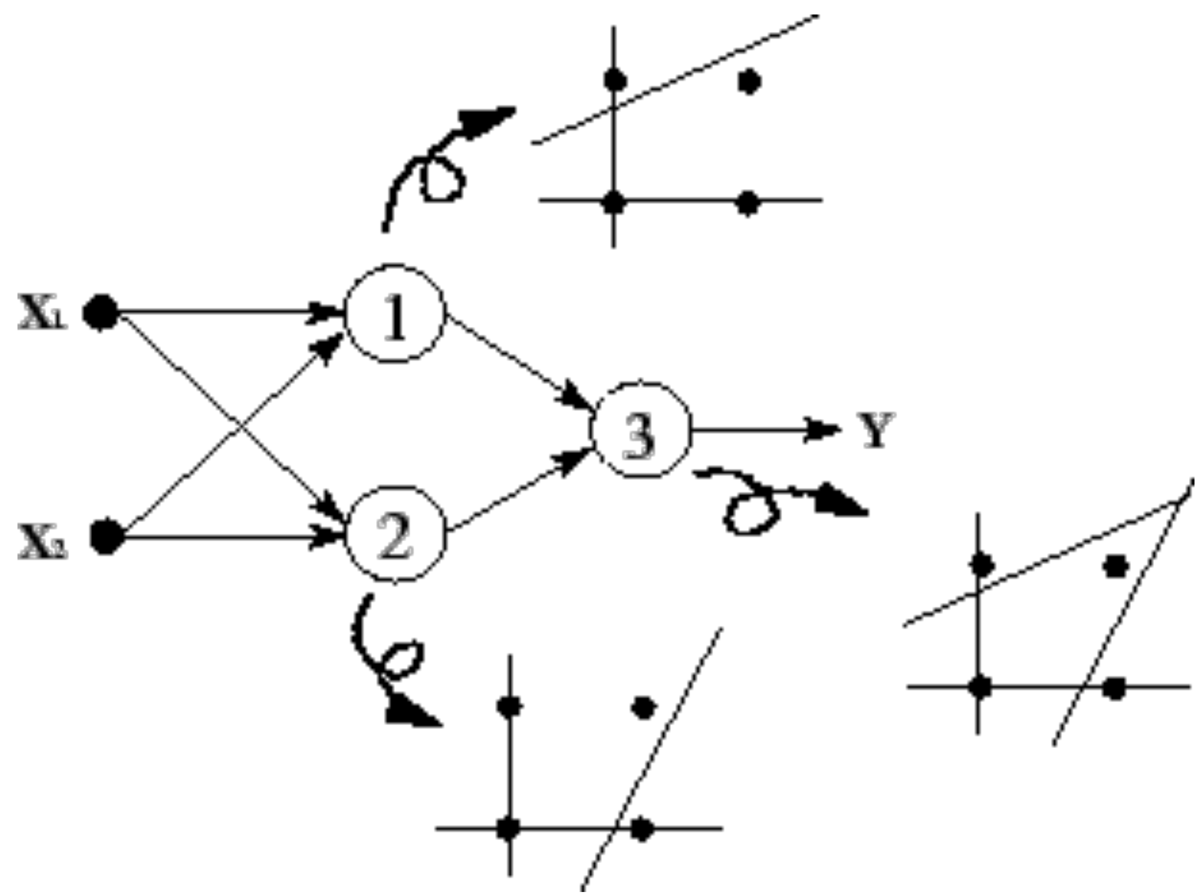
$Y = X_1 \oplus X_2$



(tutorial) http://www.theprojectspot.com/tutorial-post/introduction-to-artificial-neural-networks-part-1/7

The need for depth (II)

Need a Multi-Layer architecture to solve the exclusive OR problem with a two-stages approach

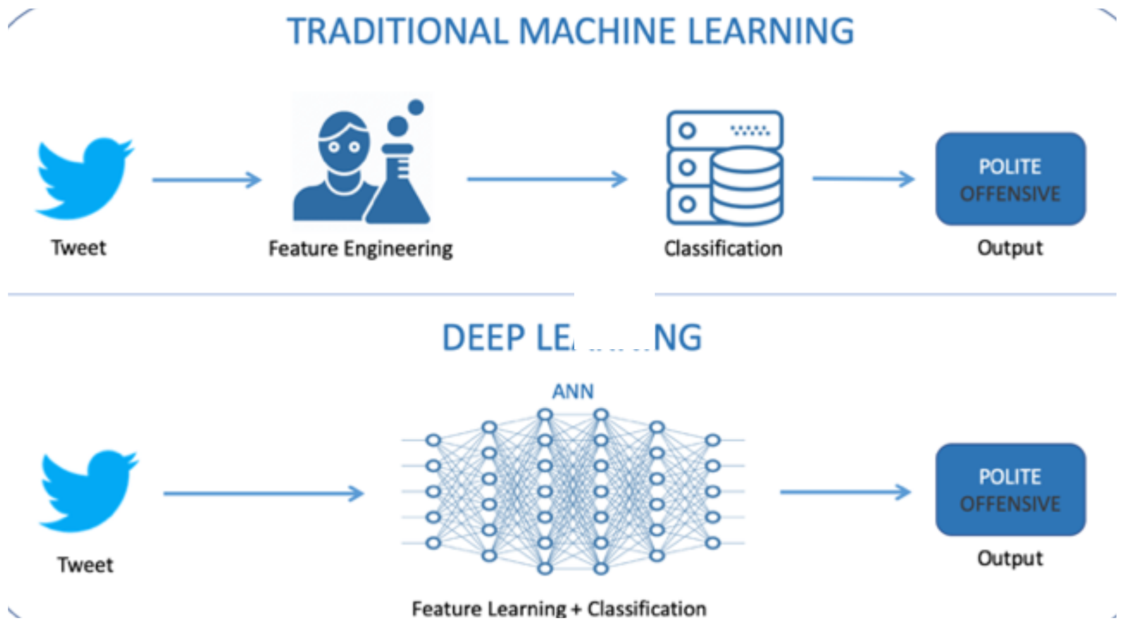


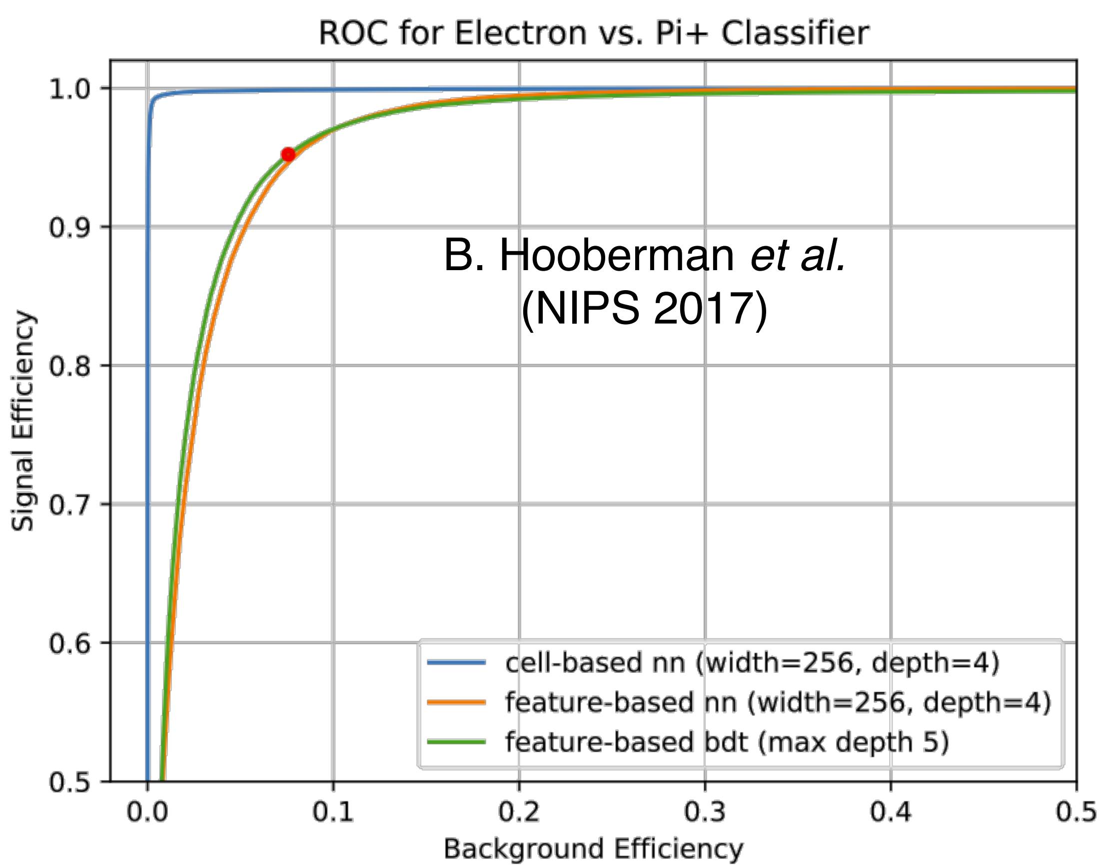
Deep Neural Networks

- "Deep learning allows computational models that are composed of multiple processing layers to learn **representations of data with multiple levels of abstraction**.
- …
- It discovers intricate structures in large data sets by using the back-propagation algorithm to indicate how a machine should change its internal parameters that are used to **compute the representation in each layer from the representation in the previous layer**…"

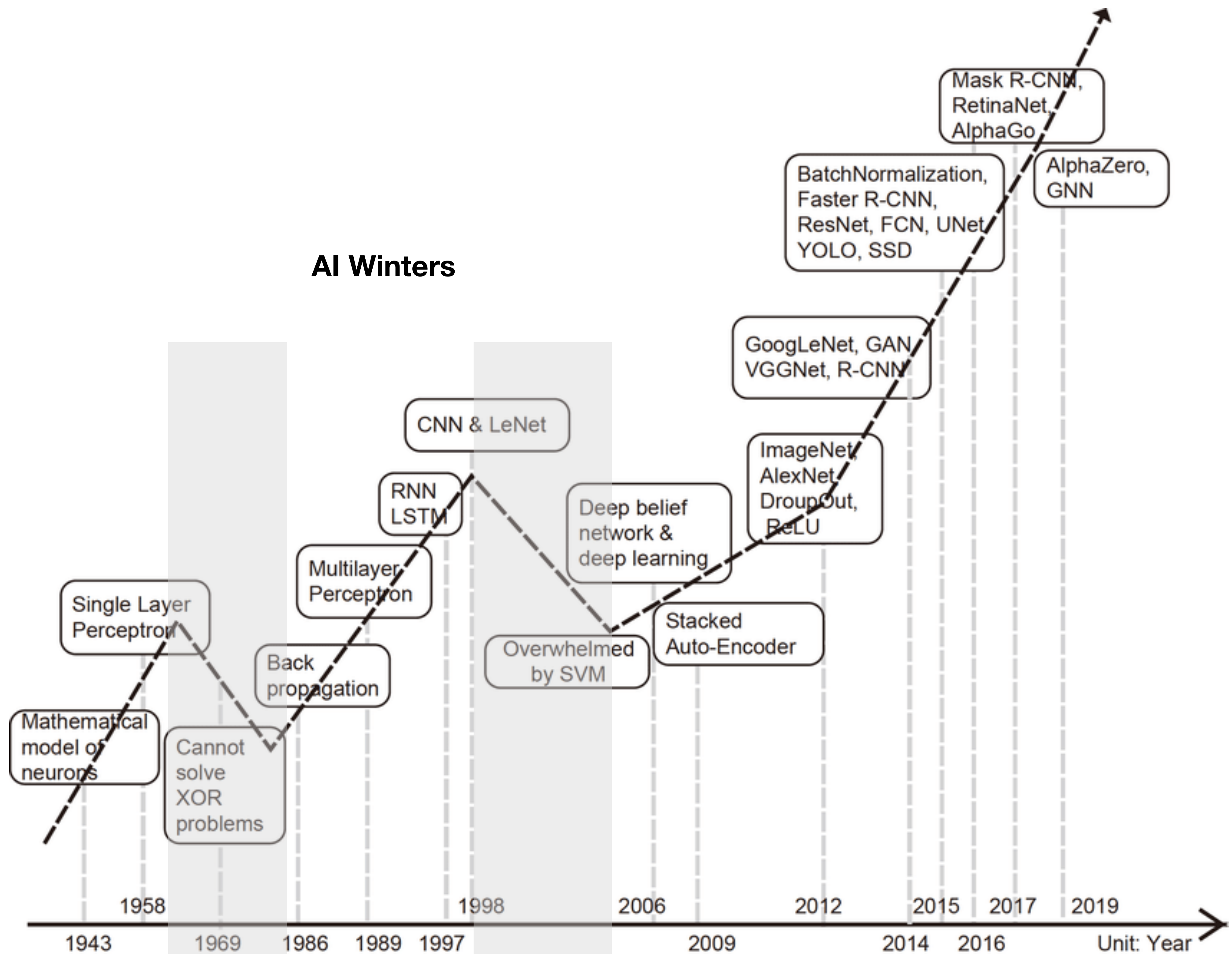
LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444 (2015).

Machine Learning …Deep Learning





A bit of history



Sofia Vallecorsa, Ilaria Luise CERN - sofia.vallecorsa@cern.ch | ilaria.luise@cern.ch

What you should already know…

- Stochastic Gradient Descent and Optimisers
- The problem of vanishing gradients
- Tips and tricks for training and the importance of data
- Regularisation strategies
- Example architectures : Convolutional Neural Networks
-

- Small gradients slow down stochastic gradient descent
  - Limits ability to learn
- Gradients for layers far from the output vanish to zero.

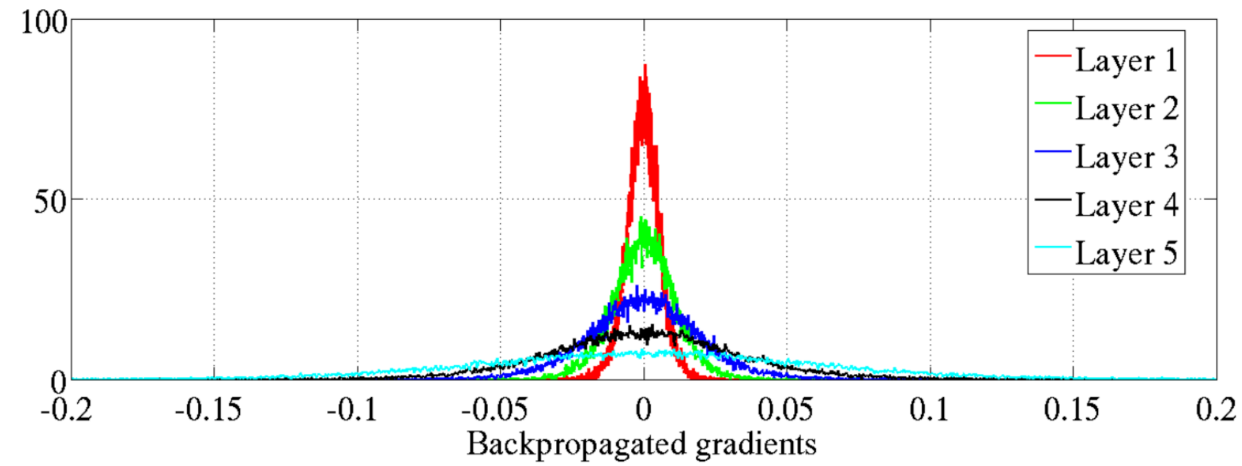
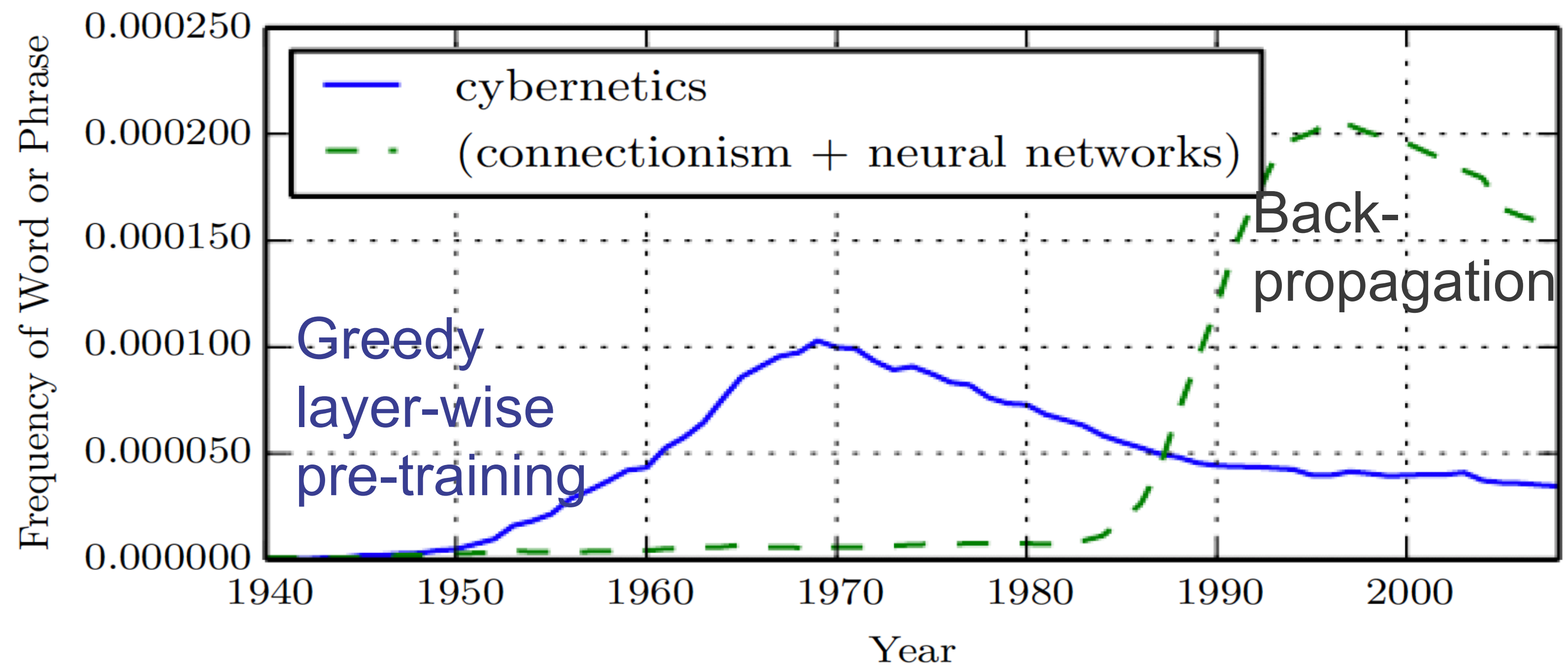


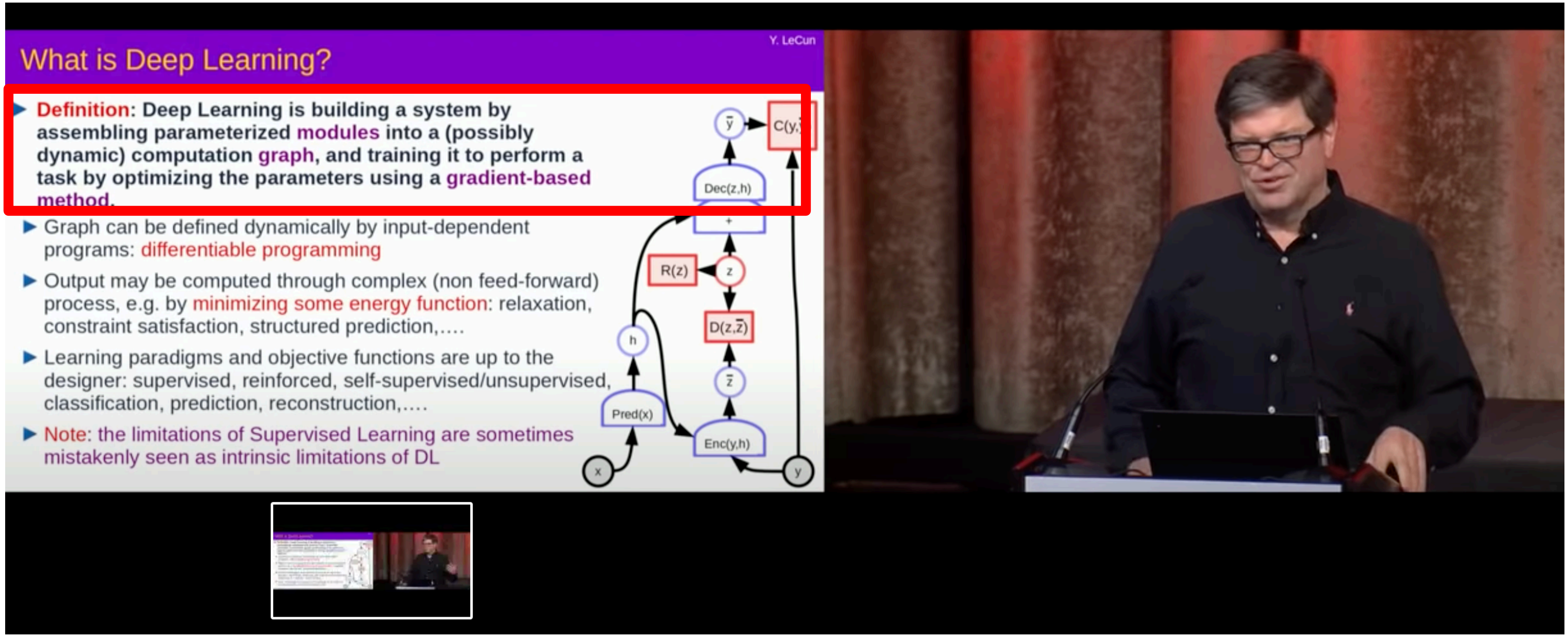
Image from I. GoodFellow, Y. Bengio, A. Courville, "**Deep Learning**"



Sofia Vallecorsa, Ilaria Luise CERN - sofia.vallecorsa@cern.ch | ilaria.luise@cern.ch

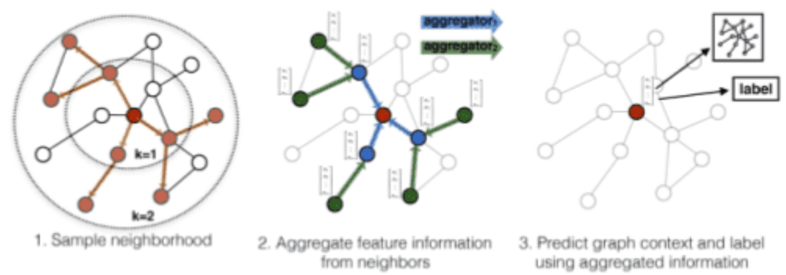
AAAI 2020: Turing Award Keynote

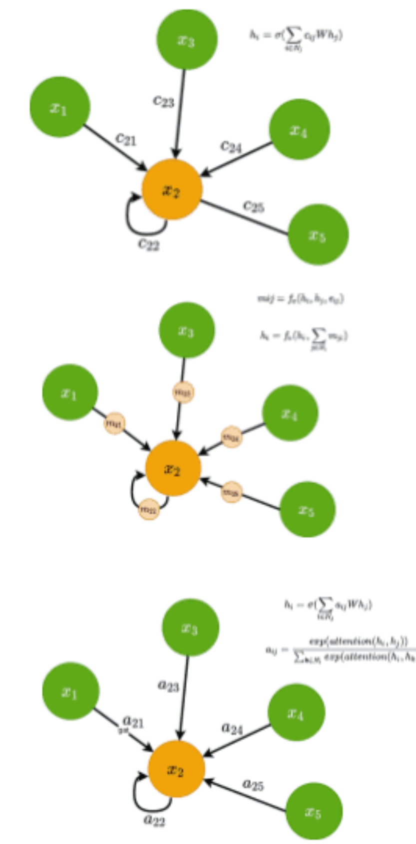
Deep Learning: more than just a deeper NN

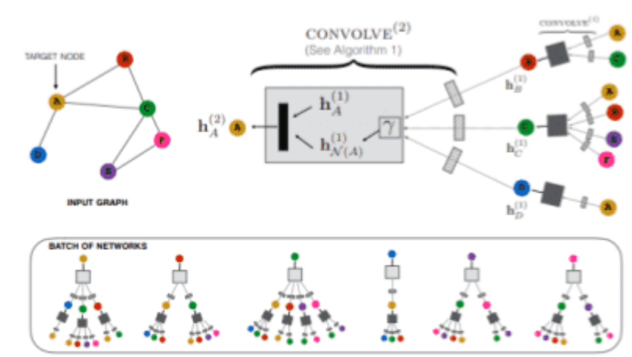


AAAI 20 keynotes Turing Award Winners (Geoff Hinton Yann Le Cunn, Yoshua Bengio): https://www.youtube.com/watch?v=UX8OubxsY8w

Different primitives for different data representations







Perceptrons and MLP

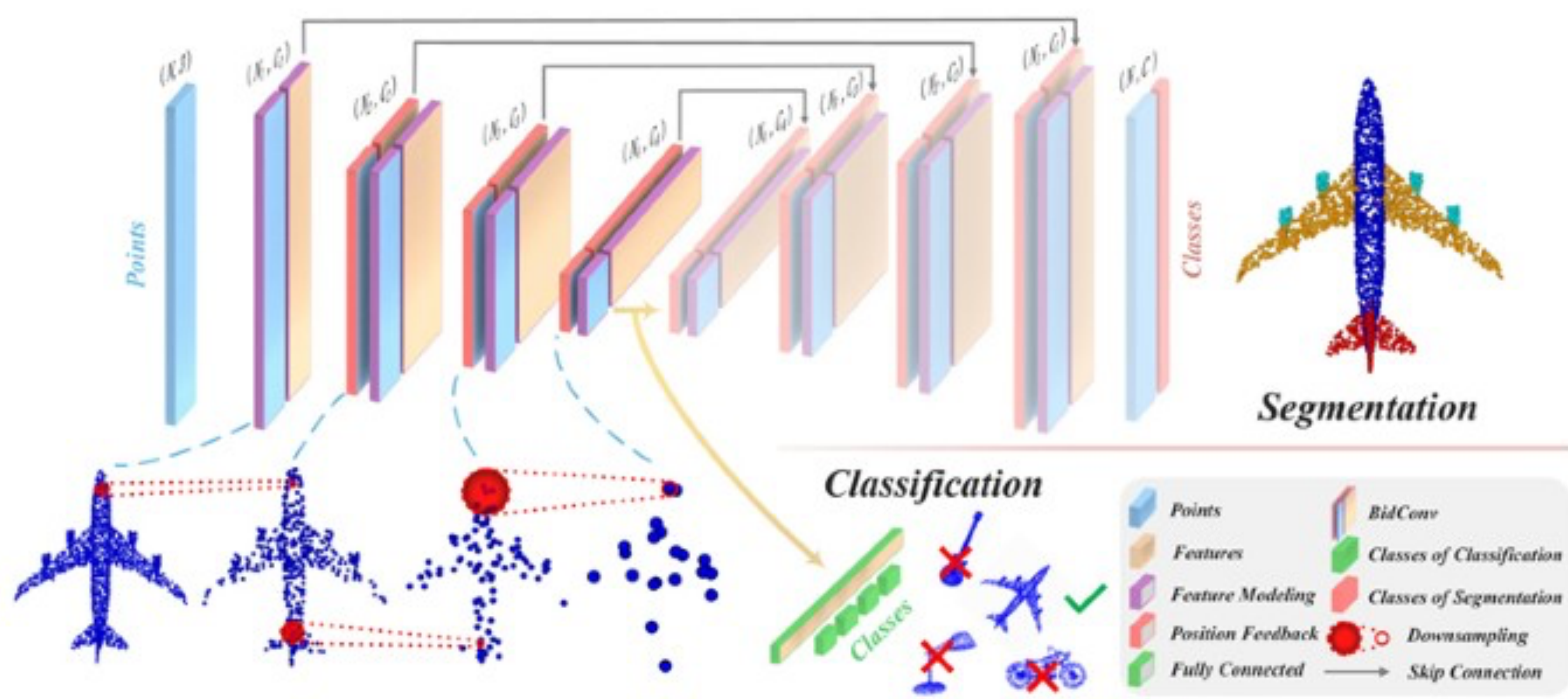
Convolutions

Graphs

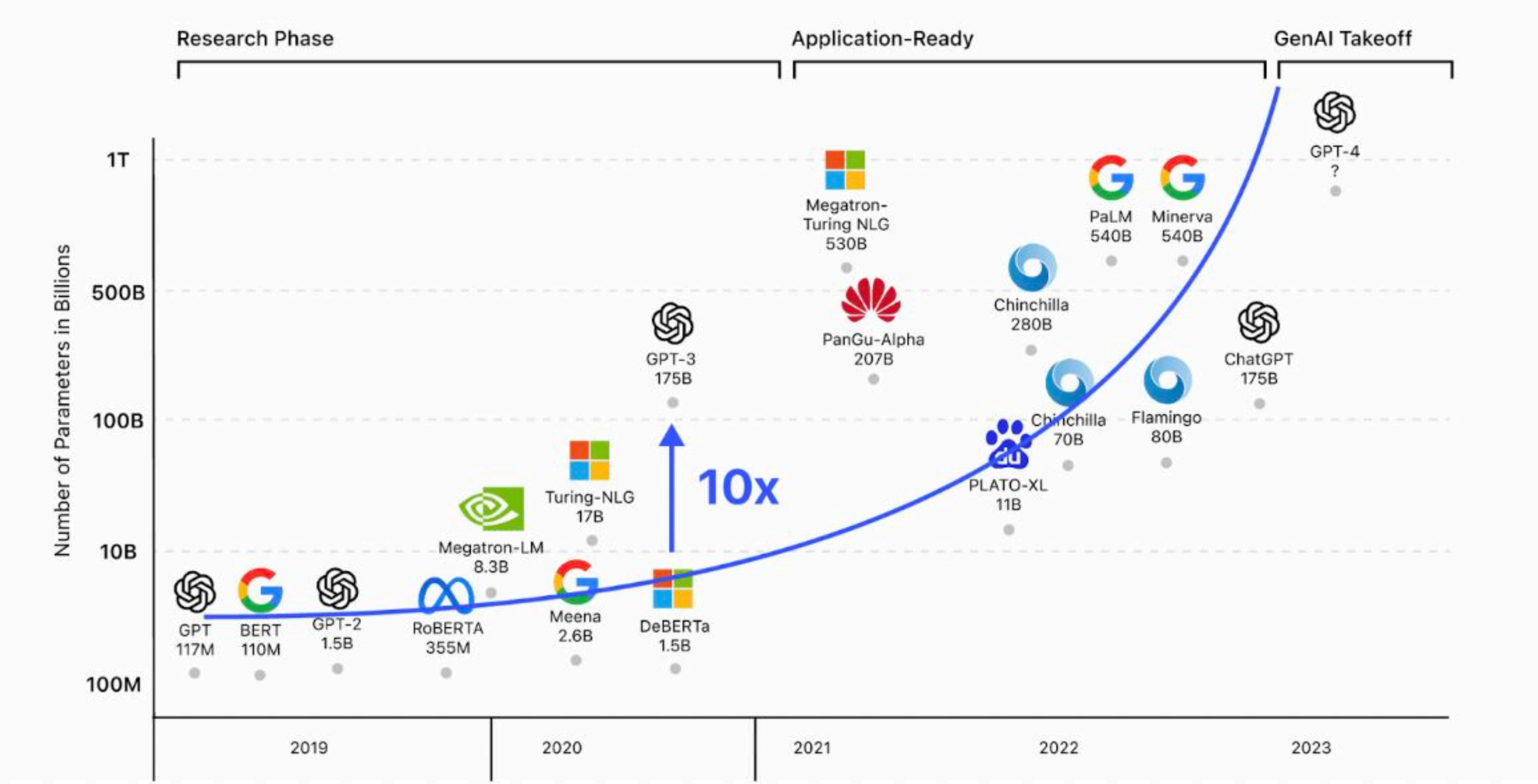
Recurrent Units (and LSTMs)

Point Cloud

…



Then.. AI TakeOff….

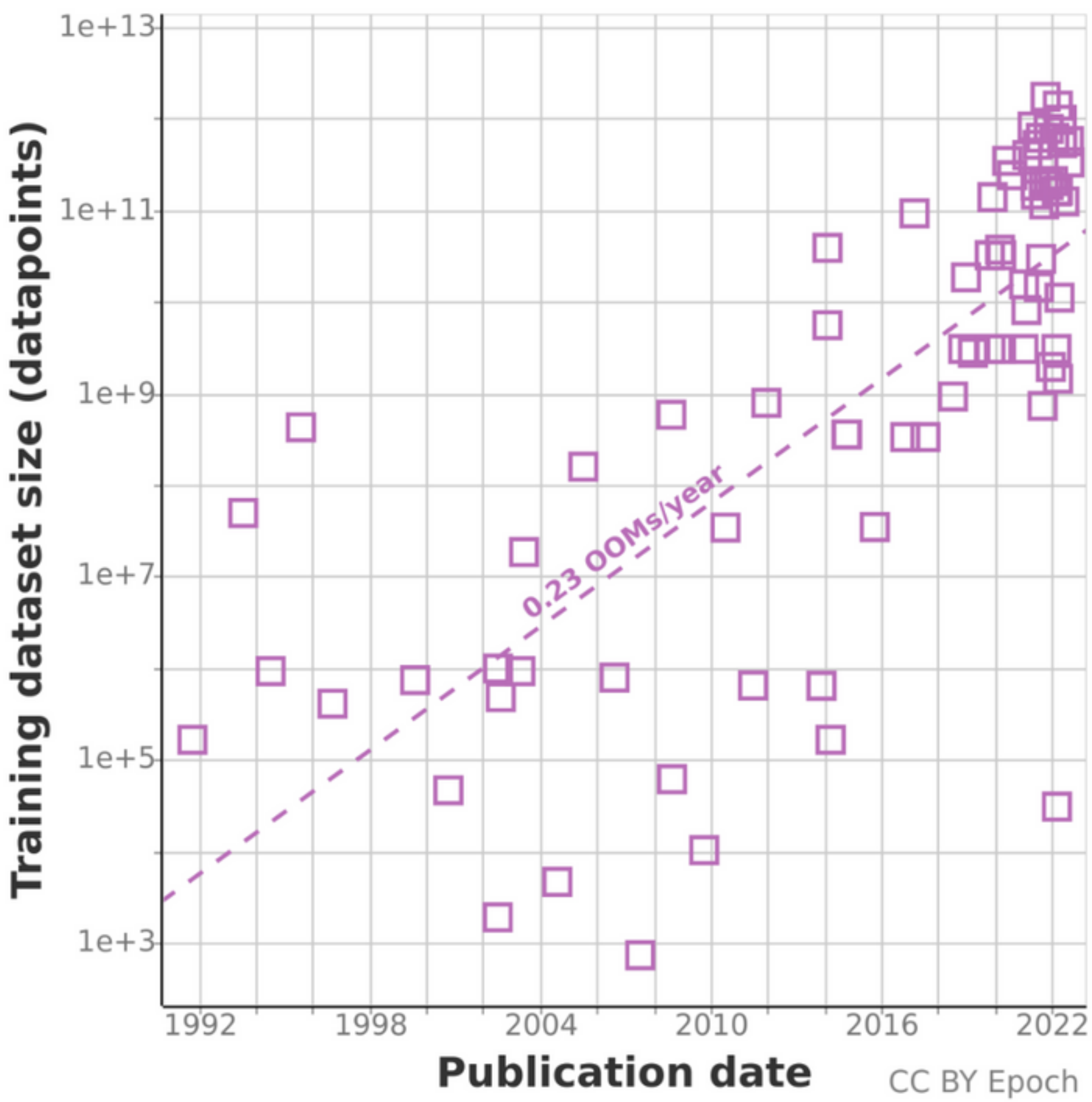


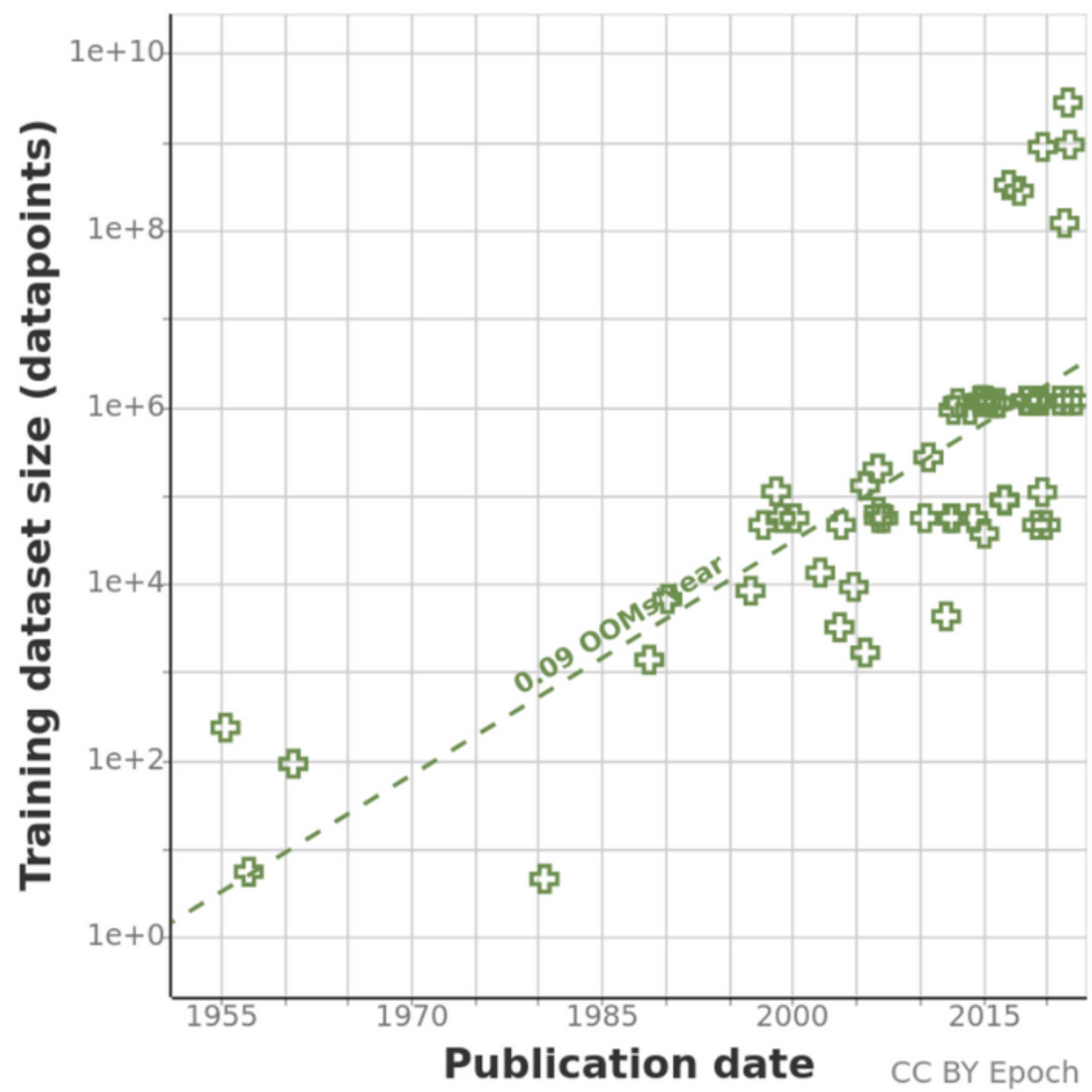
Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Dataset sizes

Domain	Data points
Vision	#Images (eg: a model trained on 3B images has a dataset size of 3B)
Language	#Words (eg: a model trained on 1T English tokens has a dataset size of ~750B words, the exact quantity depends on the tokenization)

Training datasets for language (left) and vision (right)





https://epochai.org/blog/trends-in-training-dataset-sizes

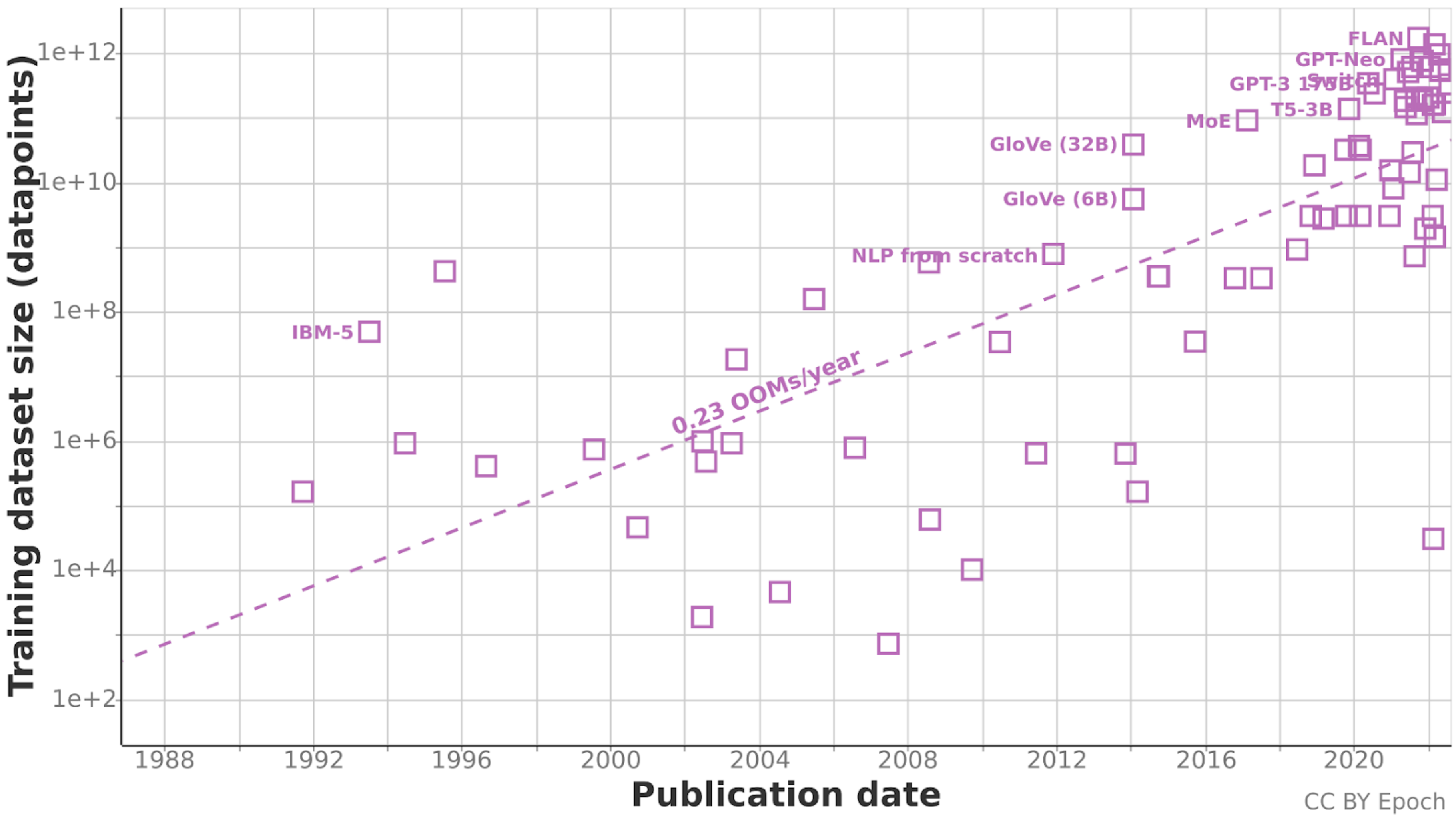
Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Machine learning at scale, for science

Machine learning has been proven a very good tool to:

- Extract information from (very large) datasets
- Efficiently analyse very large amounts of data
- Easily handle data from different sources
- Scalability to HPC environments

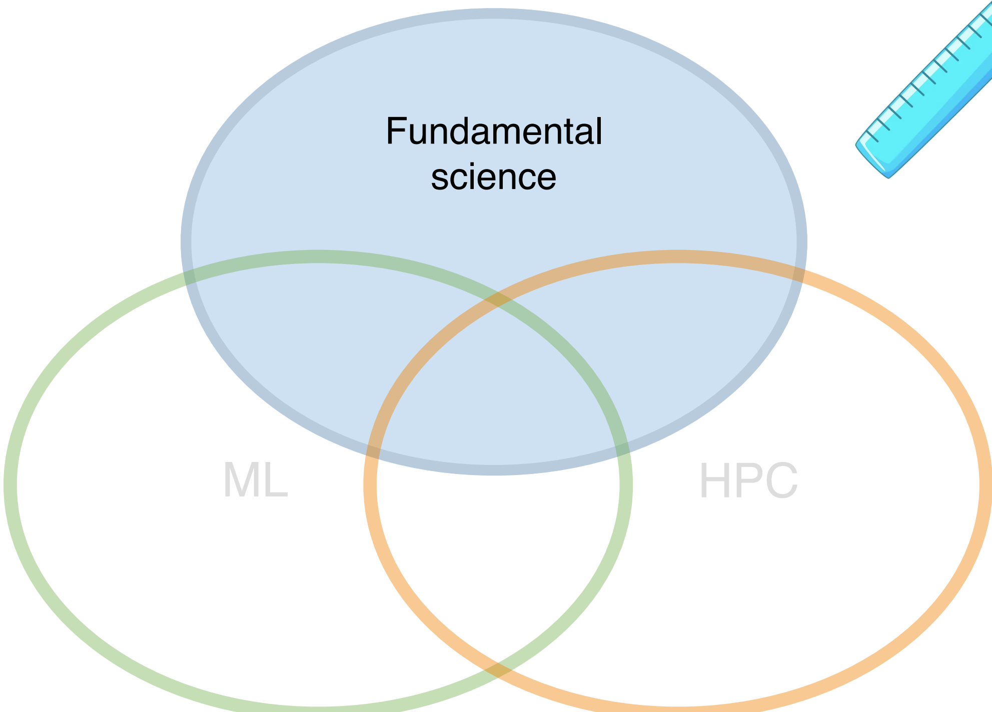
Observation based datasets in physics are comparable or larger than these!



Can we use these tools for fully data-driven science?

Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Scientific opportunities



Multi-scale dependencies:

- **Model complex higher-order, statistical relationships between observations, fields, …**
- improve current simulations

Compact representations:

- **Condense dataset information in a compact representation**
- eg. condense the info in a few GB rather than TB

Multi-source models:

- **Enable multimodal and multi-source learning**
- eg. build models based on scientific data, GDP, birth rate etc..

New discoveries:

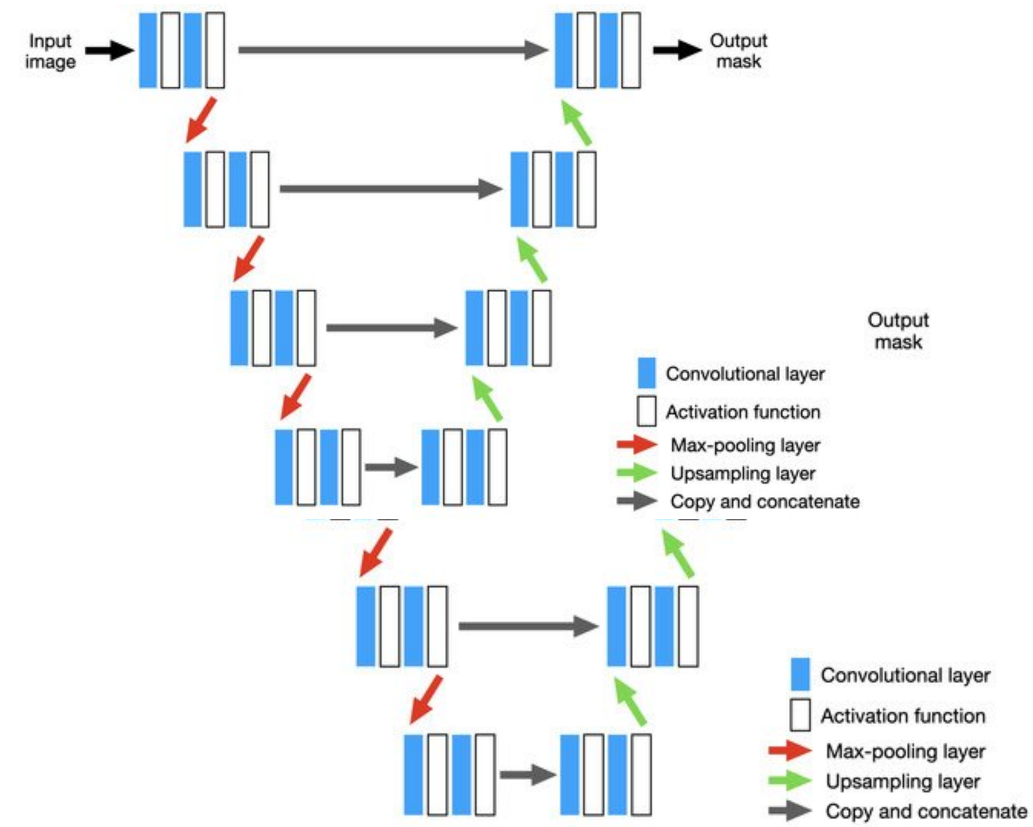
- **Explore the potential of unsupervised learning to extract new information directly from data**
- Learn unknown correlation patterns

Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Transfer learning, pre-training, fine-tuning

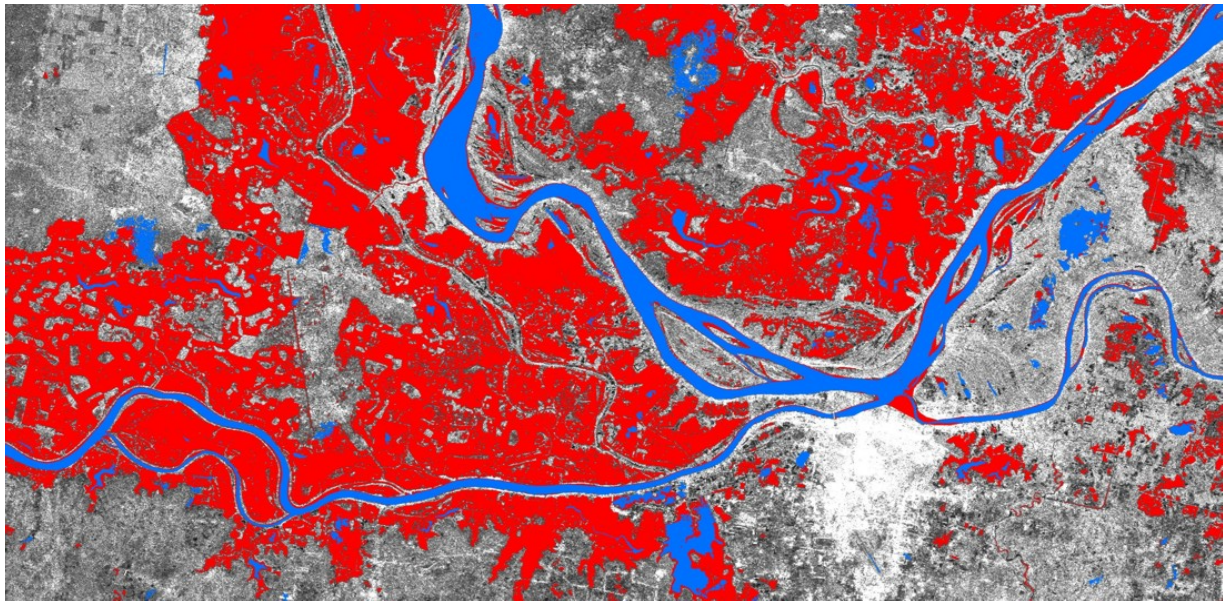
*"**Transfer learning** and **domain adaptation** refer to the situation where what has been learned in one setting … is exploited to improve generalization in another setting"*

Deep Learning, 2016.



- Transferring knowledge to similar task
- Can be used to train large models
- Keep/modify pre-trained model?
  - CNN features are more generic in early layers and more dataset-specific in later layers
- Ex. **Flood detection in satellite images using U-Net**
- The **pre-training/fine-tuning** strategy has become key to the development of foundation models

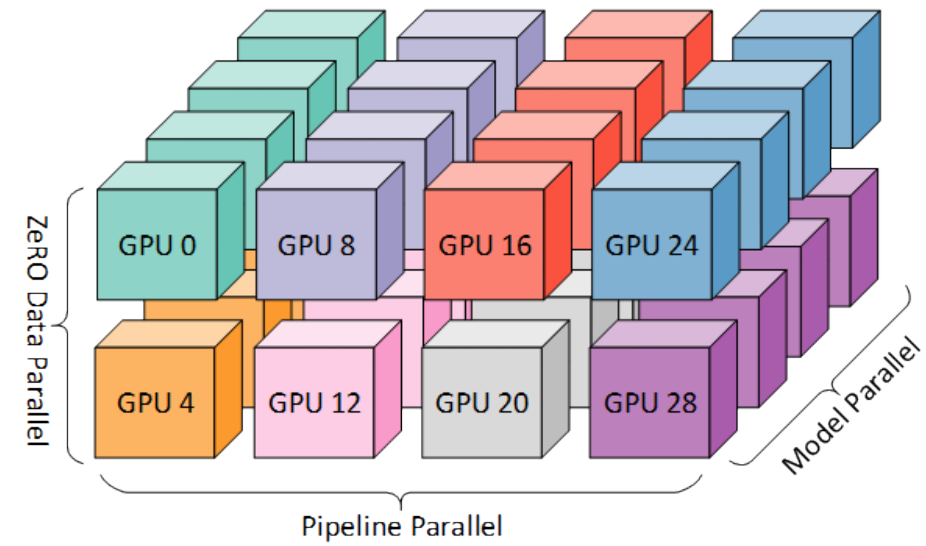
(More on this later)



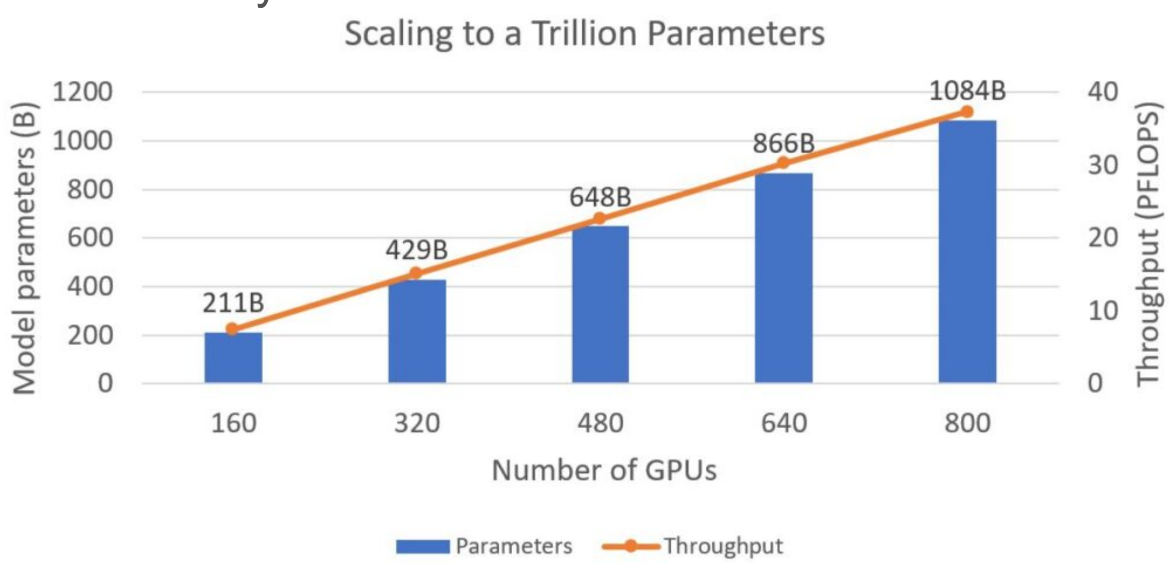
Nemni, Edoardo, et al., *Remote Sensing* 12.16 (2020): 2532.

Accelerating training

- **Data parallelism**
  - Compute gradients on several batches independently
  - Update the model synchronously or asynchronously
- **Model Parallelism**, **Hybrid techniques**
- **Reduced precision** (INT8, BF16, …)
- Extreme parallelism using **combined strategies** and SGD algorithm optimization. Ex.
  - DeepSpeed and ZeRO-2 on Microsoft Azure



https://www.microsoft.com/en-us/research/blog/deepspeed-extreme-scale-model-training-for-everyone/

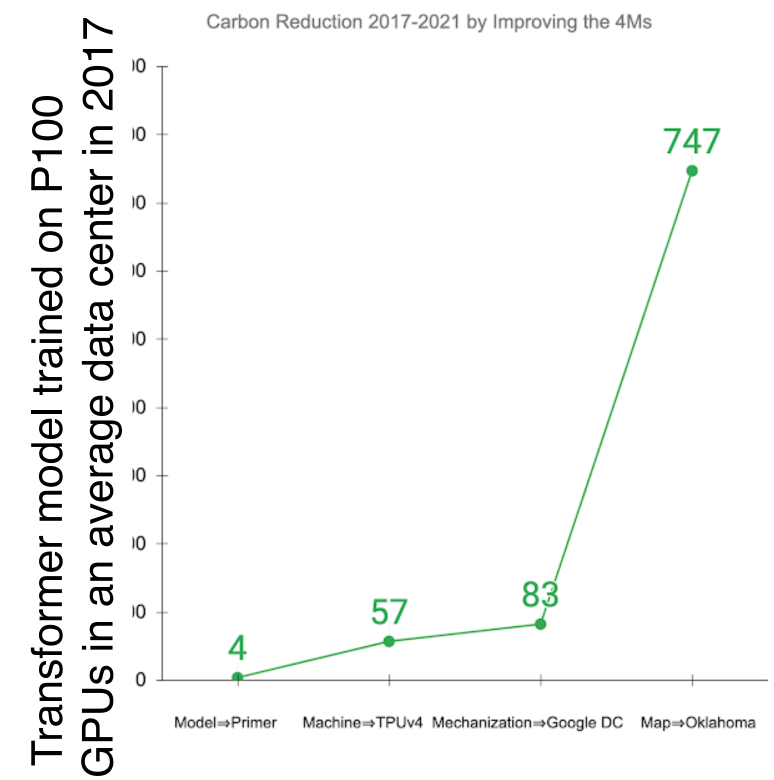


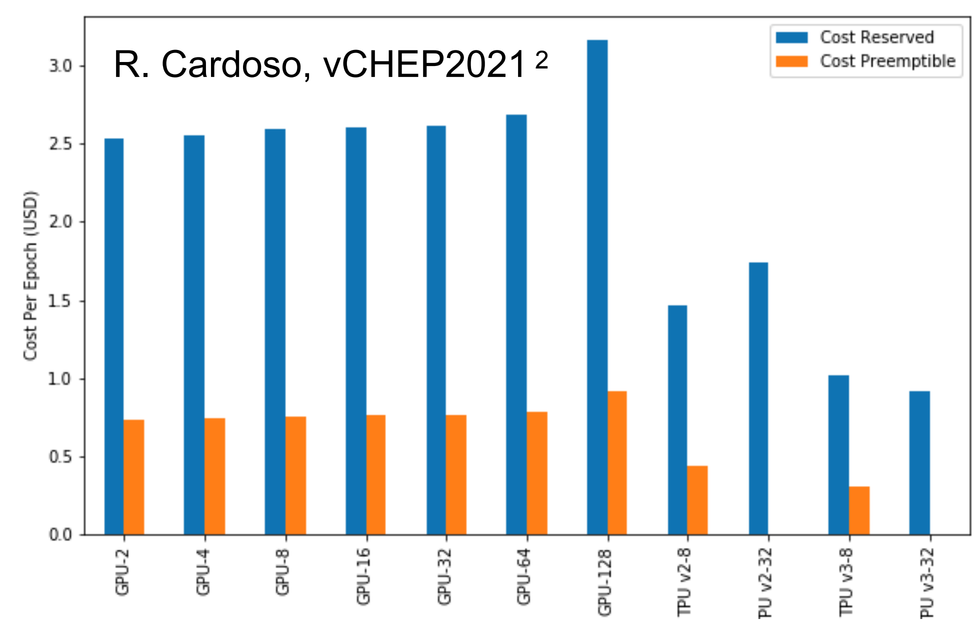
Sofia Vallecorsa, Ilaria Luise CERN - sofia.vallecorsa@cern.ch | ilaria.luise@cern.ch

Sustainable AI

- AI inference more **energy efficient** than classical algorithms
- Energy cost of **AI training** can be high
- The community is defining **best practices**[1]
    - **Efficient AI architectures** can reduce computation by 3x–10x.
    - **AI-optimized processors vs general-purpose** can improve energy efficiency by 2x–5x[2].
    - **Cloud computing vs on-prem** reduces energy usage by 1.4x–2x
- **Efficient training strategies**
    - Self-supervision, few-short learning, pre-training

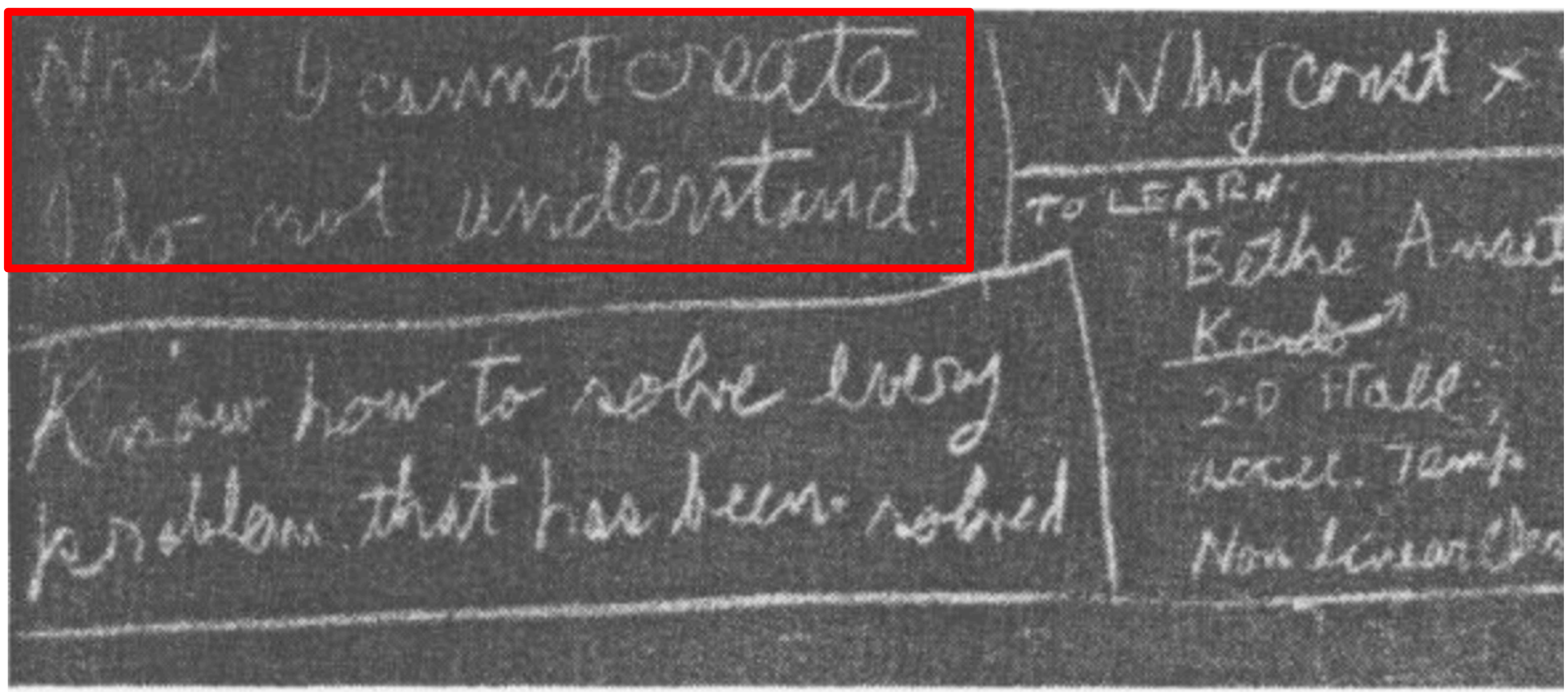
https://ai.googleblog.com/2022/02/good-news-about-carbon-footprint-of.html





2 Cardoso, Renato, et al. "Accelerating GAN training using highly parallel hardware on public cloud." EPJ Web of Conferences. Vol. 251. EDP Sciences, 2021.

Generative Models



R. Feynman

Generative models

The problem:

Assume data sample follows p_{data} distribution

Can we draw samples from distribution p_{model} such that $p_{model} \approx p_{data}$?

31

Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Generative models

The problem:

Assume data sample follows p_{data} distribution

Can we draw samples from distribution p_{model} such that $p_{model} \approx p_{data}$?

Maximum Likelihood Estimator:

- Assume some form for p_{model} (prior knowledge, parameterized by θ)
- draw samples from $p_{\theta *}$

$$\theta^* = \arg\max_{\theta} \sum_{\mathbf{x} \in \mathcal{D}} \log(p_{\text{model}}(\mathbf{x}; \theta))$$

Generative models don't look for mathematical expression of p_{model}

Train NN as a generator $g : \mathbb{R}^m \rightarrow \mathbb{R}^n$

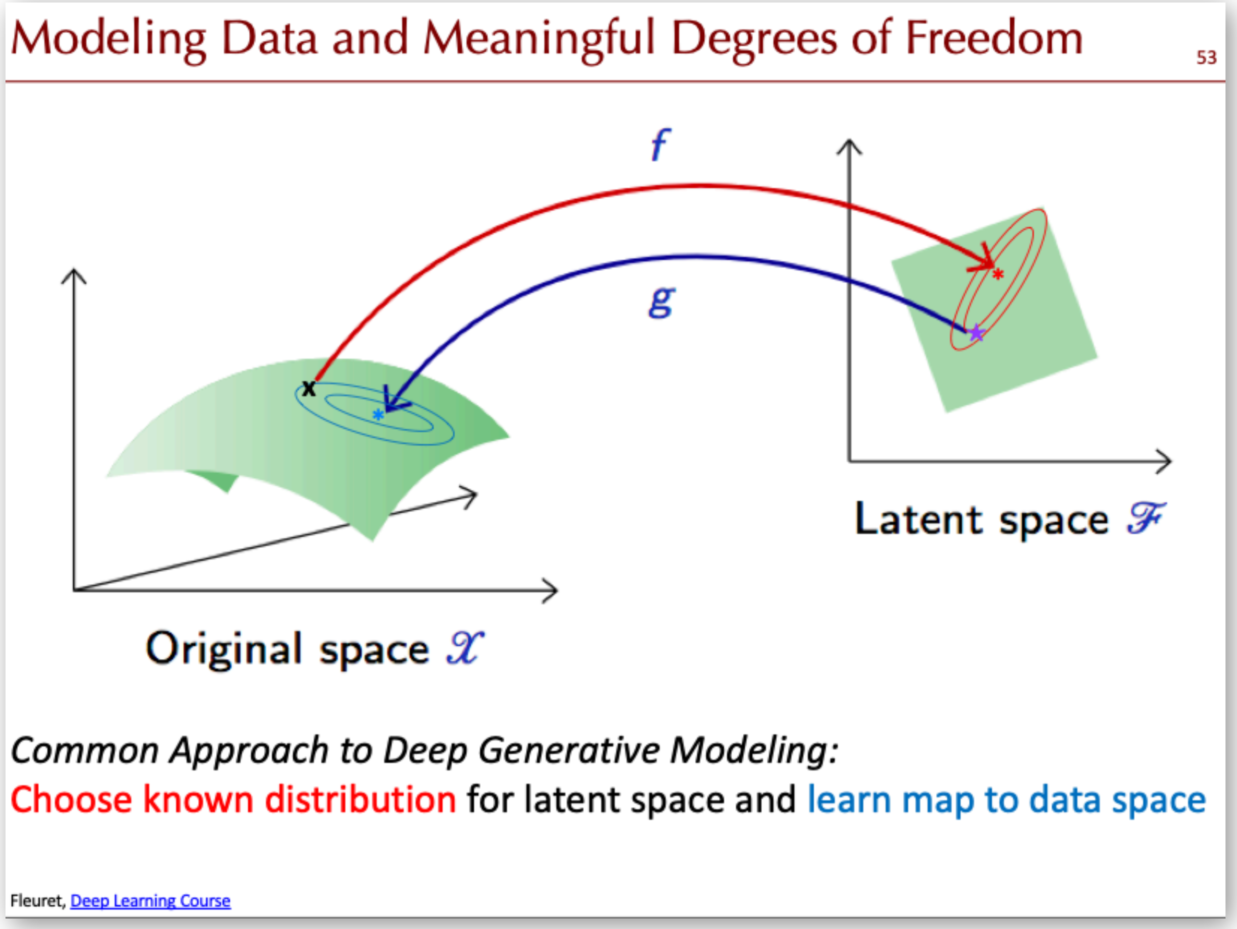
that maps samples from a tractable distribution supported in $\mathbb{R}^m$ to points in $\mathbb{R}^n$

31

Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Latent Representation

See M. Kagan lecture: https://indico.cern.ch/event/1392500/

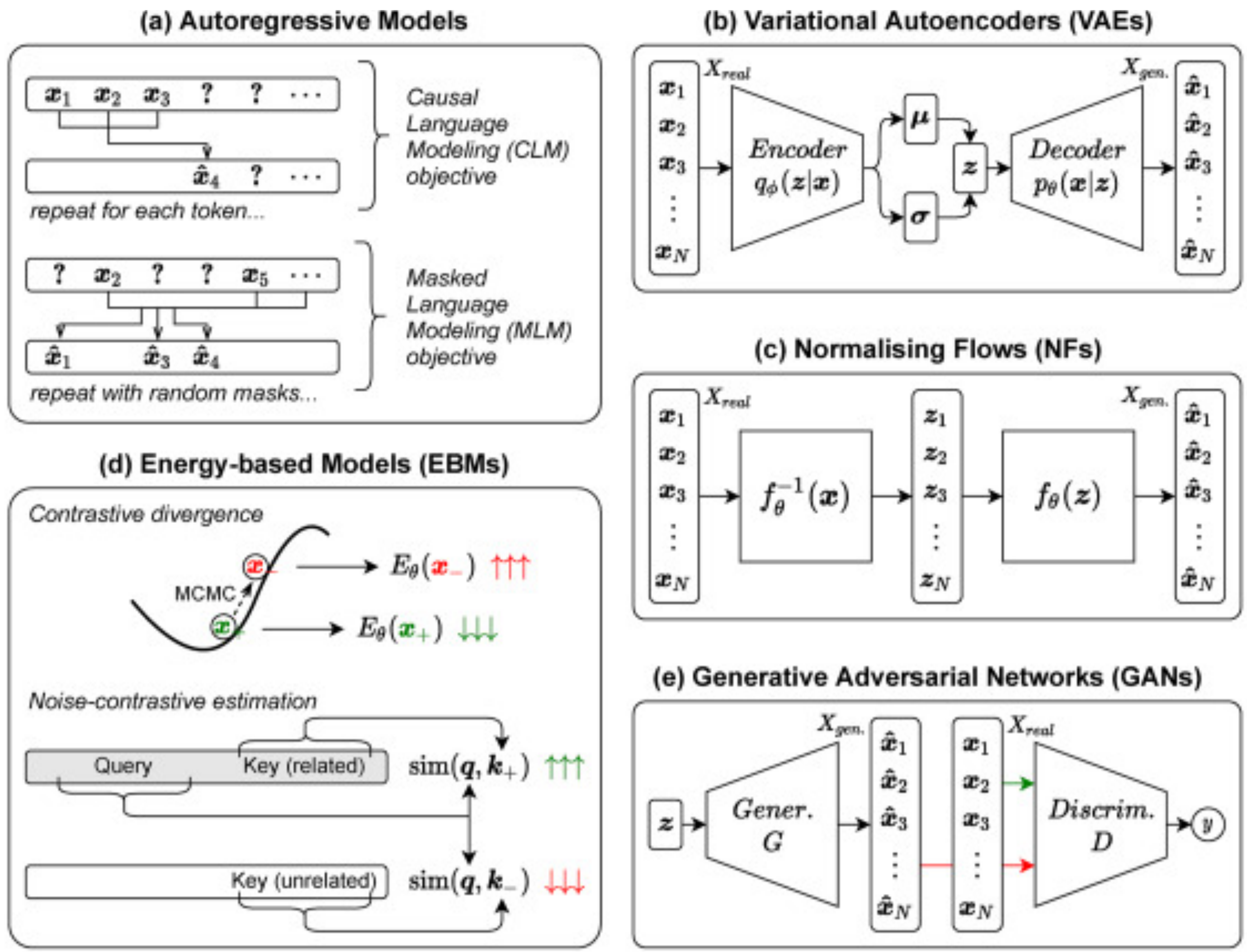


- Information content is preserved within a **hidden manifold with lower dimension**
- Can manipulate **latent space** (style specification, hypothesis testing directly in data, …)
- Can optimise latent representation according to a specific task (**guided compression**)
- Can help with **multi-modality**

NB: Problems exhibiting complex symmetries may benefit from latent space representations connected to the specific underlying symmetry group!

Deep Generative Models

Deep models allow **higher levels of abstractions** and **improve generalization** wrt to shallow models



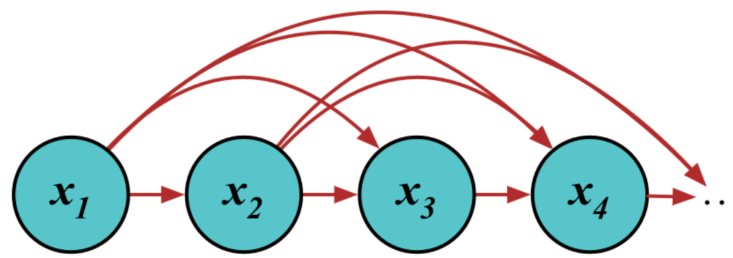
Current Opinion in Structural Biology

See Danilo Rezende tutorial on Deep Generative Models

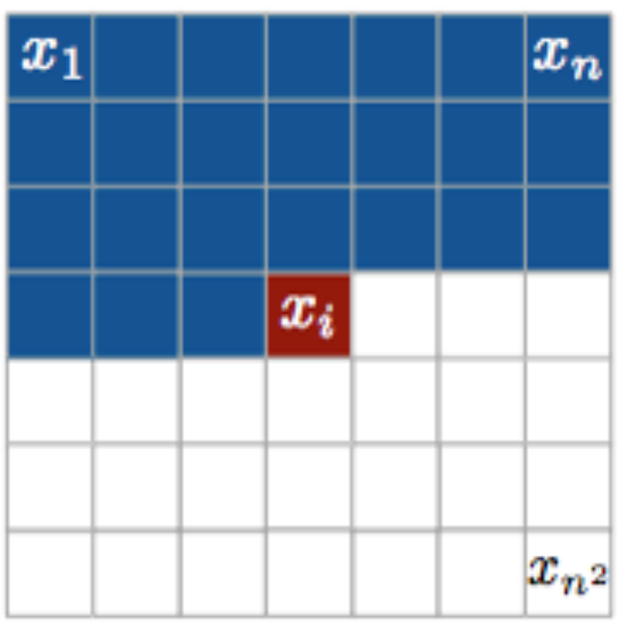
Fully Observed Models

Directly observe data without introducing new local (latent) variables

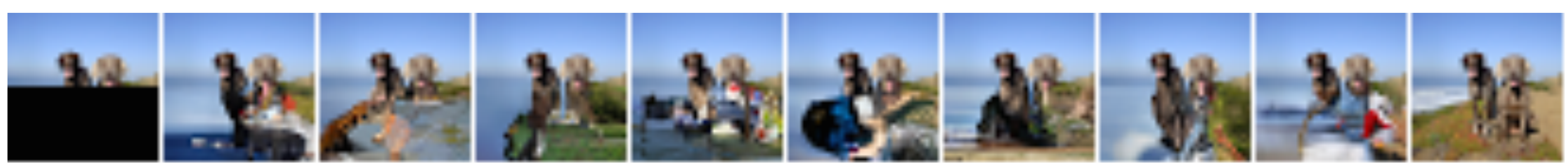
$$p(x_{1,\ldots,N}) = \prod_{i=1}^{N} p(x_i | x_{1,\ldots,(i-1)})$$



Ex. Pixel Recurrent Neural Networks:



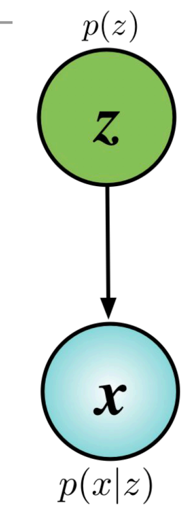
$$p(x_t \mid x_{1:t-1}) = p(x_t^{red} \mid x_{1:t-1}) p(x_t^{green} \mid x_{1:t-1}, x_t^{red}) p(x_t^{blue} \mid x_{1:t-1}, x_t^{red}, x_t^{green})$$



https://arxiv.org/pdf/1601.0675

Auto-Encoders

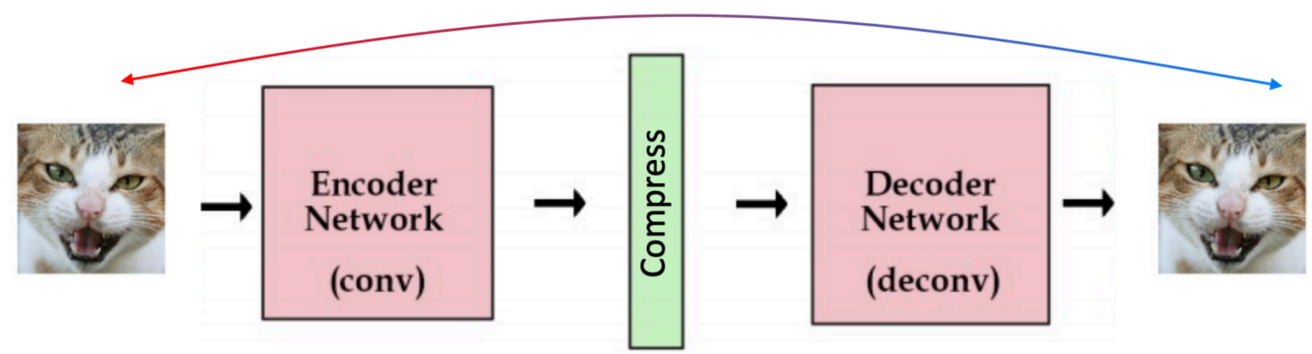
Examples of latent variables models (and implicit..)



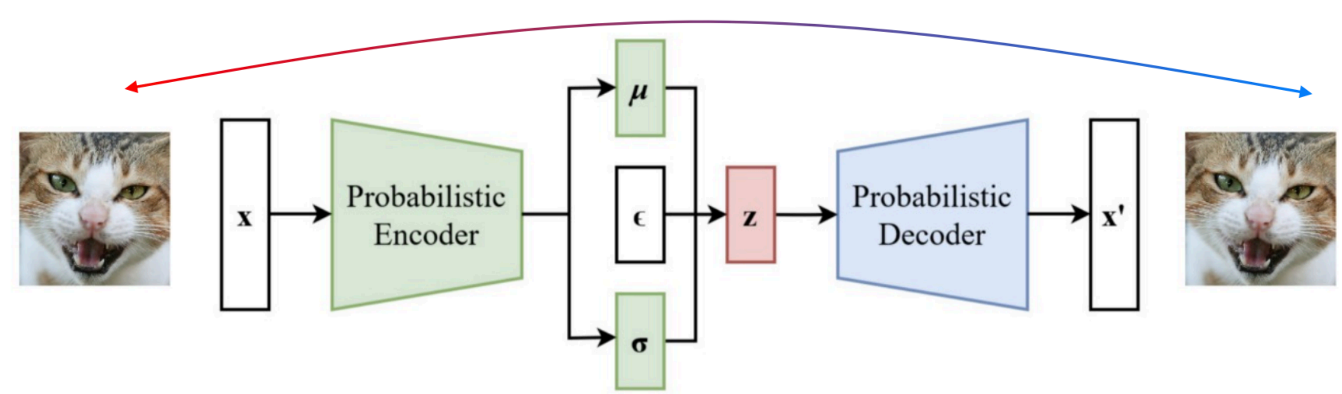
$x \in \mathbb{R}^{d_x} \quad z \in \mathbb{R}^{d_z} \quad \theta \in \mathbb{R}^{d_\theta}$

$\mathcal{D} = \{x_i\} \quad i \in \{1, \ldots, N\}$

Ex. Auto-Encoder



Ex. Variational Auto-Encoder



Explicit constraints on encoded representations (learn the **latent variable distribution**)

Two components in the loss function (**reconstruction loss and KL divergence** to constrain latent to prior)

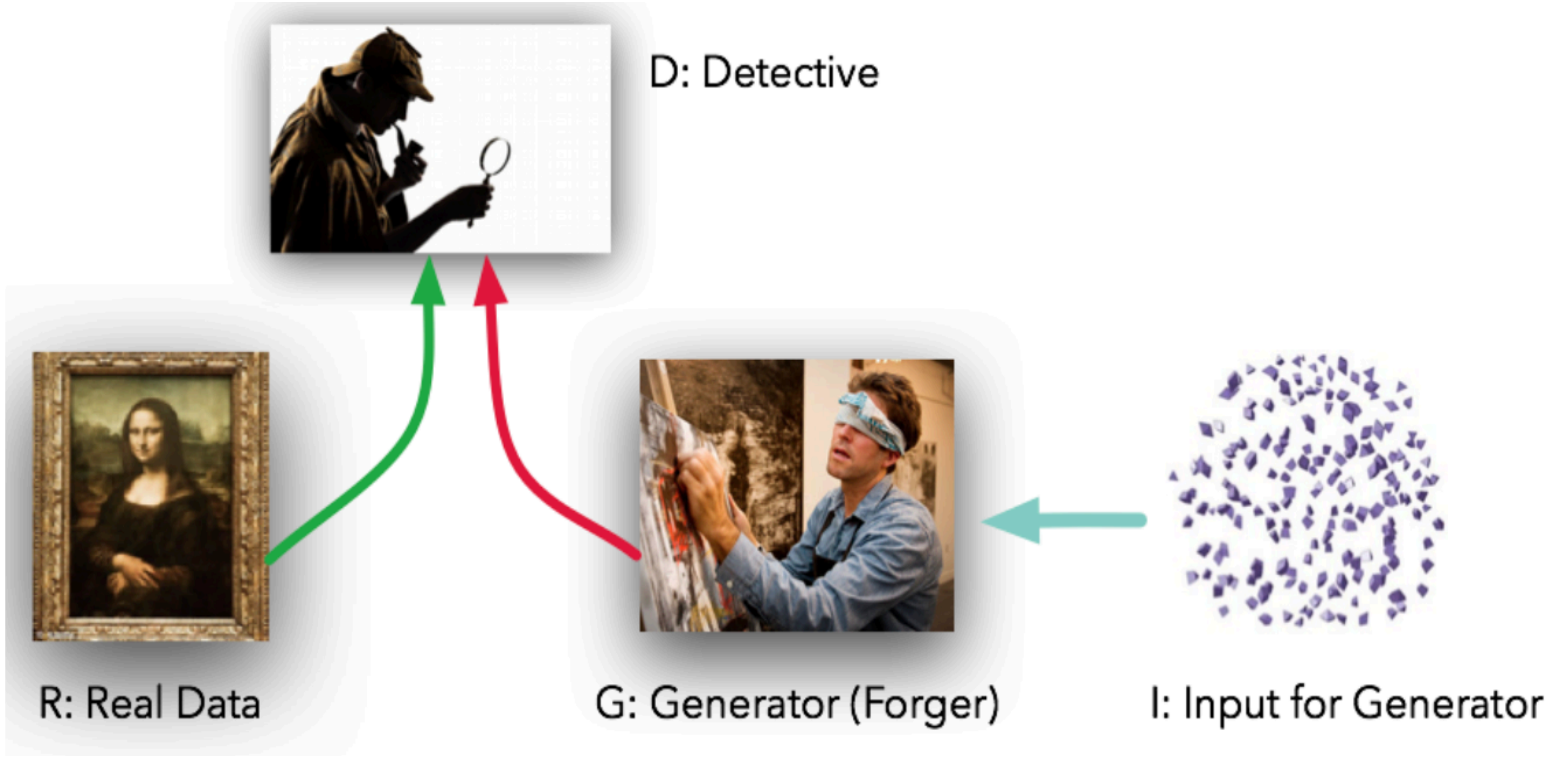
Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Likelihood-free learning

Density estimation by comparison

Sample-based comparison between **estimated** q(x) and **true distribution** p(x)

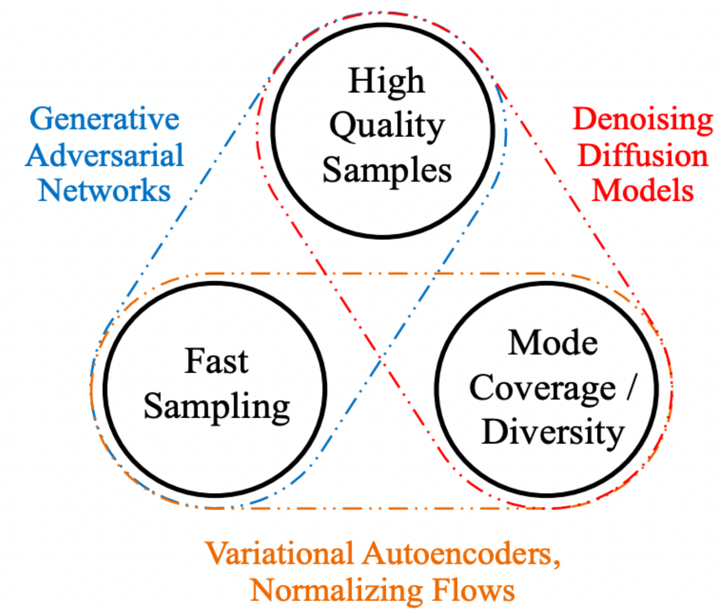
- Build **auxiliary model** to indicate how data simulated from the generative model differs from observed data.
- **Adjust model parameters** to better match the data distribution

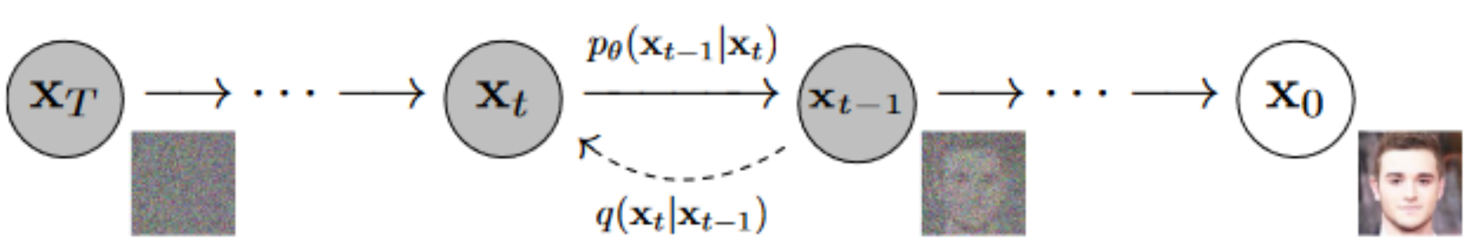


Ex. Generative Adversarial Networks
(I. Goodfellow 2014)

Diffusion models

- **Parametrized Markov Chains** trained using variational inference to produce samples matching the data after finite time.
  - Chain transitions are **reverse diffusions** (gradually adding noise to the data)
- Ex. DDPM (Diffusion Denoising Probabilistic Models) based on U-Net architecture, https://arxiv.org/pdf/2006.11239.pdf:
  - Iteratively add Gaussian noise to input image, eventually reaching pure noise
  - Generation process **inverts the diffusion:** start from pure noise sample, then iteratively de-noise it.





Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Normalizing Flows

Explicit density estimation

Bijective, differentiable maps between two continuous variables

- Compositional

  $x = g(z) = g_n \bullet \cdots \bullet g_2 \bullet g_1(z)$

- Simple prior density to complex target

$$\ln p(x) = \ln q(z) - \sum_i ln \left| \det\left(\frac{\partial g_{i+1}}{\partial g_i}\right) \right|$$

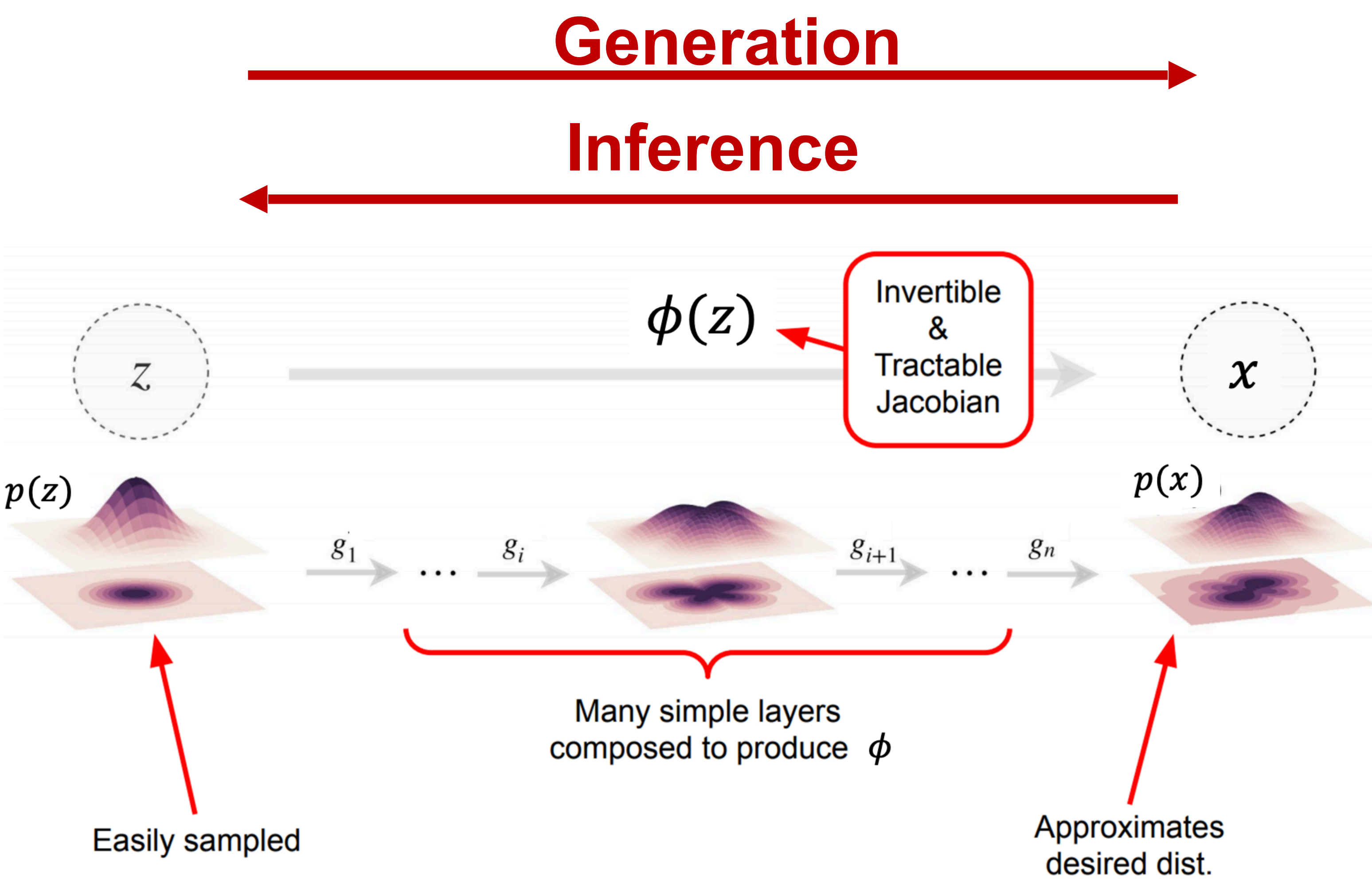


Image credit: G. Kanwa

Dinh et al. arXiv:1410.8516
Rezende Mohamed arXiv:1505.05770
Papamakarios et al. arXiv:1912.02762

Energy based models

*Model is an **energy function** measuring goodness of each (x,y) sample*

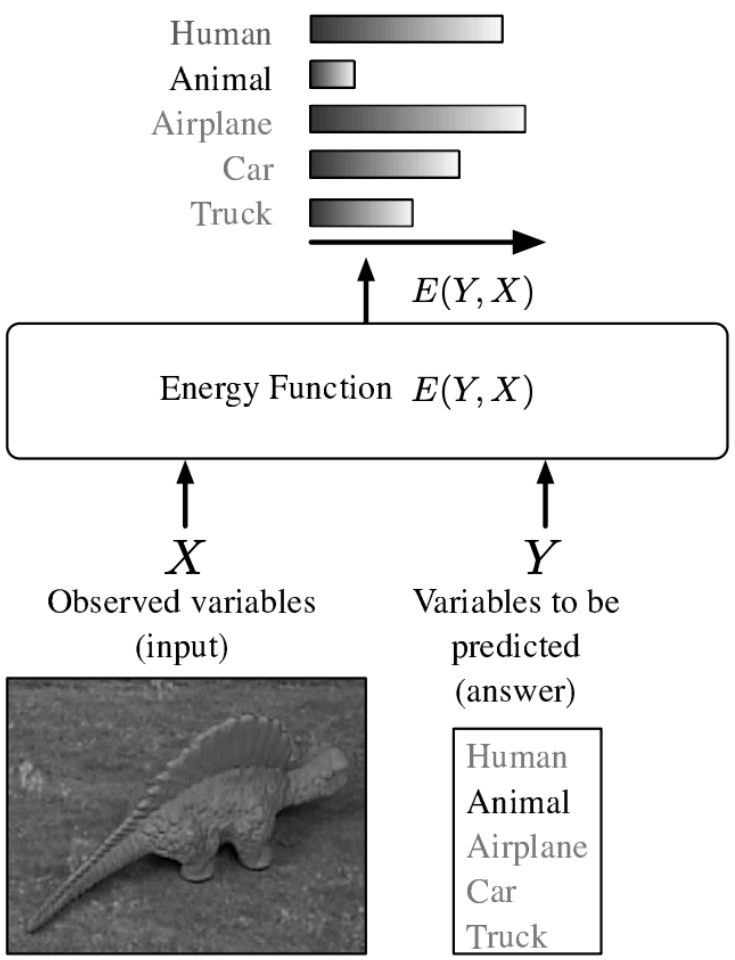
Inspired by **statistical mechanics**

- **Training**: Finding the best energy function E(W,X,Y).
- Minimise **loss functional** so that for any X_i, inference results in Y_i

$$\mathcal{L}(E,\mathcal{S}) = \frac{1}{P}\sum_{i=1}^{P} L(Y^i, E(W,\mathcal{Y},X^i)) + R(W)$$

- **Inference:** strategy to find Y that minimizes E(X, Y) for classification, regression, generation
- Model combination can be tricky due to **E scale**
- Interpret E function as PDF (**Gibbs distribution**)

$$P(Y|X) = \frac{e^{-\beta E(Y,X)}}{\int_{y\in\mathcal{Y}} e^{-\beta E(y,X)}}$$

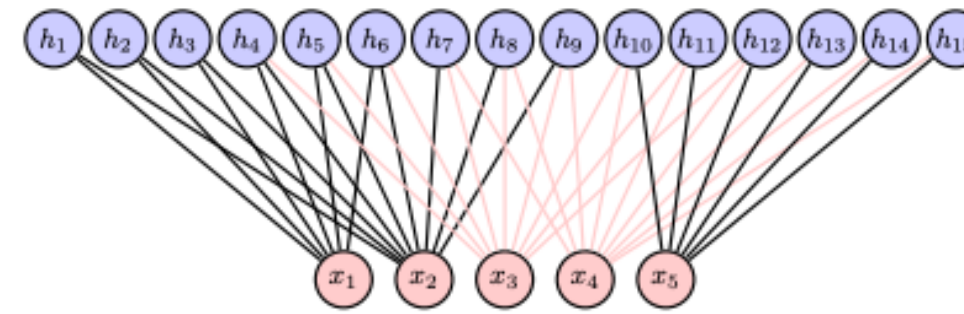


http://yann.lecun.com/exdb/publis/pdf/lecun-06.pdf

Sofia Vallecorsa, Ilaria Luise CERN - sofia.vallecorsa@cern.ch | ilaria.luise@cern.ch

Variational Calculation and Boltzmann Machines

Probability is a Boltzmann distribution



Ex. Compute **expected value** of physical observable

- Statistical mechanics defines a probability function

- Minimize the **free energy** -ln $\mathcal{Z}$ (intractable in general) by defining its variational form for a normalized variational probability q(x)

$$\pi(x) = \frac{e^{-E(x)}}{\sum_x e^{-E(x)}} \qquad \mathcal{Z} = \sum_x e^{-E(x)}$$

- L is an upper bound for the physical free energy – ln $\mathcal{Z}$
- Approximation is exact when variational distribution approaches the target

$$L = \sum_x \mathrm{q(x)ln}\frac{q(x)}{e^{-E(x)}} = \langle \mathrm{E(x) + ln\ q(x)} \rangle x{\sim}q(x)$$

$$L + \ln Z = KL(q\,||\,\pi) \geq 0$$

Attention and Transformers

A step back

See M. Kagan lecture on July 5th :
https://indico.cern.ch/event/1392500/

Recurrent States

29

- Input sequence $x \in S(\mathbb{R}^m)$ of *variable* length $T(x)$
- Recurrent model maintain a **recurrent state** $\boldsymbol{h}_t \in \mathbb{R}^q$ updated at each time step t. For $t = 1, \dots, T(x)$:

$$\boldsymbol{h}_{t+1} = \phi(\boldsymbol{x}_t, \boldsymbol{h}_t; \theta)$$

  – Simplest model:

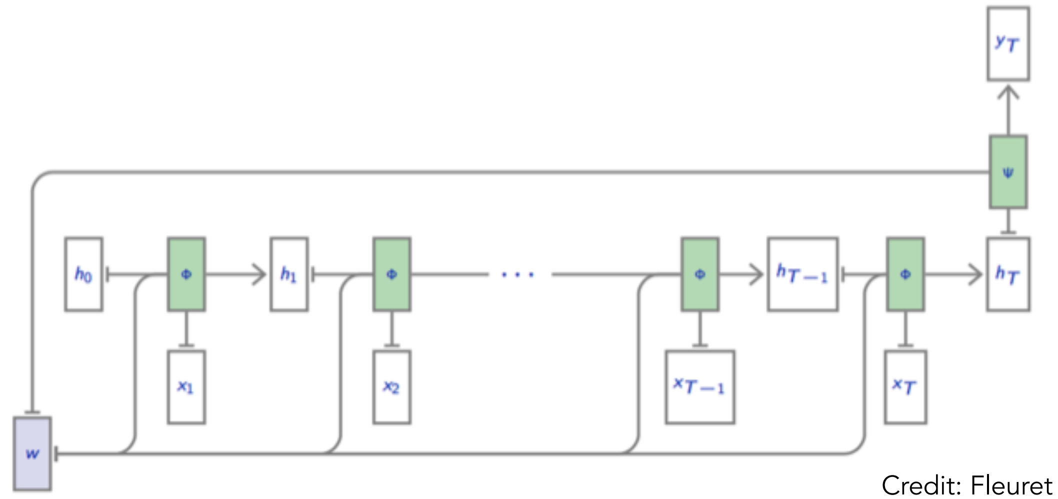
$$\phi(\boldsymbol{x}_t, \boldsymbol{h}_t; W, U) = \sigma(W\boldsymbol{x}_t + U\boldsymbol{h}_t)$$

- Predictions can be made at any time t from the recurrent state

$$\boldsymbol{y}_t = \psi(\boldsymbol{h}_t; \theta)$$

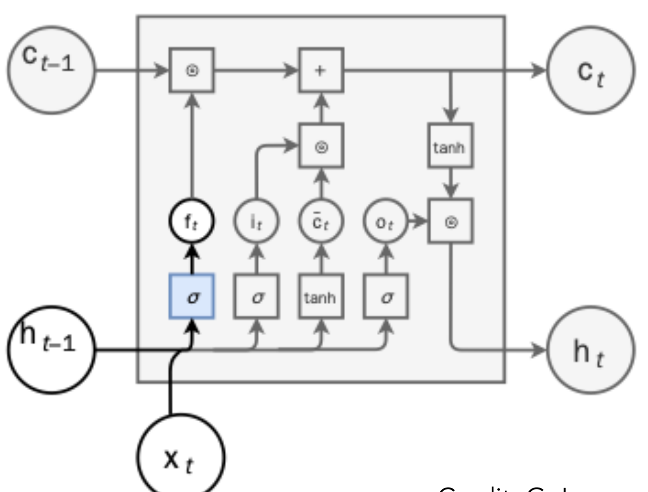
Credit: F. Fleuret

Recurrent Networks:



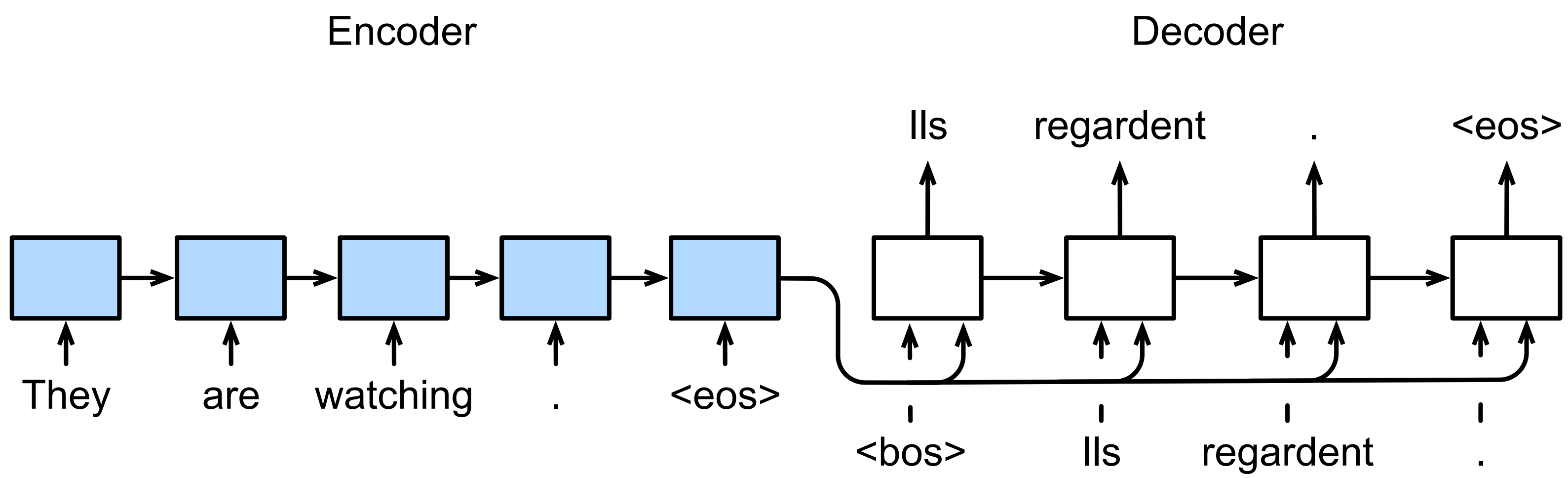
Credit: Fleuret

LSTMs:



Credit: G. Louppe

Seq2seq models



Credit: d2l.ai

Seq2seq models analyse sequences

Predict probability distributions of the next token given previous context

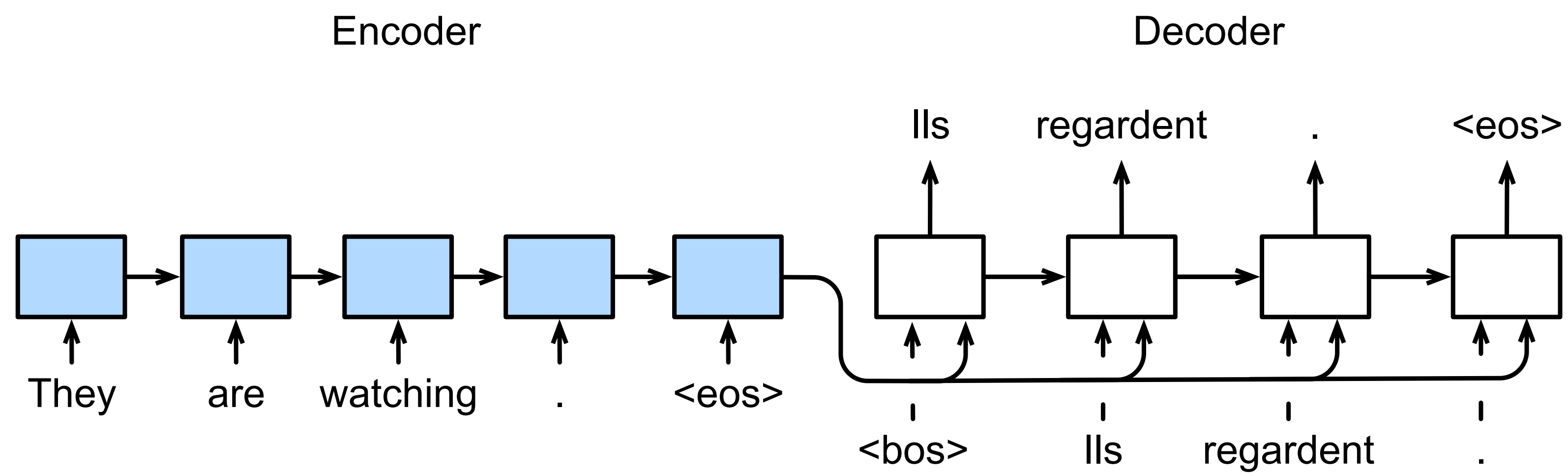
Encoder compresses the sequence in a fixed size vector

Fixed size latent vector is a bottleneck

Decoder **next-step generation is suboptimal** since latent vector contains the same information

Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Information bottleneck requires attention



Credit: d2l.ai

Can we avoid compression and pass the decoder entire input?

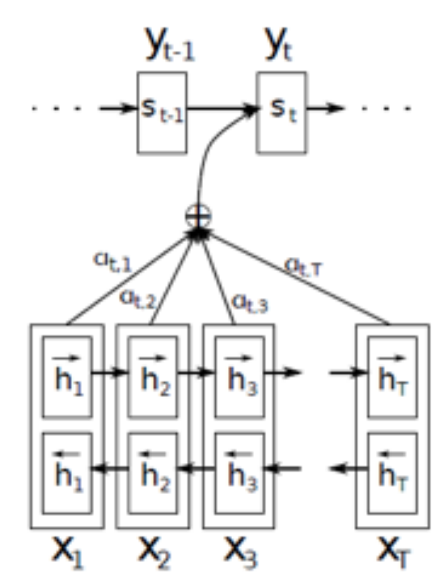
Need a mechanism to **focus on most relevant** input tokens at each prediction step

Introduce **softmax to calculate probability** (maintain differentiable architecture)

Output is **independent of the order** of input examples (set instead of sequences)

Use **relationships between input elements** (as graph representation).

Attention mechanism as originally formulated in a bi-directional LSTM Auto-Encoder
https://arxiv.org/abs/1409.0473



Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Attention mechanism

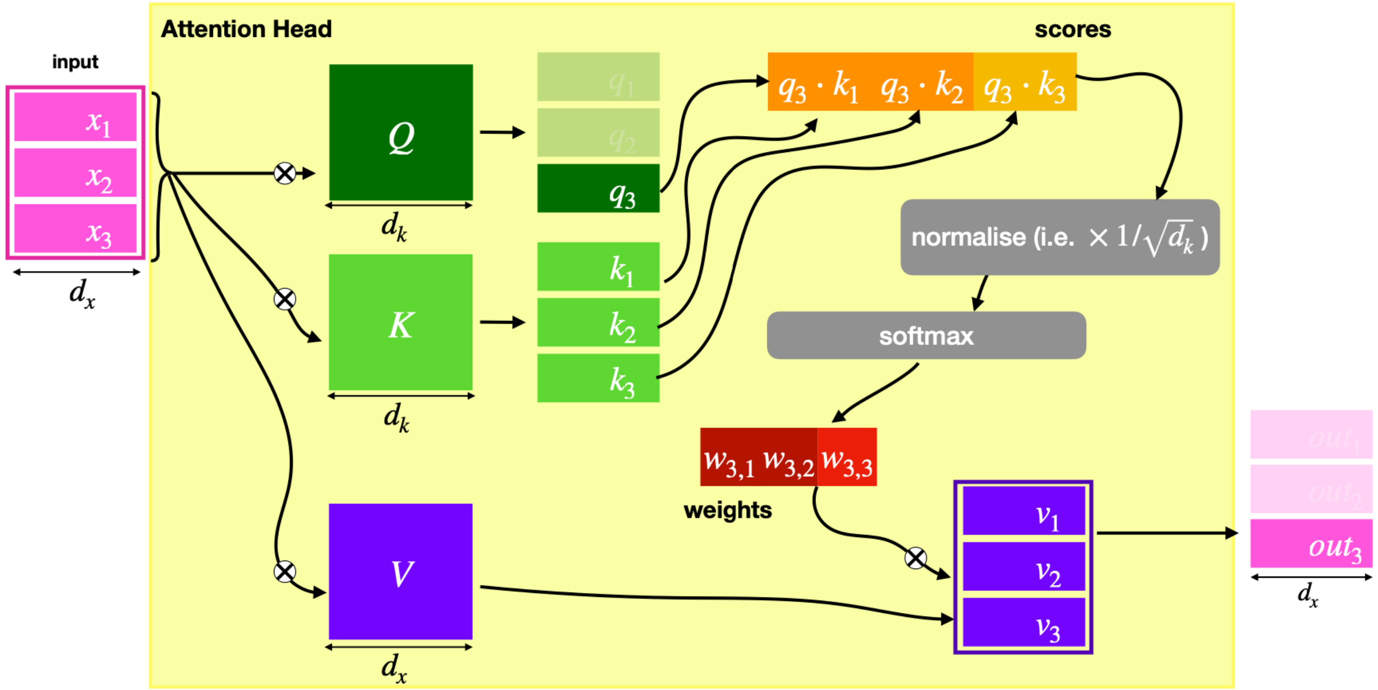
See tutorial G.. Weiss tutorial at IML workshop : https://indico.cern.ch/event/1297159/

A key-value database (differentiable, entries are continues vectors): $Q = \{q_1, q_2, \ldots, q_m\}$ QUERIES
$K = \{k_1, k_2, \ldots, k_n\}$ KEYS
$V = \{v_1, v_2, \ldots, v_n\}$ VALUES

A normalised **similarity** function between query-key pairs:

$$S_{ij} = \text{SIMILARITY}(q_i, k_j) \qquad A_{ij} = \text{NORMALIZE}(S_{ij}) = \frac{e^{S_{ij}}}{\sum_{l=1}^{n} e^{S_{il}}}$$

A **weighted average** over values {V}, based on similarity: $O_i = A_{ij} V^j$



Credit: G. Weiss

NB. Weights are probabilities (use softmax)

$$\text{SIMILARITY}(q_i, k_j) = \frac{q_i \cdot k_j}{\sqrt{D}}$$

Self-attention uses same input for values, keys and queries.
Focus on relationship between elements (adds context)

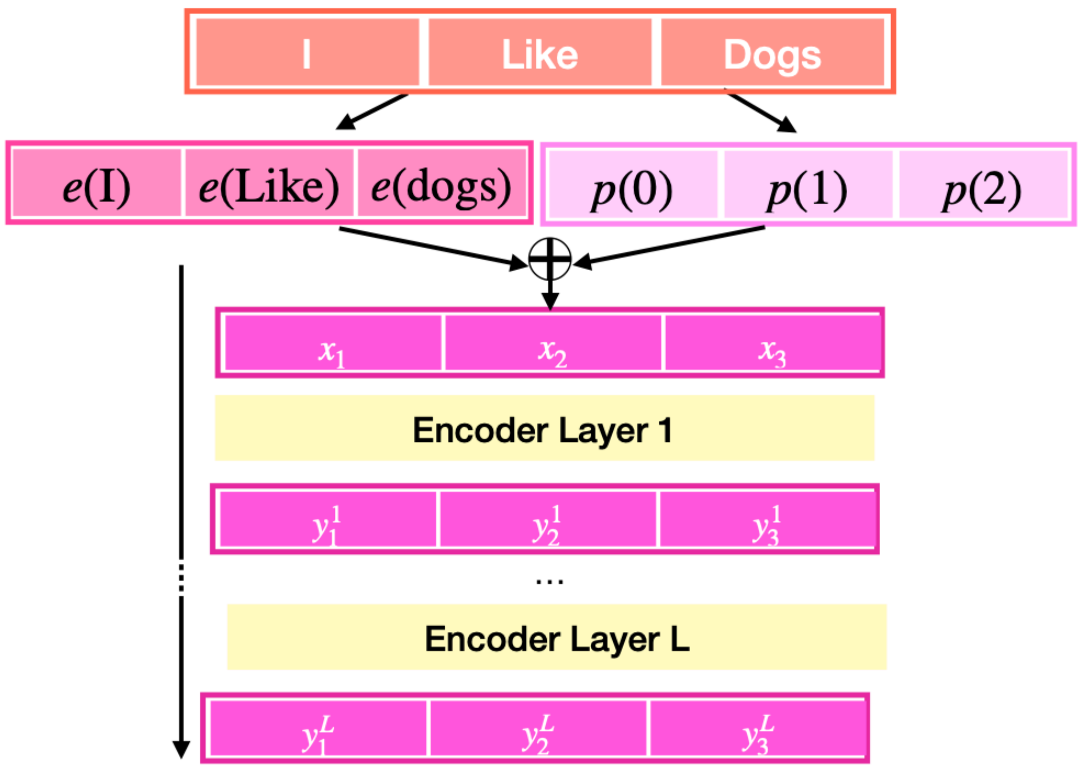
Multi-head attention splits input token in subgroups and processes them in parallel

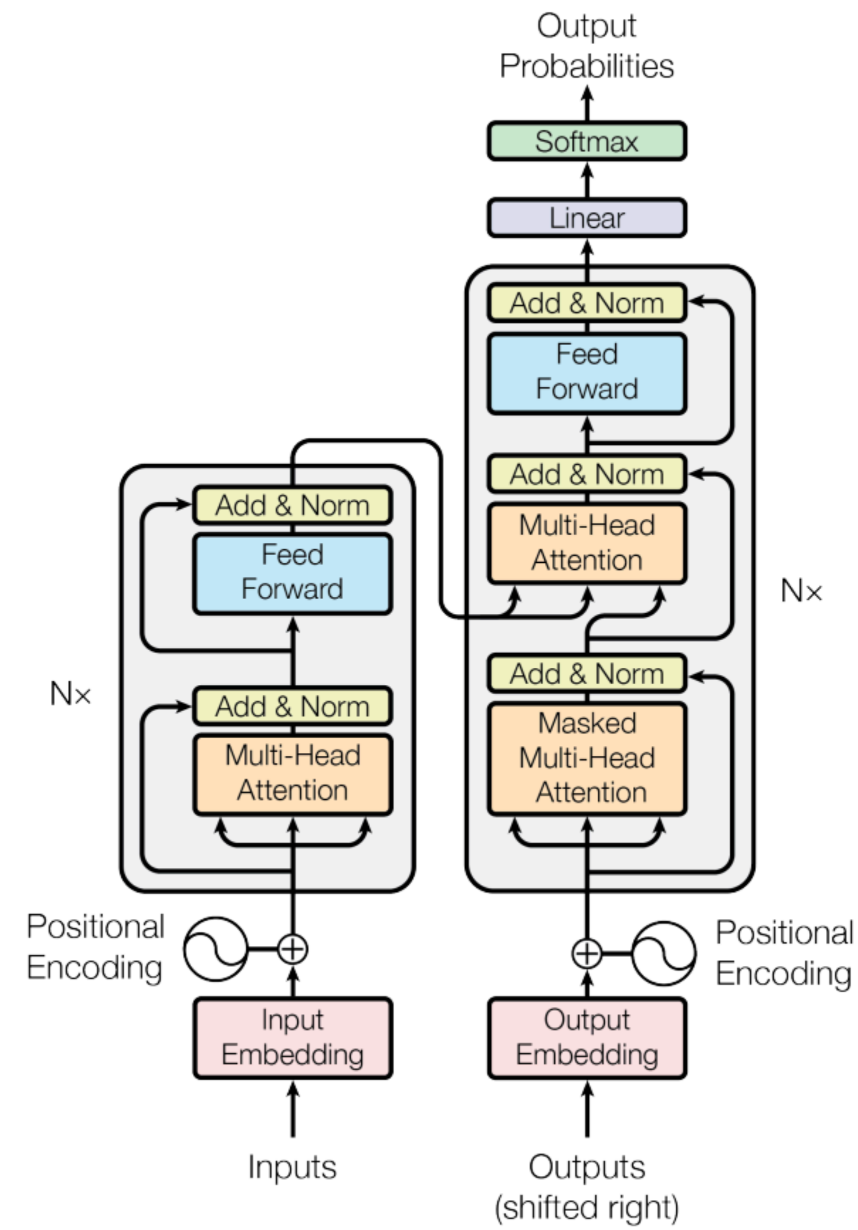
NB: Scaled dot-product is permutation equivariant

Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch

Transformers

See tutorial G.. Weiss tutorial at IML workshop : https://indico.cern.ch/event/1297159/





Transformer components include:

- Multi Head **Attention**
- **Normalisation** layers
- Position Independent **Feed Forward Layers**
- **Skip Connections**

NB. All tokens are processed in parallel

Vaswani et al., *Advances in Neural Information Processing Systems*, 2017, 5998–6008

Ilaria Luise, Sofia Vallecorsa CERN - ilaria.luise@cern.ch I sofia.vallecorsa@cern.ch