Rediscovery of Numerical Luescher's Formula from the Neural Network

Based on *Chin.Phys.C* 48 (2024) 7, 073101

Yu Lu (陆宇)

In collaboration with

Yi-Jia Wang (王一佳) Ying Chen (陈莹) Jia-Jun Wu (吴佳俊)

2024.11.08
@IHEP

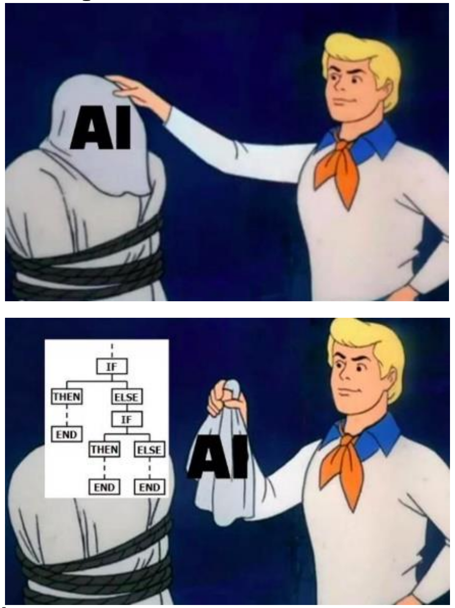
University of Chinese Academy of Sciences

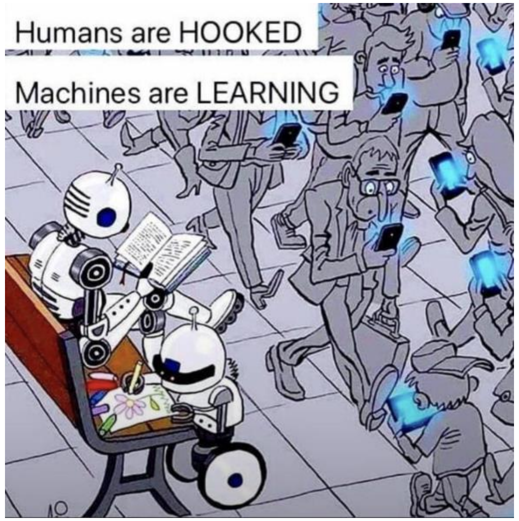
Contents

- Three Aspects of the Neural Network(NN)
  - Whether → Architecture
  - What → Loss
  - How → Training
- Examples
  - Extension of Architecture
  - Variational Auto Encoder (VAE)
  - Other examples & Comments
- Rediscovery of Luescher's Formula
- Comment & Outlook

Roadmap

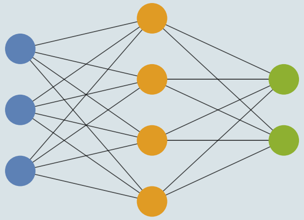
- Hard-Code(硬编码)
  → Knowledge Base(知识库)
  → Design Complicate Rules
  - 有多少人工就有多少智能(still true more or less)
  - e.g., cat-dog, ears? shape? size?
- Mis-impression:
  - Machines are stubborn, they can never be “creative”
  - Machines can only learn/memorize what people teach them
- ☆Self-discovery of the pattern/representation
- Simple Concepts → Hierarchy → Complicate Concepts
  - Multilayer perceptron (MLP, 多层感知机)
    or Feed Forward Neural Network(NN)
    (前向全连通神经网络)

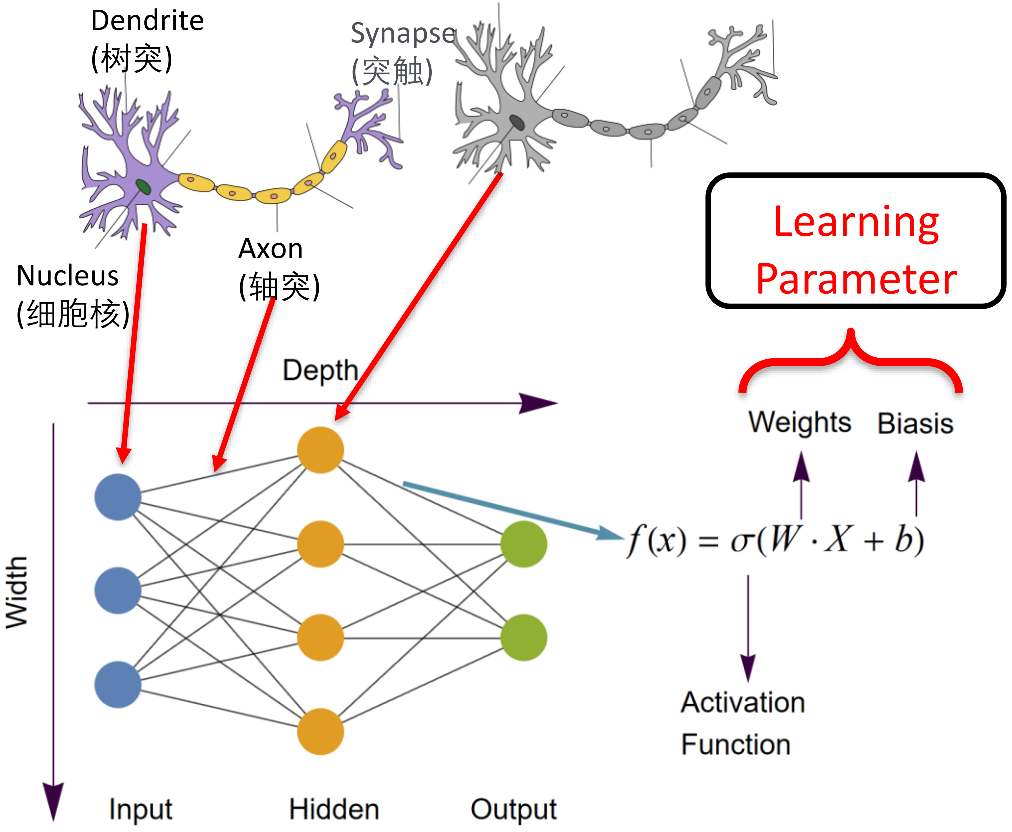




AI = Machine Learning(ML) > (≈?)NN

NN Architecture

Anatomy	NN
Dendrite(树突) Nucleus(细胞核)	Node - Input/Output/Hidden - Float Points - 1D Vector, 2D Matrix, nD Tensor…
Axon(轴突) Synapse(突触)	Activation Functions - Typically Predefined - Affine Transformation Everywhere
Complicate Network 	Stack of Layers (Except GNN) 



Essentially a nesting of simple functions

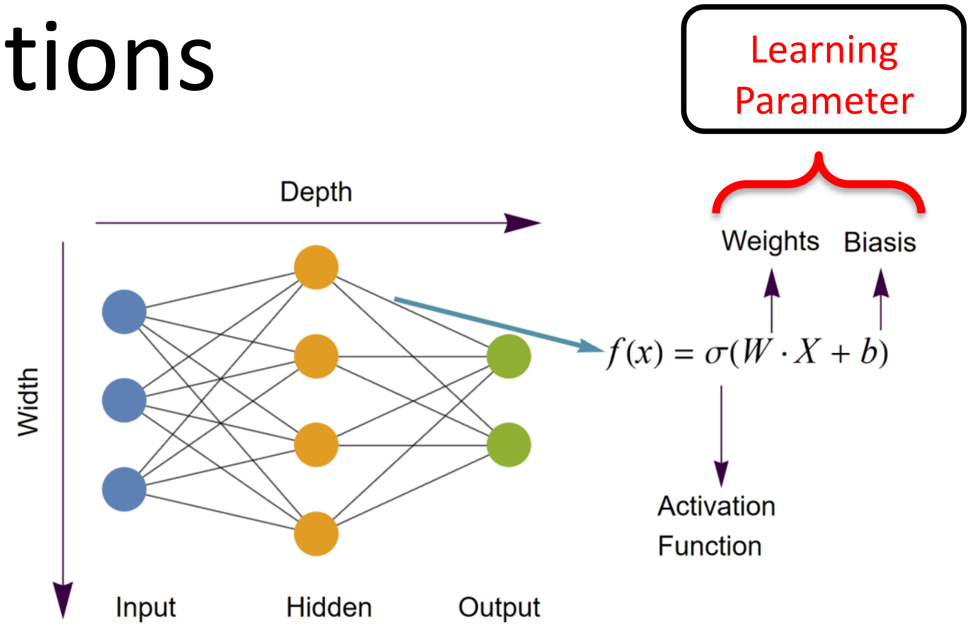
$$x^{(i)} = f_i\left(x^{(i-1)}; \theta\right), \theta = \{W_i, b_i\}$$

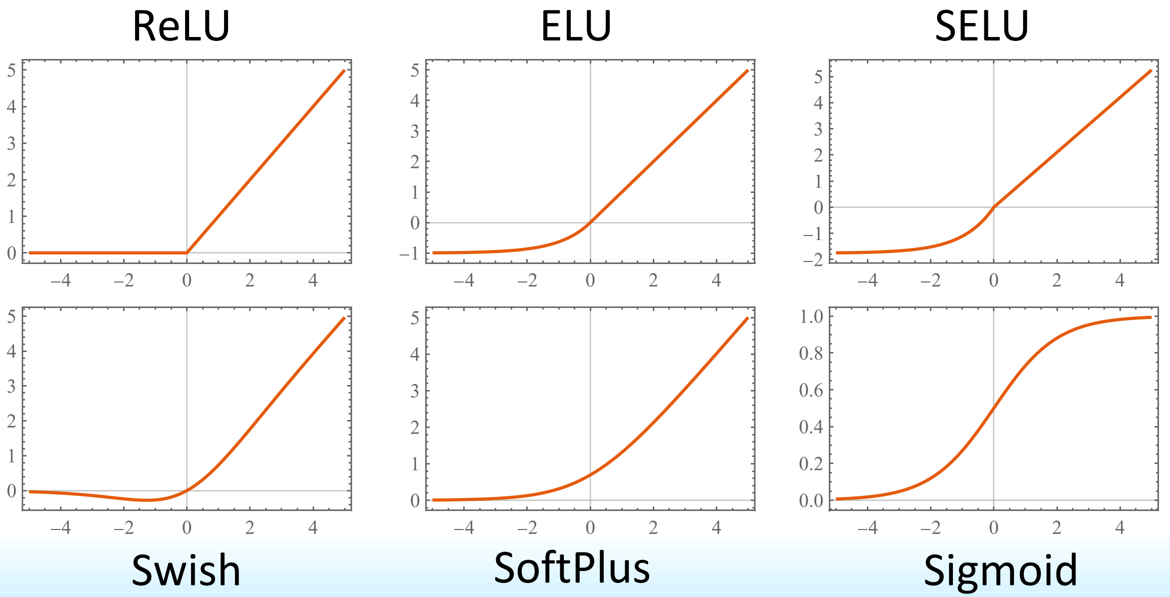
NN = Nonlinear functions + Affine Transformation

How to choose f?

Activation Functions

- Principles
  - Can be **any** continuous function
  - Typically has threshold/saturation + Varying Region
  - Simple Function V.S. Complex Structure
- Some Examples
  - Rectified Linear Unit (ReLU, 整流线性激活函数)
    - Piecewise linear function
    - Simple but **unexpectedly effective** (in CV, NLP, etc)
    - Nearly dominates NN (not in our work)
  - SoftPlus: $\log(e^x + 1)$
    - Smooth Version of ReLU
  - Triangular Functions
    - sin, cos, etc…
- That's All~

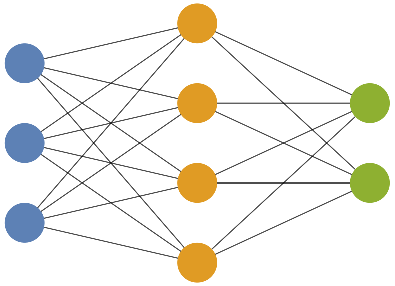




NN Can Approximate "Any" Function

Universal Approximation Theorem (**UAT**)

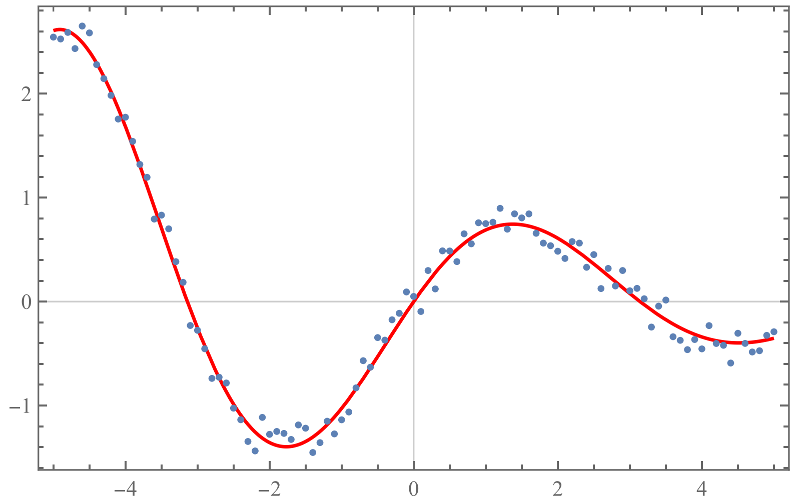
- $\approx$ A deep and wide-enough NN can approximate "any" function
  with "suitable activation functions"
  - Versatile, can do anything
- Increasing depth is more efficient than expanding the width
  $\rightarrow$ **Deep Learning**
- … With the cost of exploding number of parameters

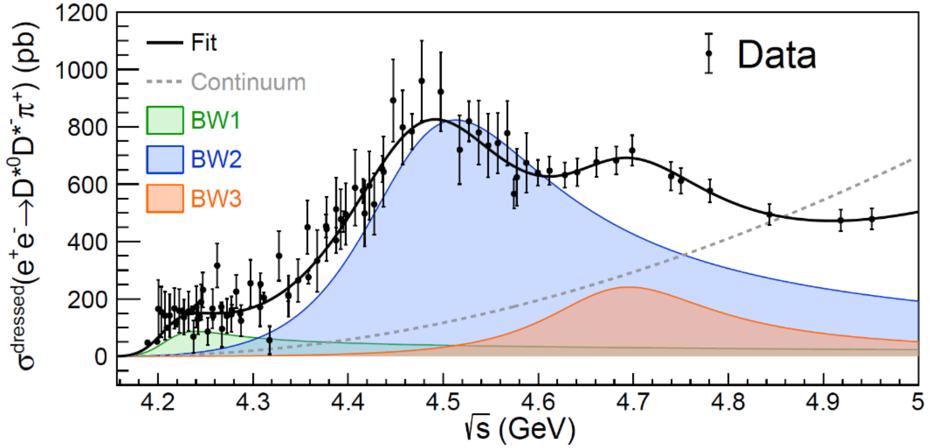


UAT: Hornik *et. al*, Neural Networks 2, 359; ibid, 4, 251; ibid, 6, 861

How to distinguish Good or Bad

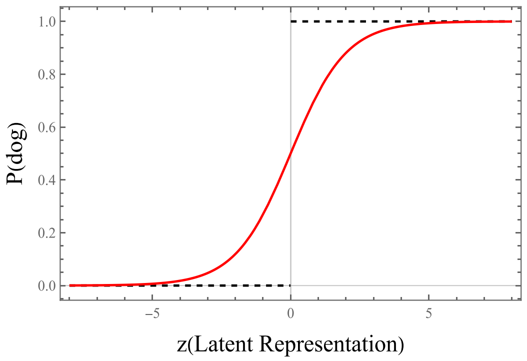
- In phenomenology:
  - a good description ≈ a good fit
  - χ^2-fit, F-test…
- What is a good NN? Let loss functions judge!
  - NN is nothing but a function: $x \to NN(x;\theta)$
  - $NN\left(x^{(i)};\theta\right)$ V.S. $y^{(i)}$
- Regression → Mean Square Error or χ^2
  - $\mathrm{MSE} \coloneqq \sum_i \left(\hat{y}_i - f(x_i;\theta)\right)^2$
- Loss function Judges All
  - Different Problems are defined by different loss functions

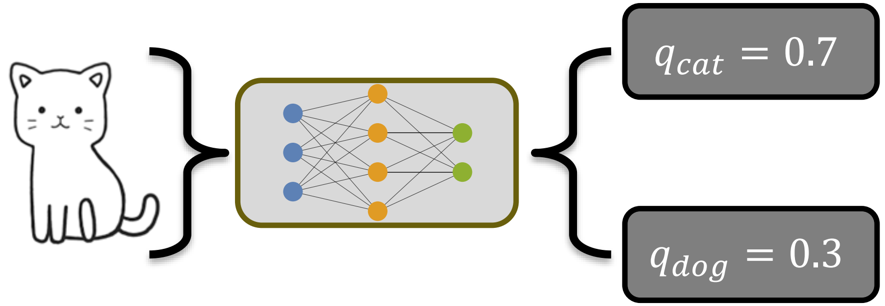




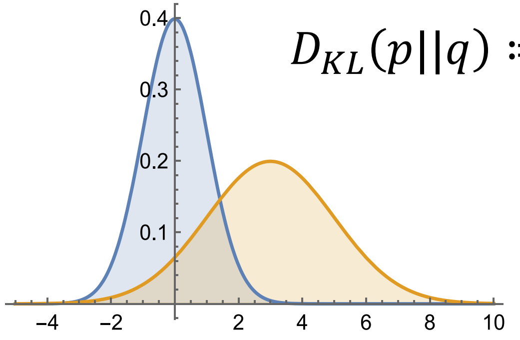
Loss function of Classification Problem

- **Soft**max activation
  - SoftMax: $\frac{\exp(z_i)}{\sum_j \exp(z_j)}$ Sigmoid : $\frac{1}{\exp(-z)+1}$
  - Probability summation =1
  - **Differentiable** version of Max
- Cross Entropy
  - $H(p,q) = -\sum_x p(x) \log q(x)$
- Generalizations
  - KL Divergence etc

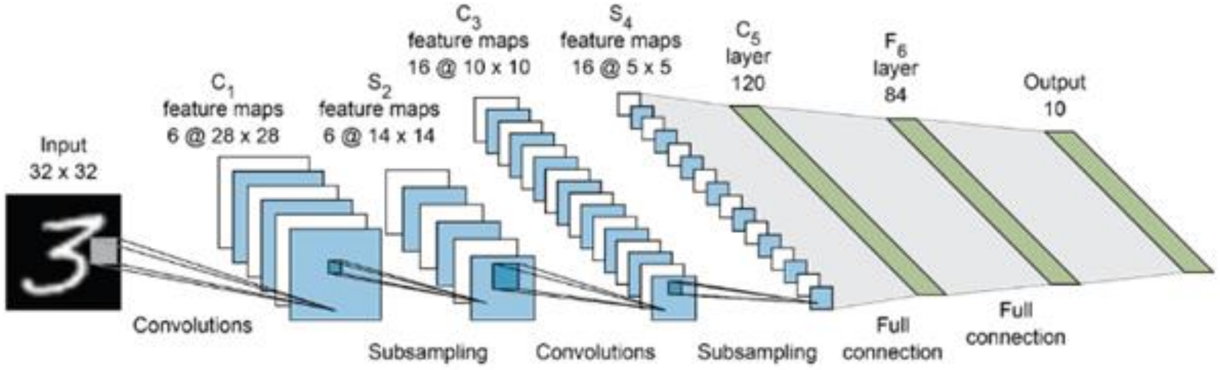




$$H = -\log 0.7 \approx 0.357$$

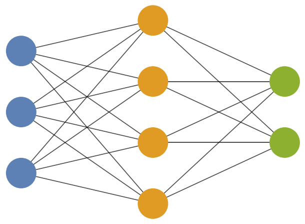


$$D_{KL}(p||q) \coloneqq \sum_x p(x) \log \frac{p(x)}{q(x)}$$



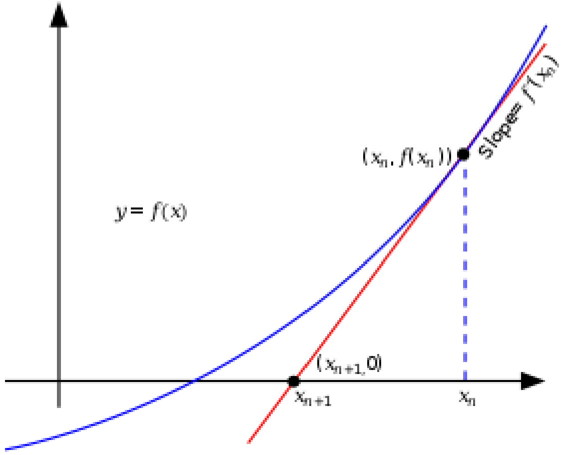
How to Train a NN

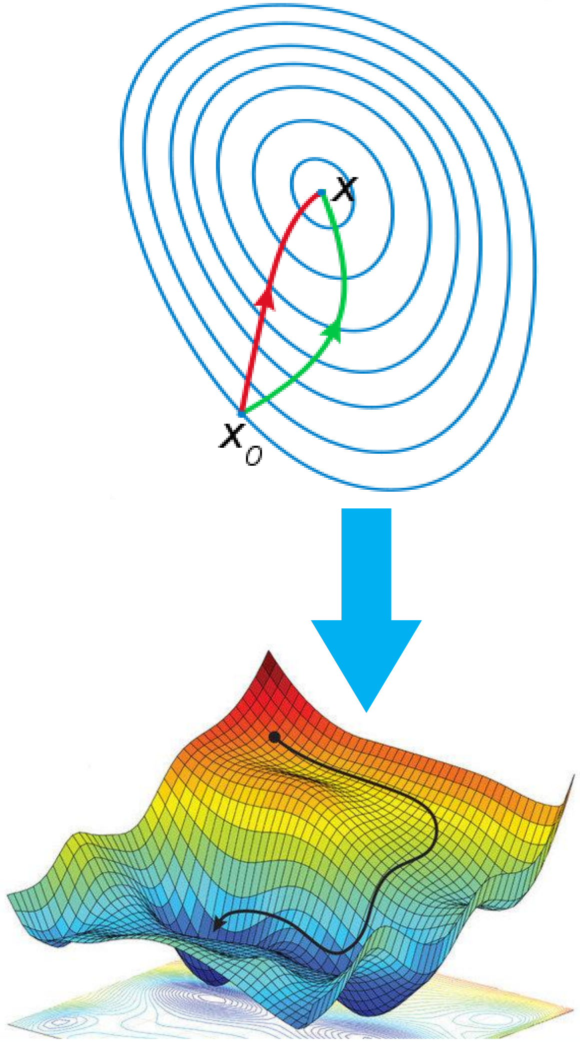
	Physical Model	Neural Network
Parameters	~10(<100)	>10^5 (even 10^9, 10^10)
Calculations	Integral, Differential, Recursive…	Affine Transformation (Matrix Multiplication)
Type	Non-Linear	Non-Linear
Difficulty	Hard	Simple (in some sense)
Device	(mainly) CPU	GPU/TPU
Method	“Orthodox” Numerical Methods (Quasi-) Newton Method	**Automatic Differentiation** **Stochastic Gradient Descent** **Back Propagation**
Tools	ROOT/Minuit/D.I.Y	PyTorch/JAX/TensorFlow/MXNet



How to Train a NN

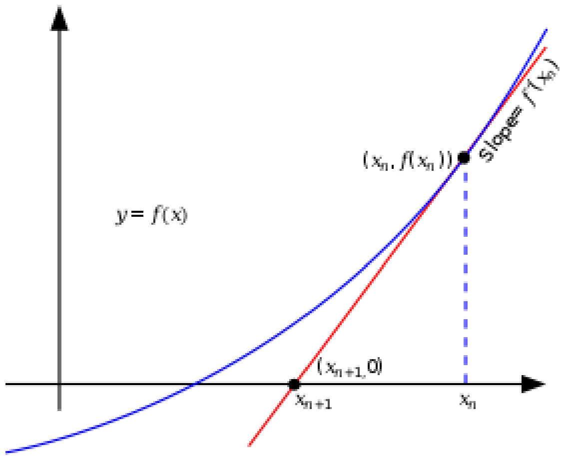
- Every problem is an optimization problem
  - Hamiltonian ↔ Lagrangian
  - Differential Equation ↔ Least Action Principle
- Orthodox way
  - Newton's method in Root Finding:
    $$x_{n+1} = x_n - f(x)/f'(x)$$
  - Newton's Method In Optimization:
    $$x_{n+1} = x_n - f'(x)/f''(x)$$
- $f'(x), f''(x)$ may be difficult
  - Use approximated $f''(x)$
  - Quasi-Newton methods: e.g. BFGS etc
- Challenges from NN
  - many saddle points / local minima





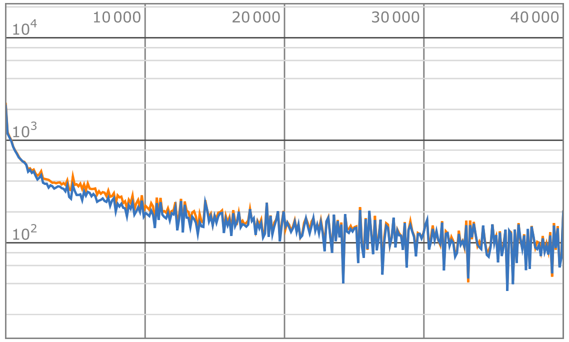
NN tells Newton:
Don't pursue perfection
Just do it Slooowly!

How to Train a NN



$$x_{n+1} = x_n - f'(x)/f''(x)$$

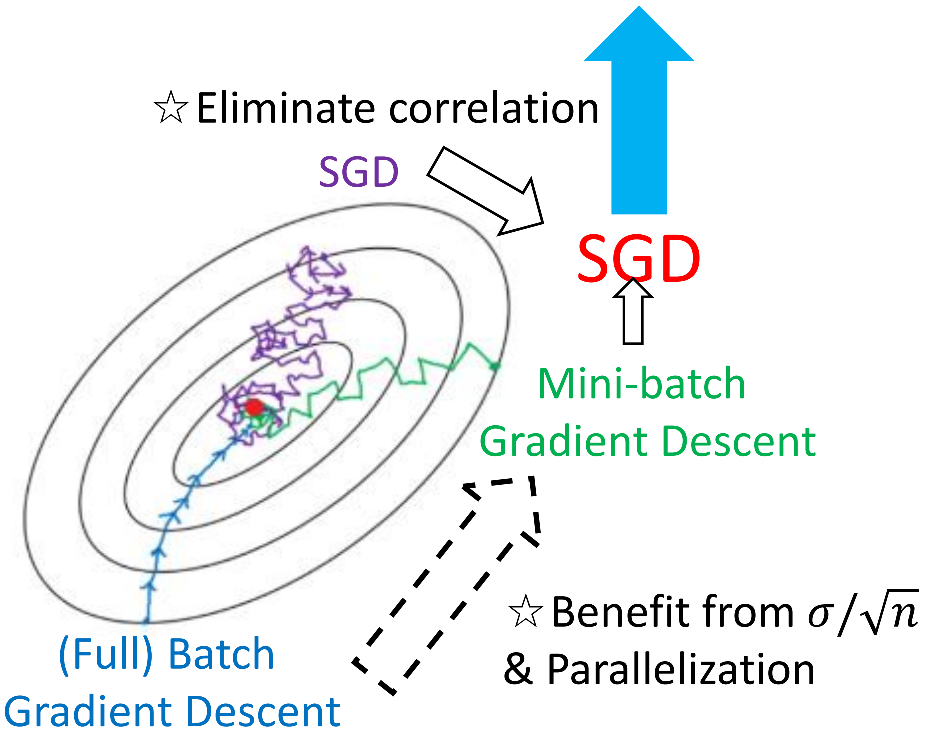
$$x_{n+1} = x_n - \epsilon\, f'(x)$$



- Stochastic Gradient Descent (SDG)
  - In practice: Randomize the data, then mini-batch
  - learning rate ϵ, (typically $\sim 10^{-3}$)

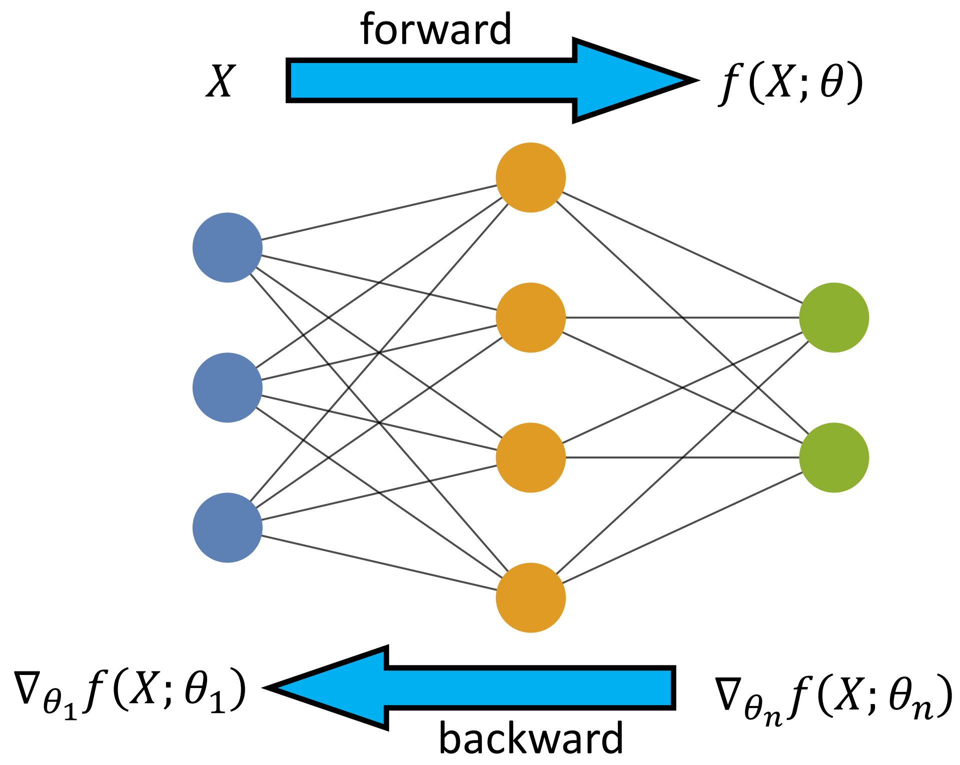
    (randomly selected)mini batch $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

    $$\boldsymbol{\theta} \to \boldsymbol{\theta} - \epsilon \frac{1}{m} \nabla_\theta \sum_i L\left(NN\left(x^{(i)}; \boldsymbol{\theta}\right), y^{(i)}\right)$$
  - ∇_θ^2 is expensive, rarely used in practice
- How to get $\nabla_\theta L$?



Automatic Differentiation + Back Propagation

- $\nabla_\theta L$ is the crux
- Back Propagation
  - Each activation function is “simple”
  - Hierarchy -> Chain rule
    - $\nabla_{\theta_1} f(g(x;\theta_1);\theta_2) = \nabla_{\theta_1} g(x;\theta_1) * f'(g(x;\theta_1);\theta_2)$
    - $f(x;\theta) \coloneqq f(\boldsymbol{W}.\boldsymbol{X} + \boldsymbol{b})$ (affine transformation) -> GPU
  - Implemented in Pytorch, Mathematica (MXNet)…
- General Technique: Automatic Differentiation (AD)
  - Not numerical $f(x+\delta) - f(x)$, nor symbolic
  - “Differentiate” the code (including loops): $\sin x \to \cos x$
  - Get $f'(x)$ along the way
  - Packages: Jax (Python), NiLang (Julia) …
  - Not sure ROOT has implement this or not



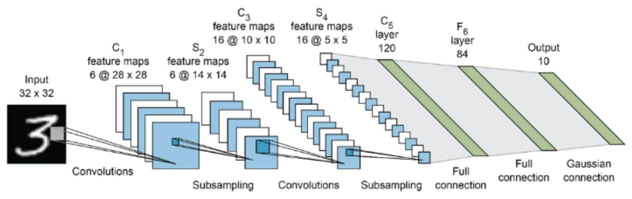
NiLang

See “Deep Learning” Chapter 6.5 for the details

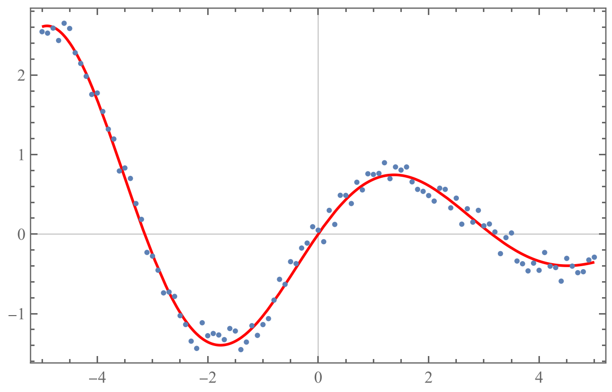
Short Summary

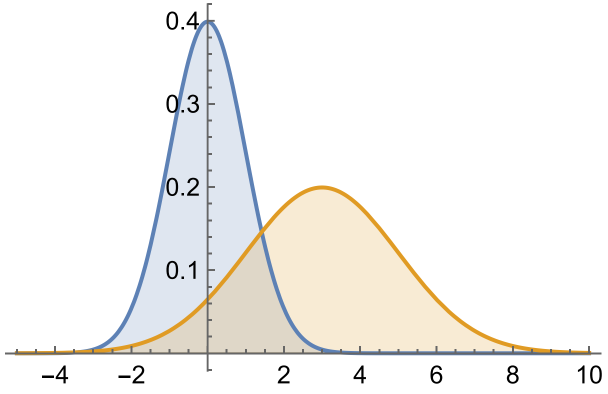
- Architecture (Whether)

UAT

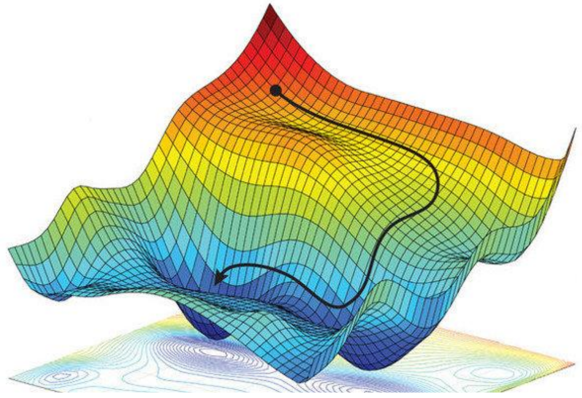


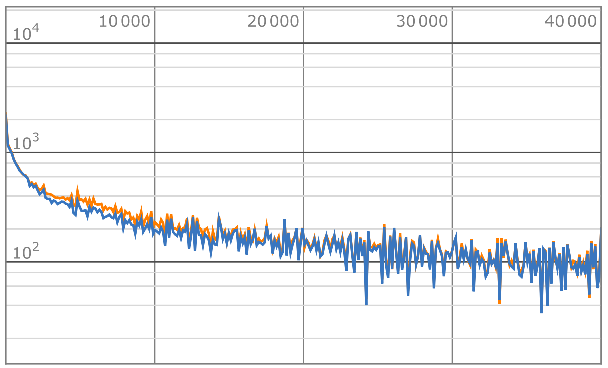
- Loss Function (What)





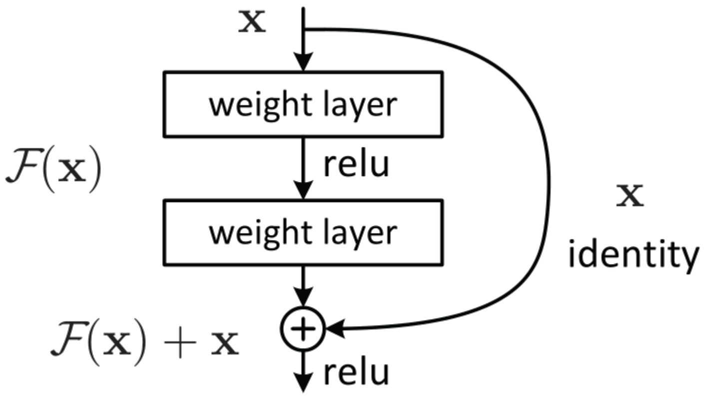
- Optimization (How)

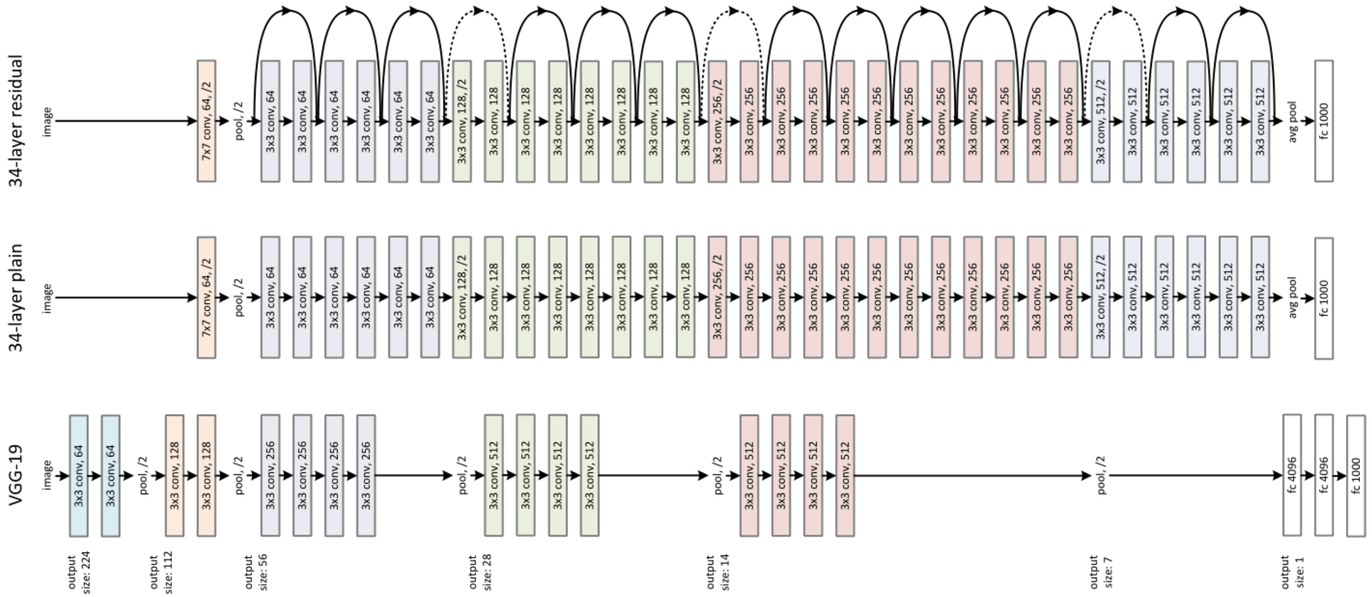




Architecture Extensions

- ResNet
  - Introduce Jump-Link to Accelerate Training & Inference
  - $F_{n+1}(x) = \boldsymbol{x} + F_n(x)$
  - Alleviate the Gradient-Vanishing/Exploding Problem
  - Make training deep NN possible





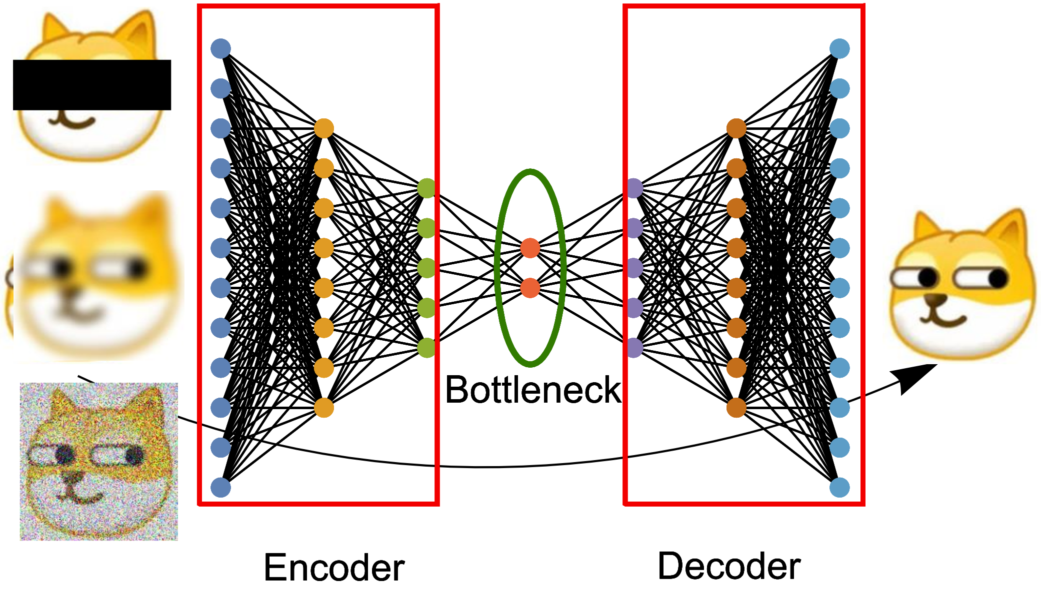
One small step for a NN
one giant leap for the ML community!

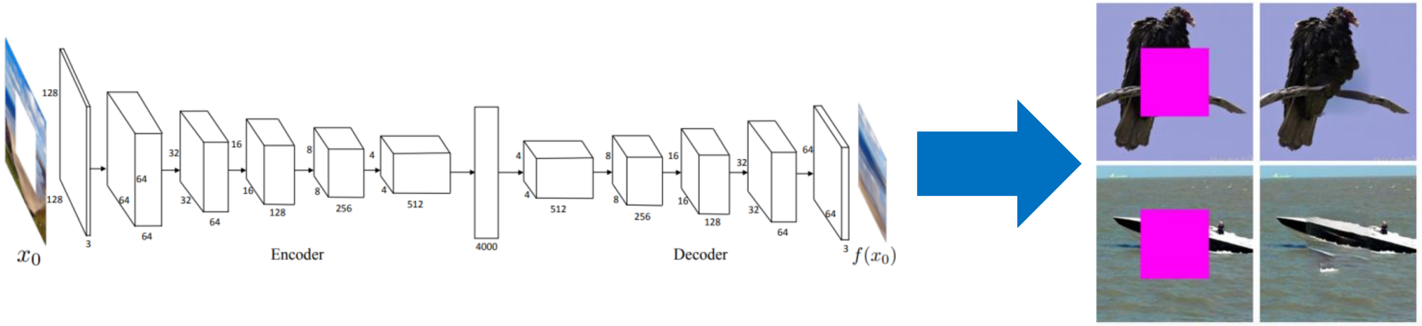
Kaiming He (何恺明) *et. al* [arXiv:1512.03385]

KAN. Ziming Liu *et. al* [arXiv:2404.19756]
UATs are still not fully understood!
Much to explore

AutoEncoder(自动编码器)

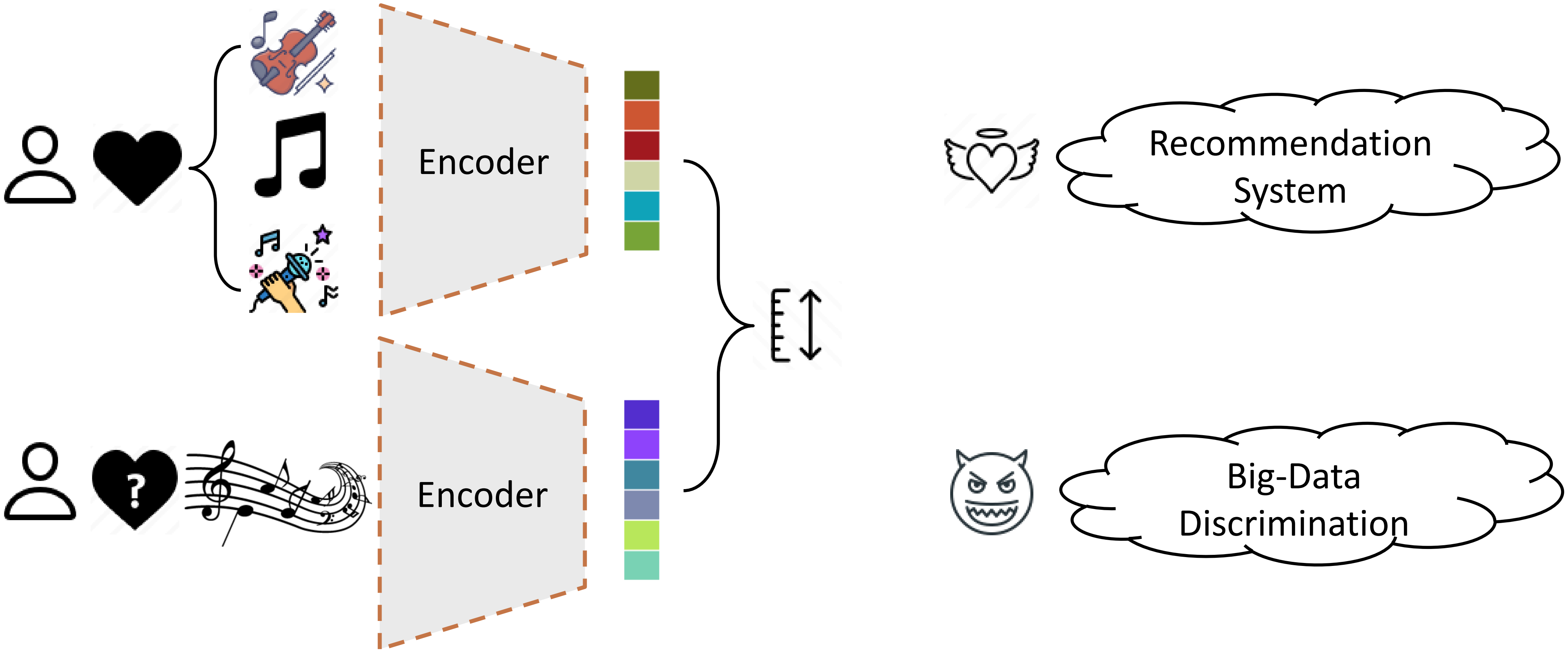
- Dimension Reduction
- Mask Trick
  - Denoising, Inpainting
  - Super Resolution (naïve DLSS)
- Not Magic, Merge Info in Other Samples to handle ill-definedness



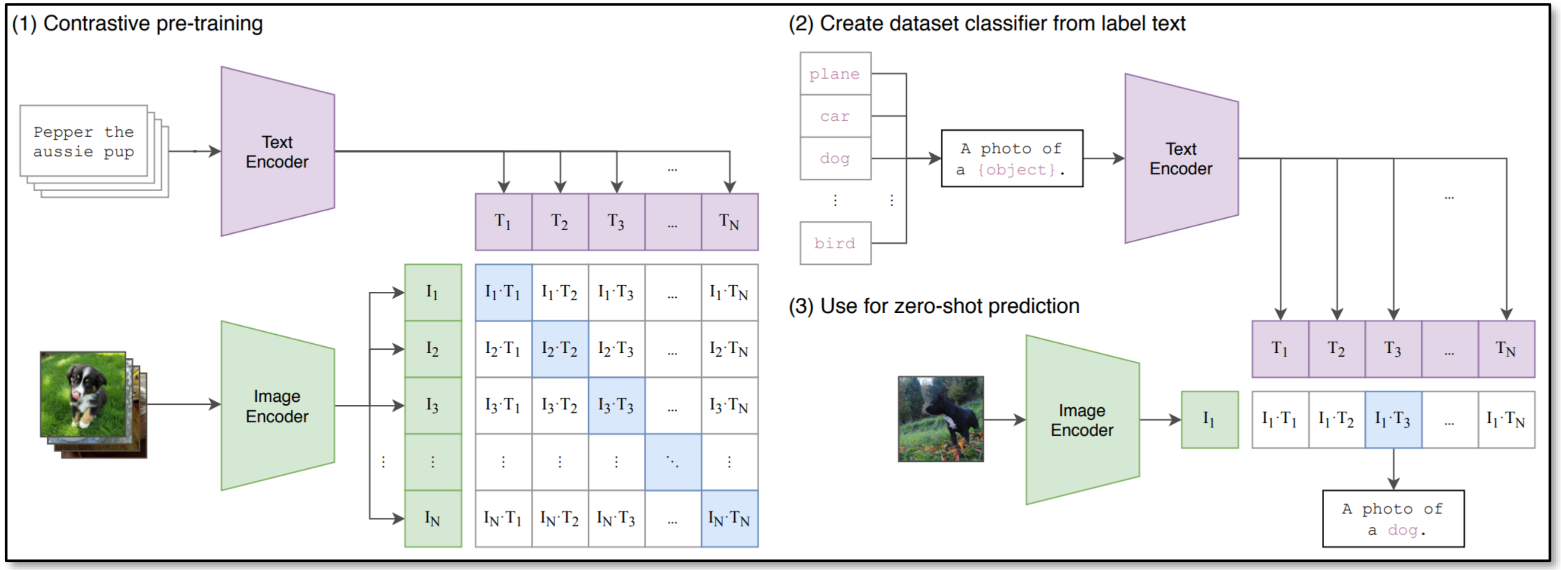


arxiv:1611.09969

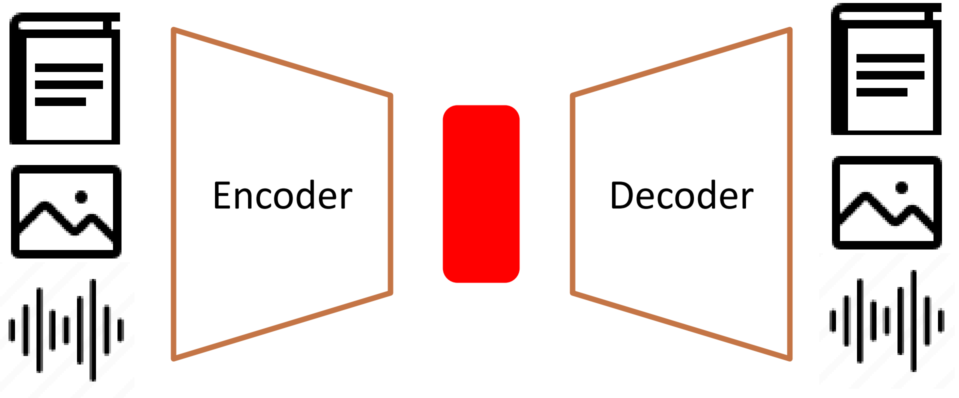
Encoder-Decoder Examples



Encoder Decoder Examples



[CLIP by OpenAI, arXiv:2103.00020]

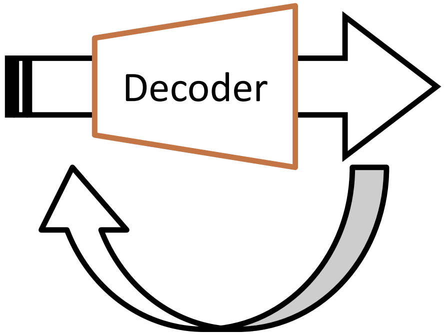


Good representation is a Treasure

Language { Q&A, Programming, Translation, Literature, …

⇩

Predict the next word



Large Language Models (LLMs)
ChatGPT

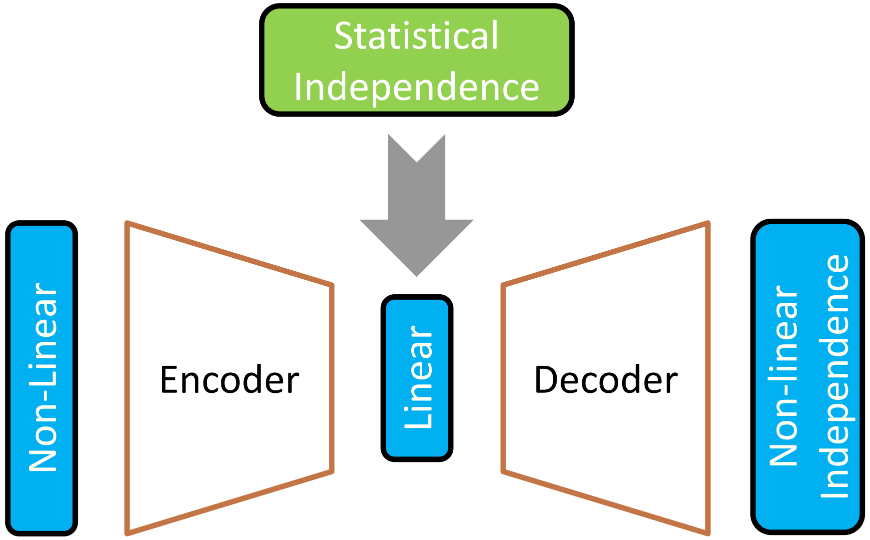
"I am going to reveal the identity of the criminal, and that person's name is …" by Sutskever

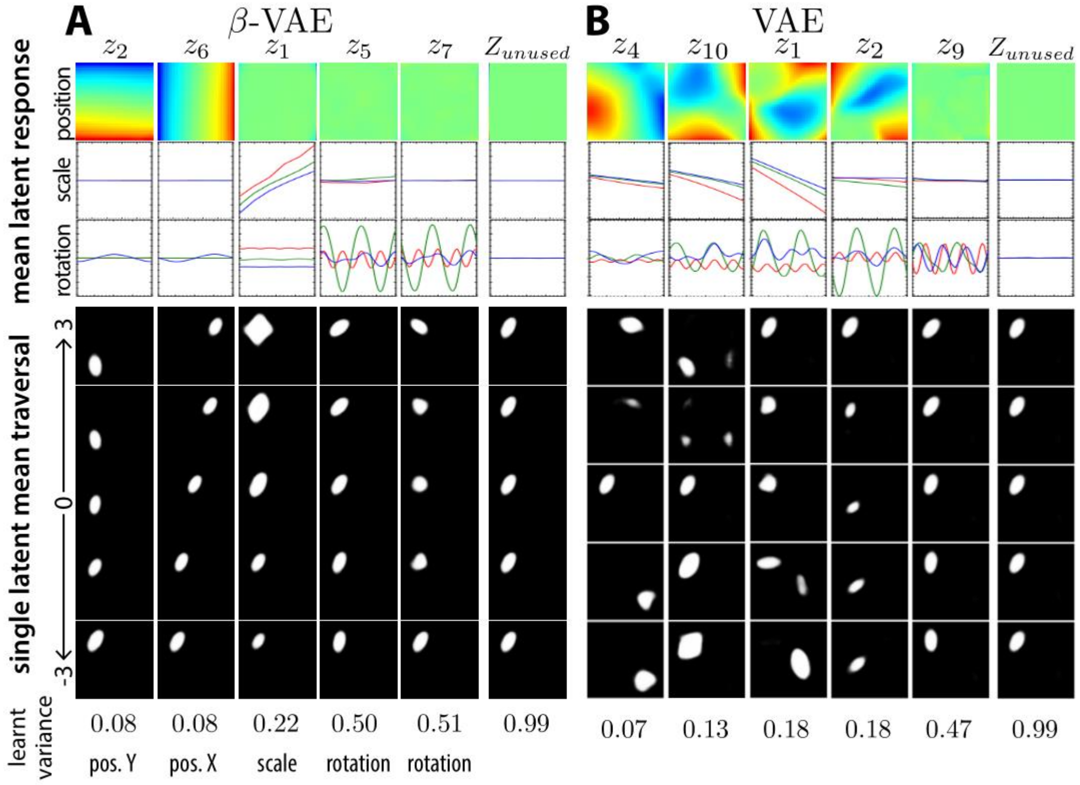
Statistical Inference ≈ logic?

"All models are wrong, but some are useful"
-- aphorism in statistics.

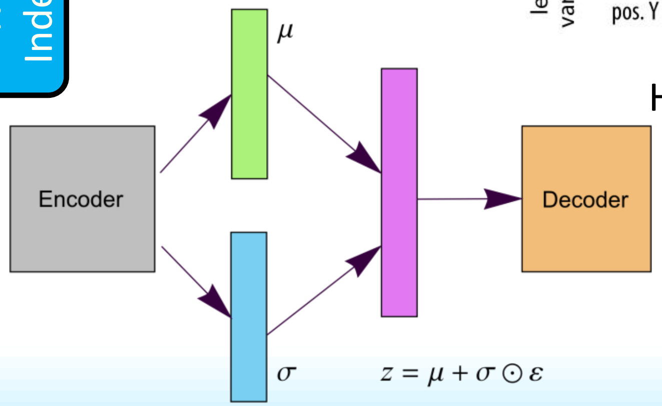
(β)Variational Auto-Encoder(βVAE变分自动编码器)

- Latent Feature Interpretation
  - Disentangle latent factors
  - Realized by Statistical Independence





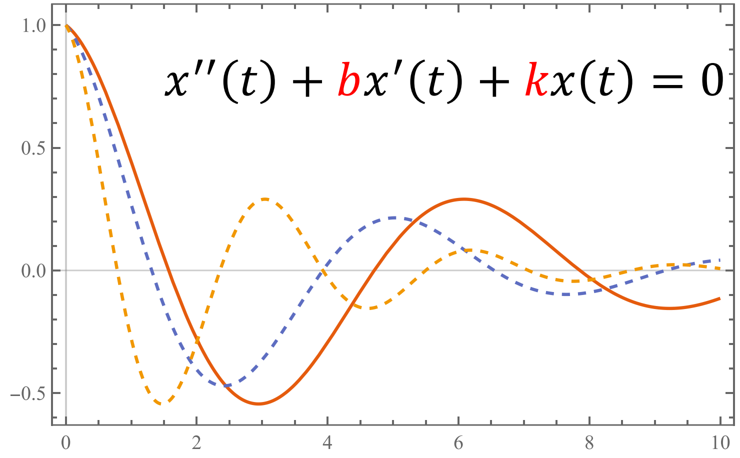
Higgins *et.al* https://openreview.net/forum?id=Sy2fzU9gl



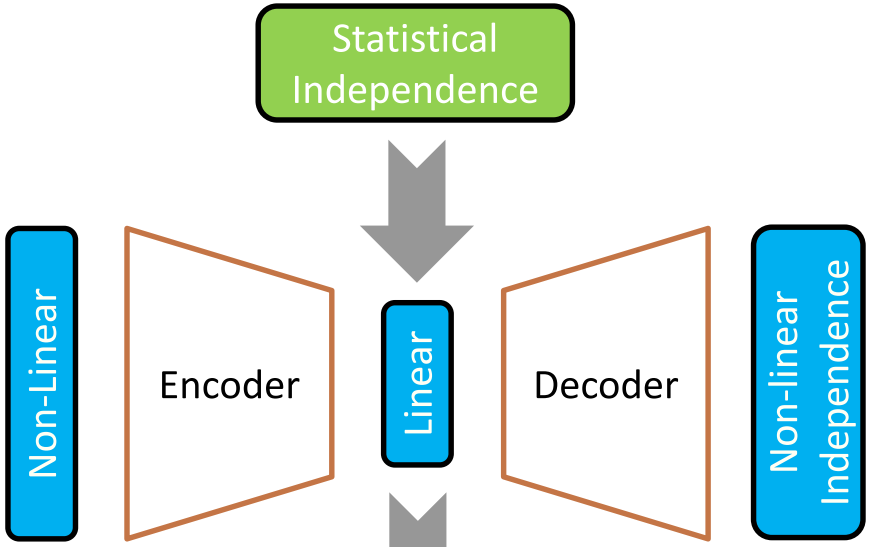
$$\operatorname{argmax} E_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}\left(q_\phi(z|x)||p(z)\right)$$

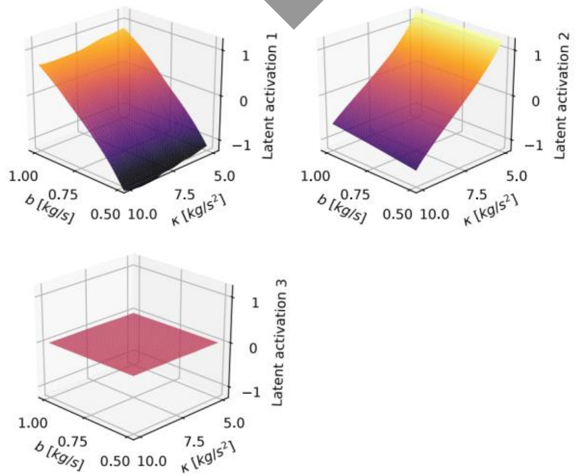
βVAE in Physics

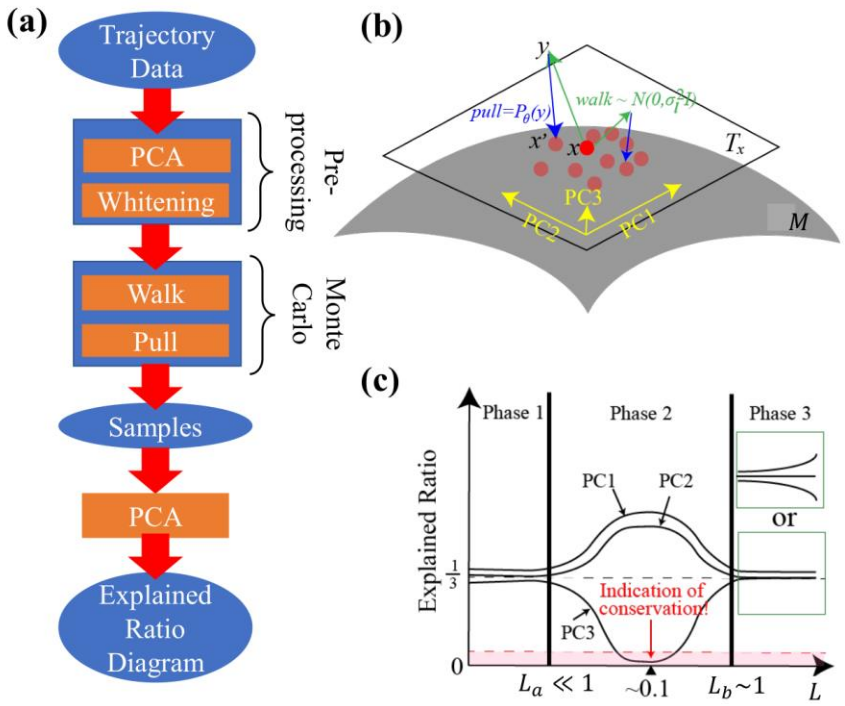
PhysRevLett.124.010508



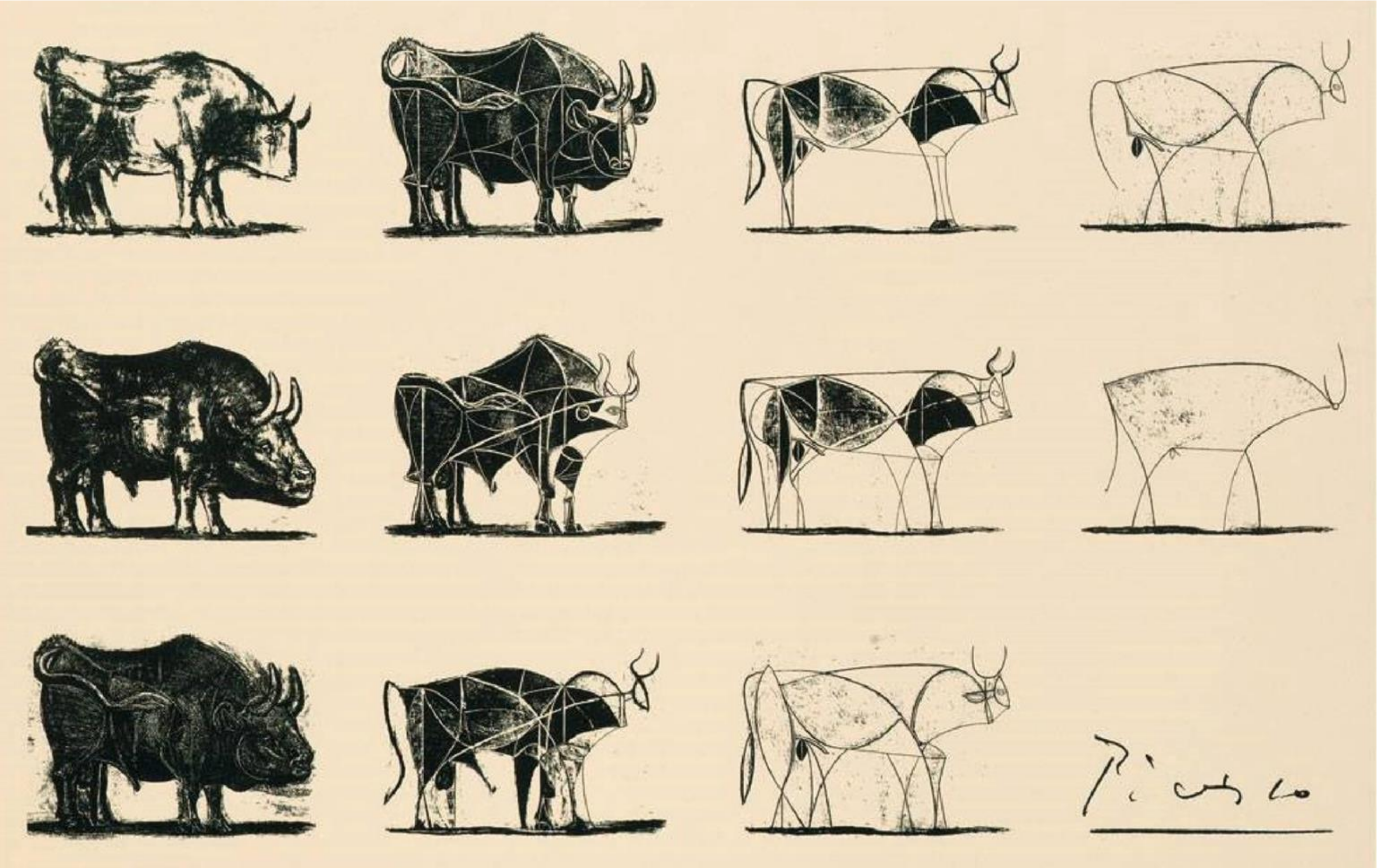
Only Two are effective D.O.F!







Philosophically Similar:
AI Poincare [PhysRevLett.126.180604]

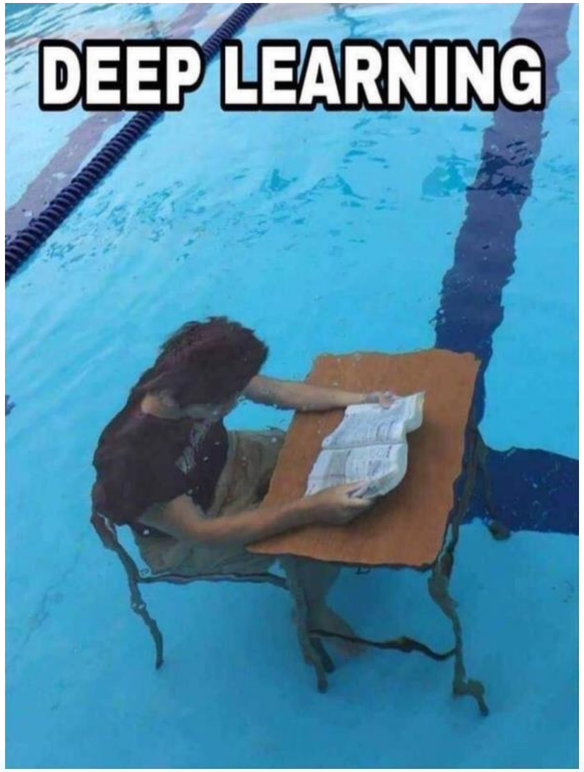


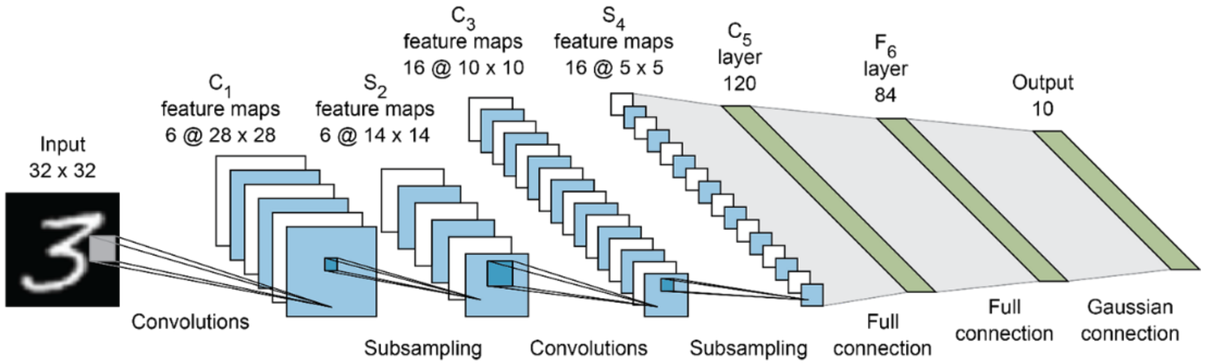
“Perfection is achieved,
not when there is nothing more to add,
but when there is nothing left to take away.”

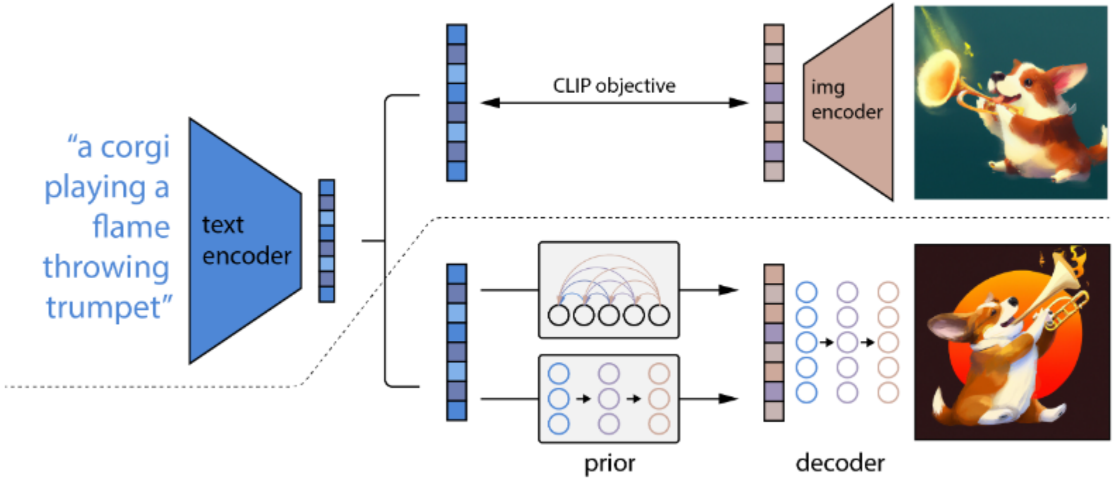
Antoine de Saint-Exupéry
安东尼·德·圣-埃克苏佩里

Parameter Number is Huge

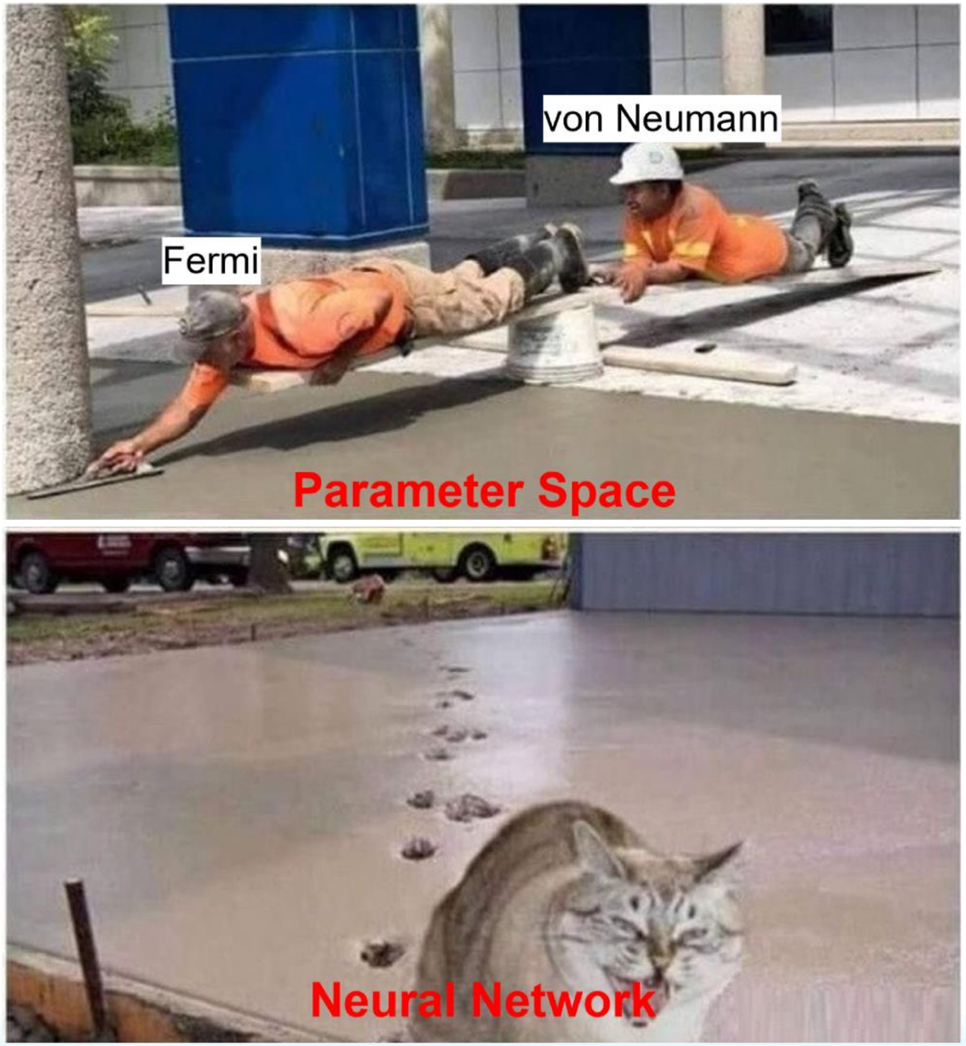
- Modern NN is wide & **deep**
  - 1998: LeNet-5, 6×10^4
  - 2022: DALL-E 2, 3.5×10^9





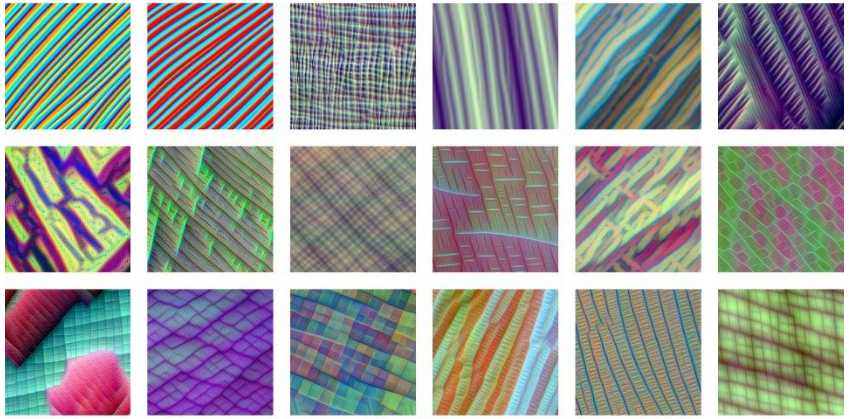


Possible Questions from Physicists

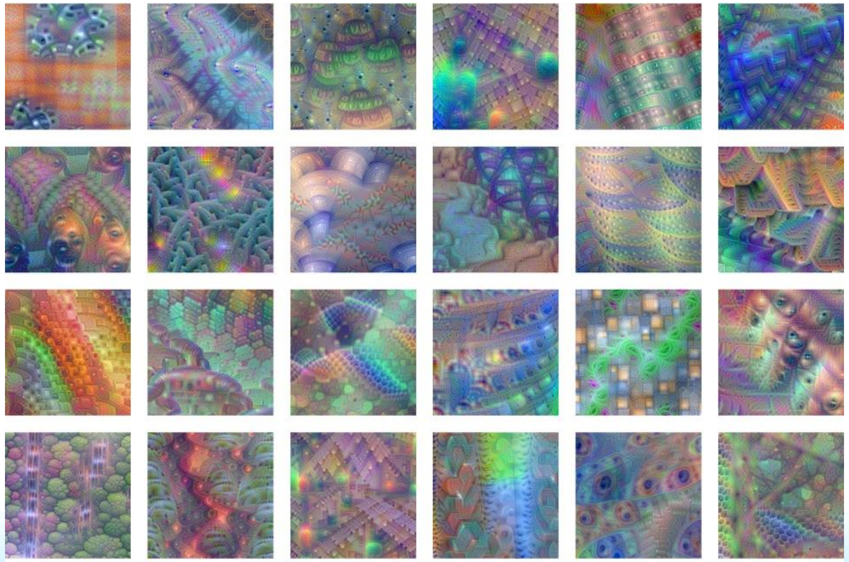


- 10^nparameters?! That's Toooooo Many!
    - Explainability/Interpretability, can be partially explained.
    - Personal comment at the end of this talk.

Possible Questions from Physicists

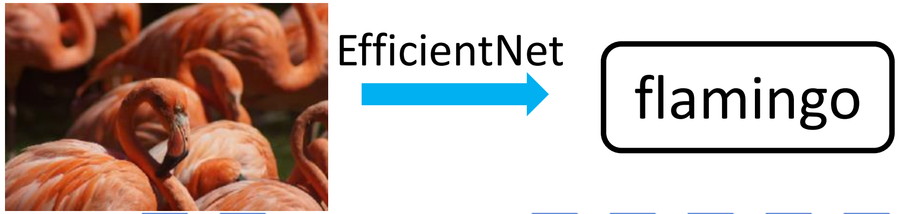


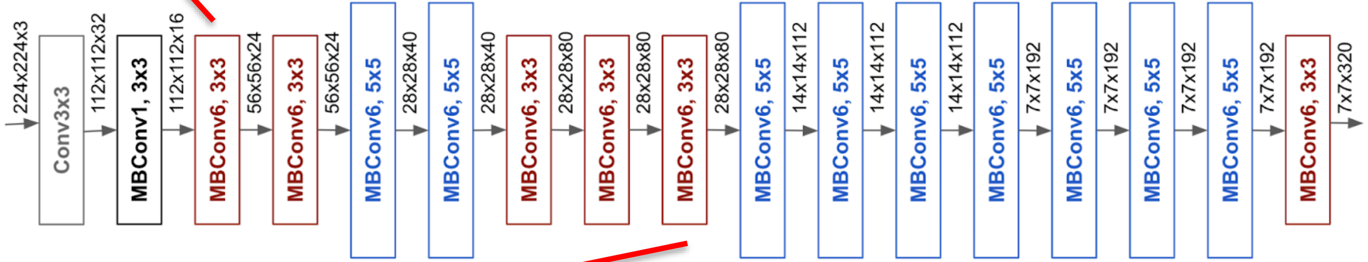
blocks 1a, 2a, 2b, 3a, 3b



blocks 5a, 5b, 5c

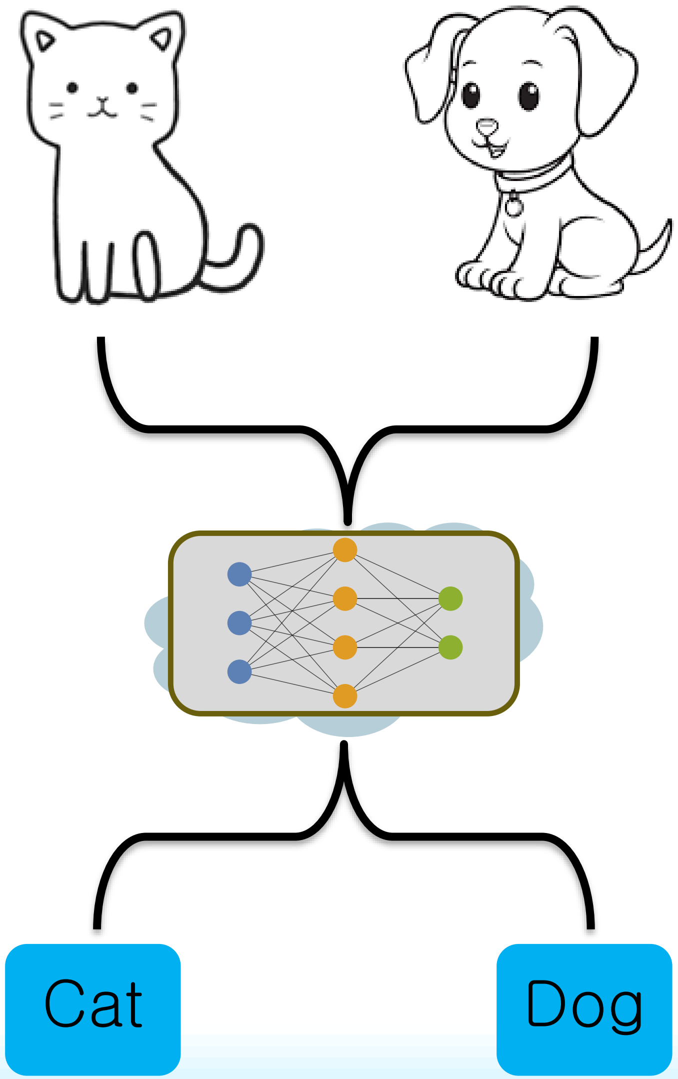
- 10^n parameters?! That's Toooooo Many!
  - Explainability/Interpretability, can be partially explained.
  - Personal comment at the end of this talk.
- Difference between NN and fitting?
  - Fundamentally the same but somehow not that trivial





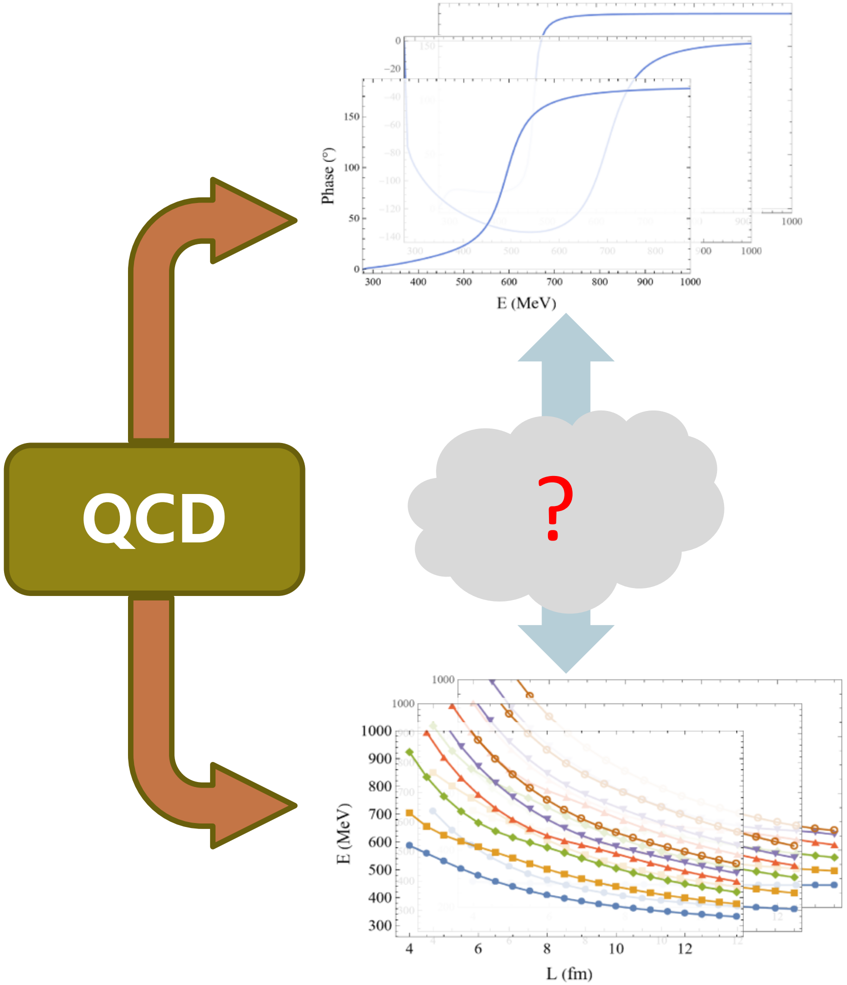
EfficientNet, arXiv:1905.11946

Possible Questions from Physicists



- 10^nparameters?! That's Toooooo Many!
  - Explainability/Interpretability, can be partially explained.
  - Personal comment at the end of this talk.
- Difference between NN and fitting?
  - Fundamentally the same but somehow not that trivial
- Why bother?
  - **Vague** idea becomes **solid**
  - In the spirit of Duck Test -> **NN≈Underline Function**

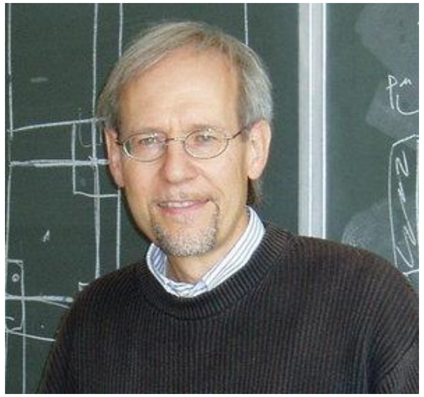
Motivation of Our Work



- QCD is hard
  - Phenomenological models/ ChiPT etc.
  - LQCD
- Is there a model-independent link between model-dependent quantities?
  - Hard/Impossible? Martin Lüscher did it in 1986
- One Workflow in LQCD
  - $\langle \hat{O}_i \hat{O}_j \rangle \rightarrow E(L) \rightarrow \delta(E)$

$$\delta(E) = \arctan\left(q \frac{\pi^{3/2}}{\mathcal{L}_{00}(1;q^2)} \right)$$

$$\mathcal{L}_{00}(1;q^2) \sim \frac{1}{\sqrt{4\pi}} \sum_{\vec{n}} \left(\vec{n}^2 - q^2\right)^{-1}$$



Commun. Math. Phys. 104, 177,
Commun. Math. Phys. 105, 153,
Nucl. Phys. B 354, 531

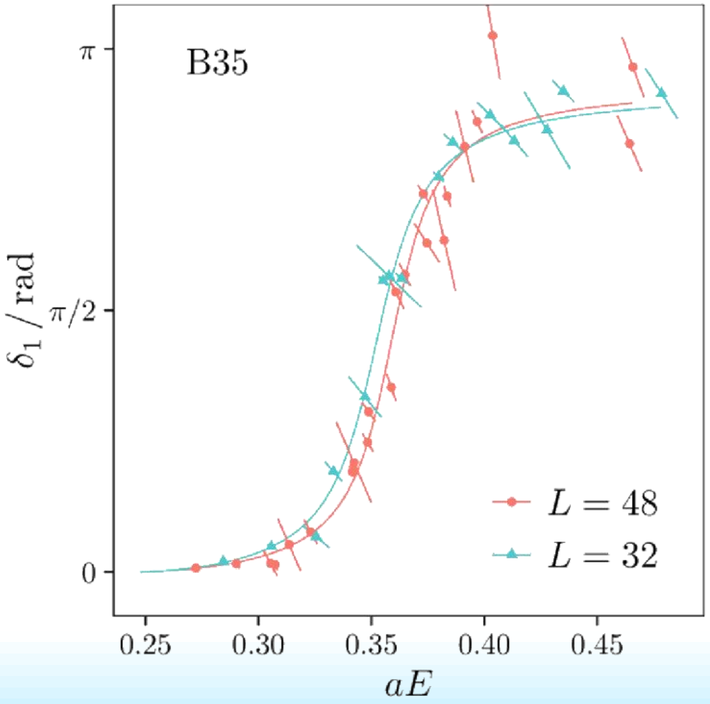


Fig. from *Eur. Phys. J. A 56, 61*

Hamiltonian Effective Field Theory(HEFT) & Data Generation

J.J. Wu *et al.* Phys.Rev.C.90.055206

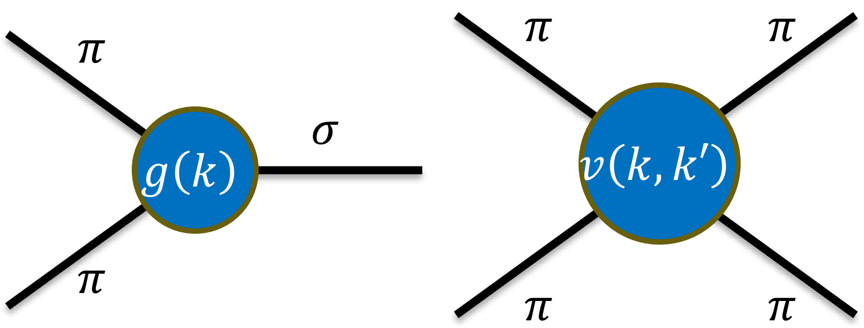
$\pi\pi \rightarrow \pi\pi$, s wave elastic scattering

$$H = H_0 + H_I$$

$$H_0 = |\sigma\rangle m_\sigma \langle\sigma| + \int d\boldsymbol{k} \left(|\boldsymbol{k}\rangle \sqrt{m_\pi^2 + k^2} \langle \boldsymbol{k}| \right)$$

$$H_I = \int d\boldsymbol{k} (|\boldsymbol{k}\rangle\, g(k) \langle\sigma| + h.c.) + \int d\boldsymbol{k} d\boldsymbol{k}' |\boldsymbol{k}\rangle\, v(k,k') \langle \boldsymbol{k}'|$$

$g(k) \propto f(k), v(k,k') \propto f^2(a,k) f^2(a,k')$

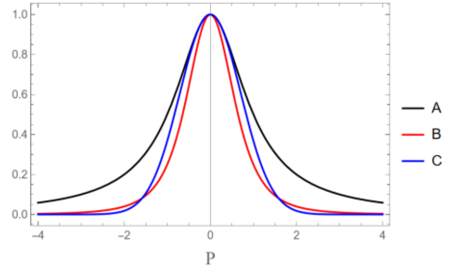


$f(a,k) =$

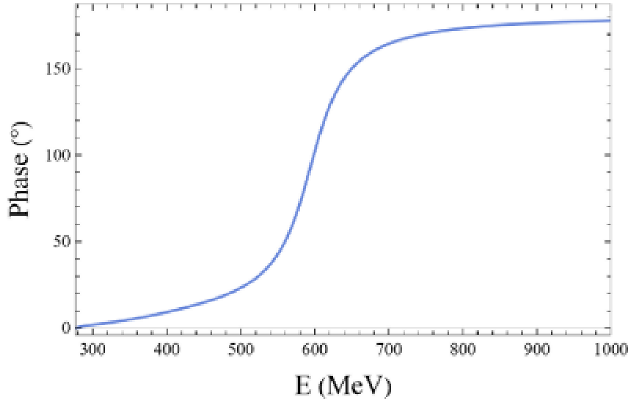
$A: \left(1+(ak)^2\right)^{-1}$

$B: \left(1+(ak)^2\right)^{-2},$

$C: \; e^{-(ak)^2}$

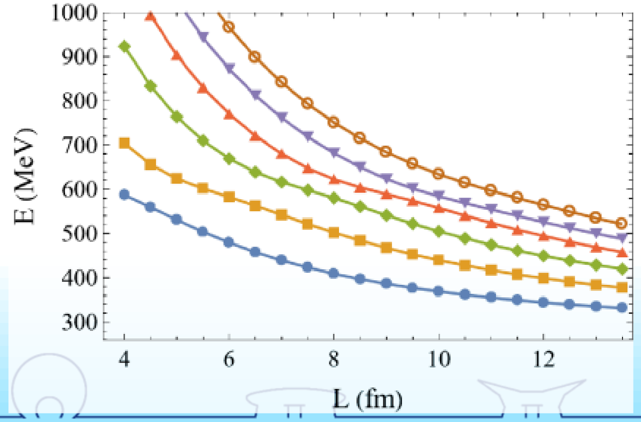


Lippmann-Schwinger equation $\rightarrow T \rightarrow \delta(E)$



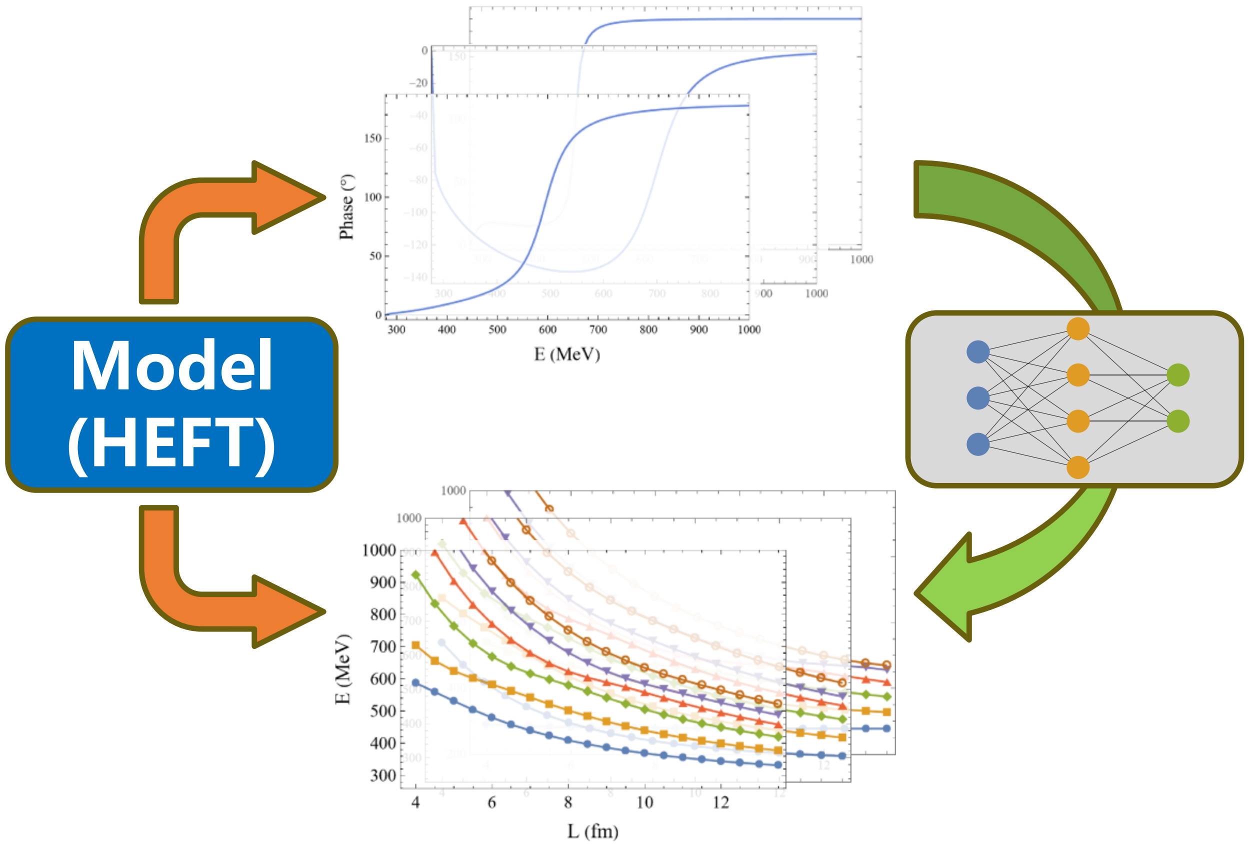
Discretization, Eigenfunction:

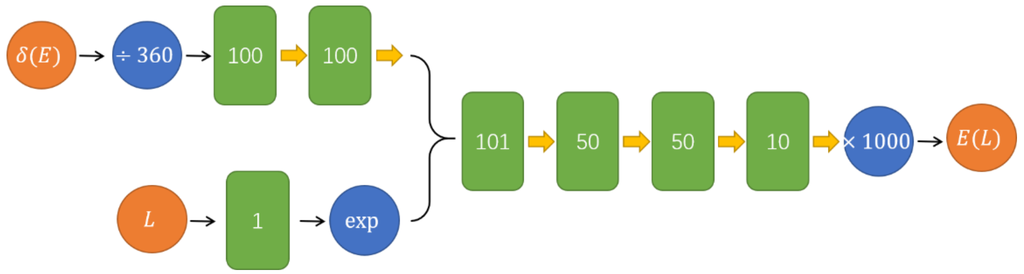
$H(L)|\psi\rangle = E(L)|\psi\rangle$



Hamiltonian Effective Field Theory(HEFT) & Data Generation

J.J. Wu *et al.* Phys.Rev.C.90.055206





- $\delta(E)$ contains the full information
- SoftPlus Not ReLU
- **Lowest** 10 Energy levels
- LossFunction: mean square error
- 2500 $\delta(E)$ for each model, batch siz e 10^4, 4*10^4 epoch

Model-dependent

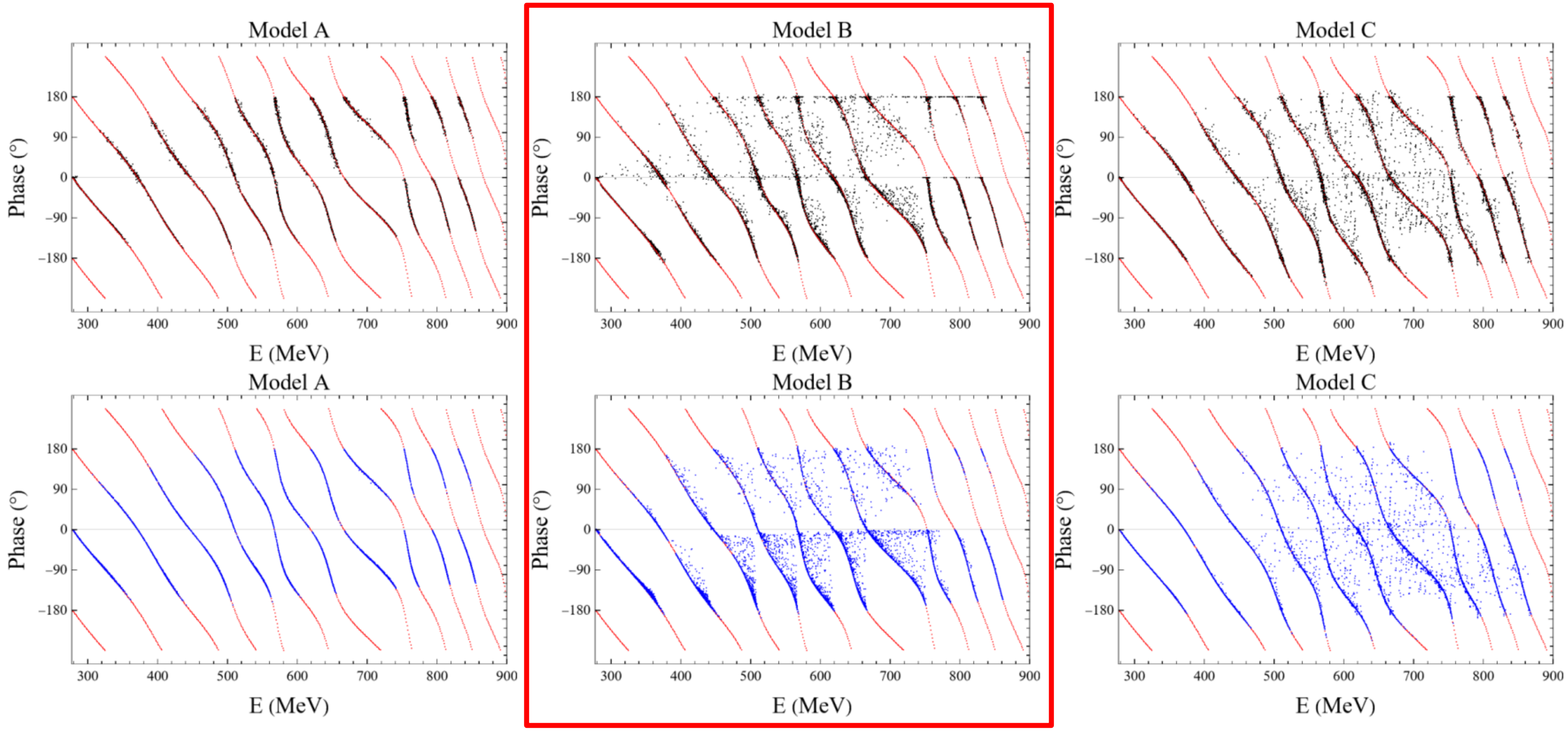
$$\text{LF} + o\left(e^{-mL}\right) = E(L)$$

Model-independent

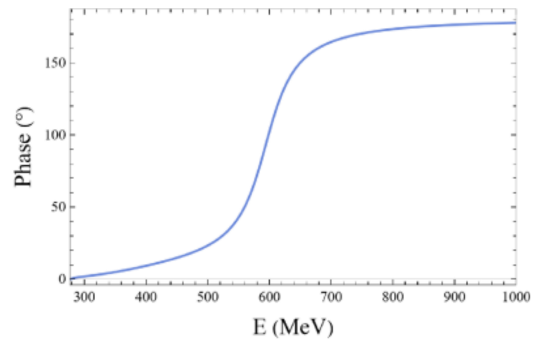
- Check $\delta(E_L)$ against LF
- Model B: NN tries to drag the points towards LF
- NN captures model-independent link (to some degree)

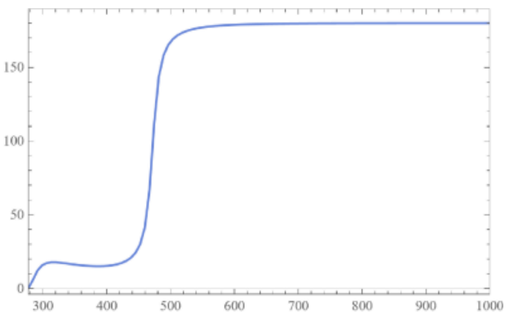
LF NN Model

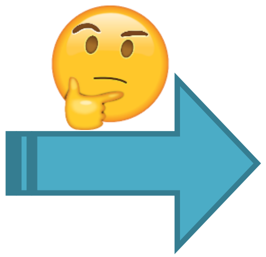
Stronger Evidence?

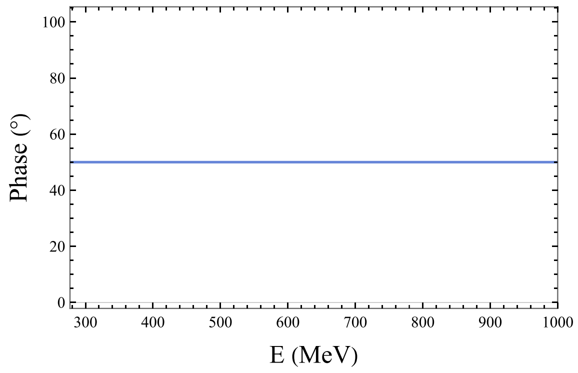


- Go Far beyond training set & challenge the NN with constant $\delta(E)$

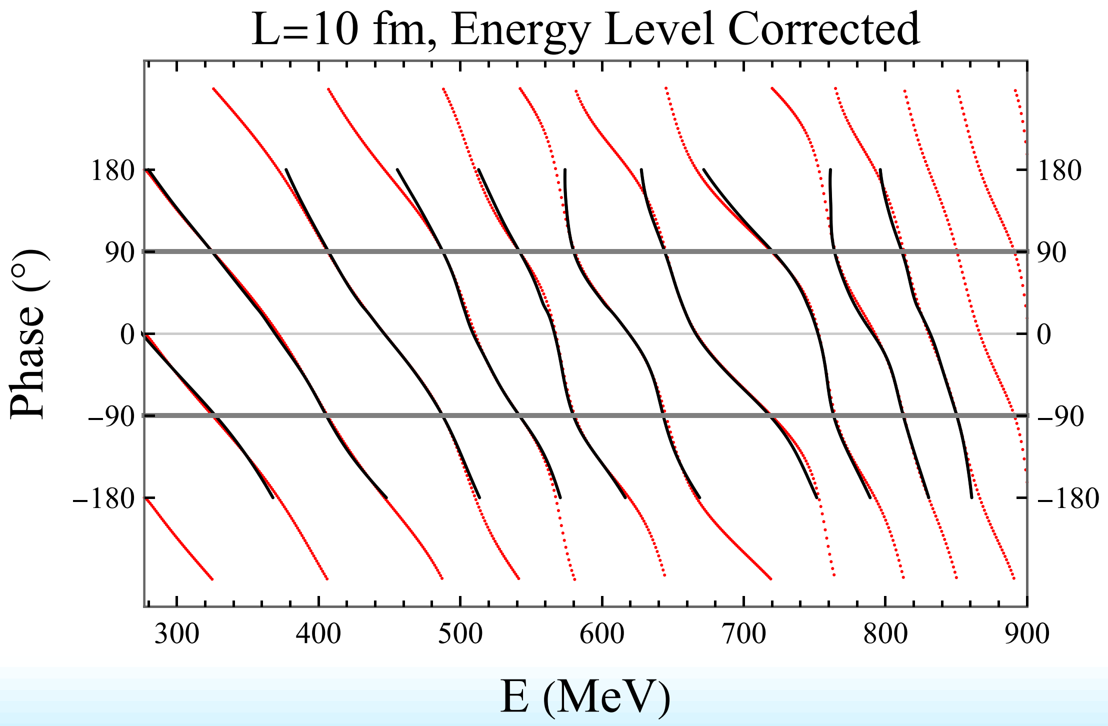






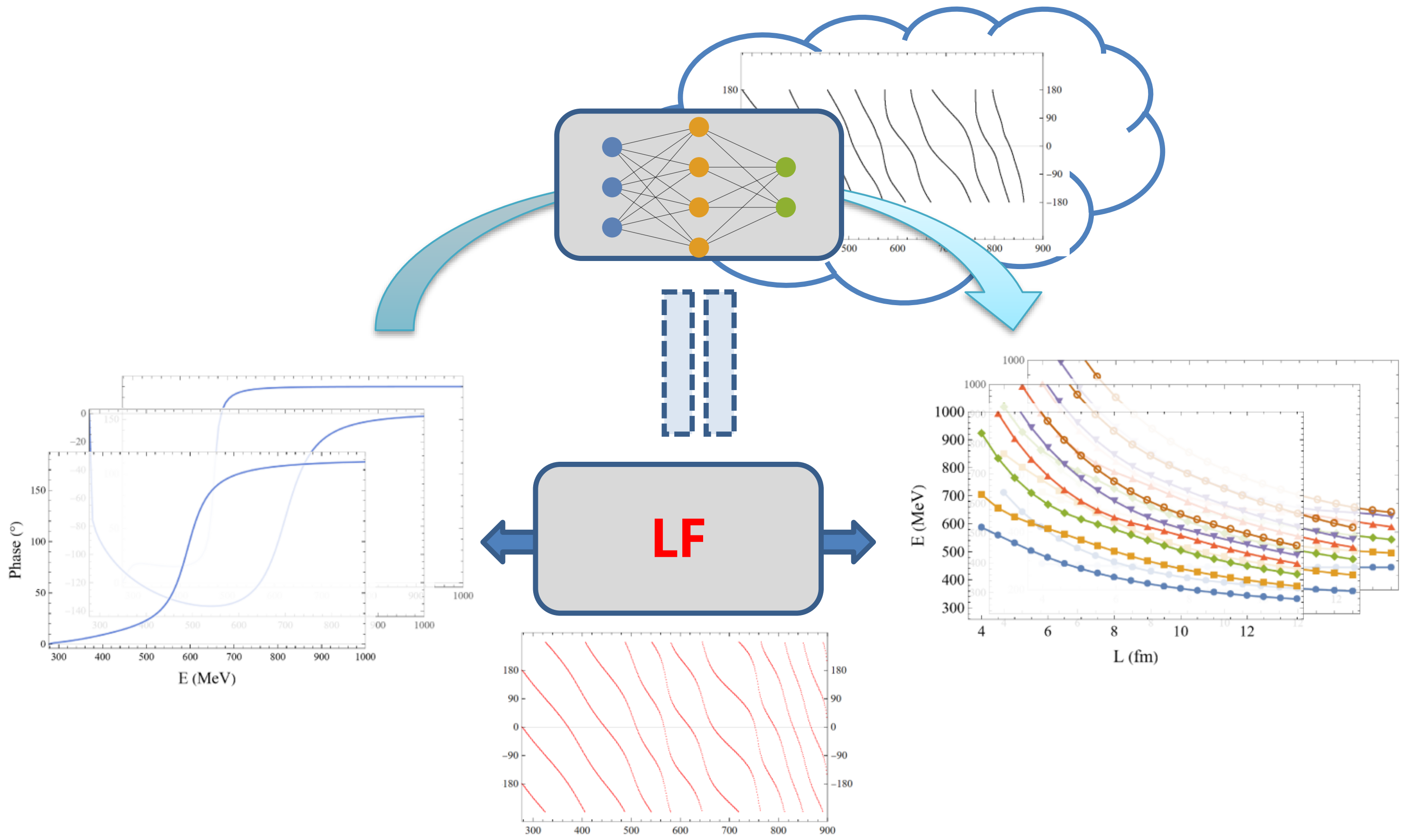


LF is Numerically Reproduced



What We Learned

Even $\delta(E), E(L)$ are both model-dependent,
NN can extract model-independent link (LF)



$$\text{LF} + o\left(e^{-mL}\right) \to E(L)$$

Model-dependent

Model-independent

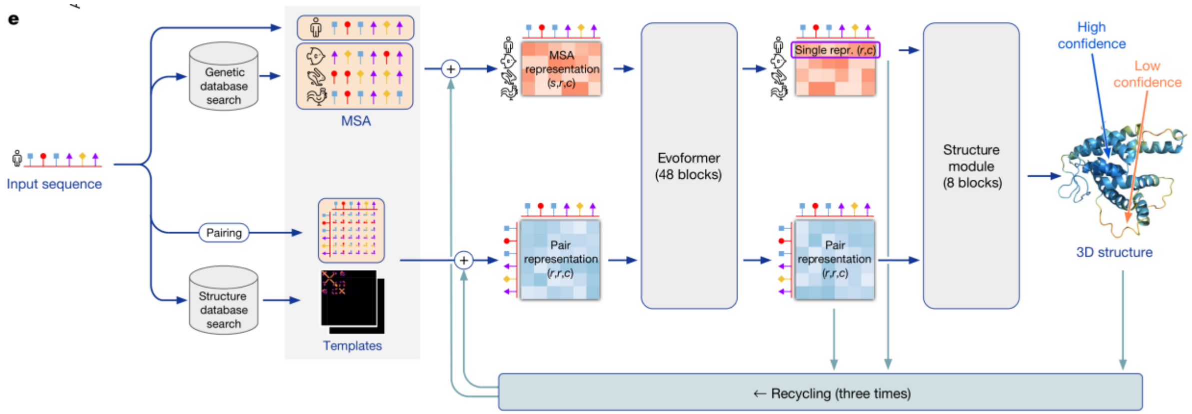
Where there is a link,
there is a NN :)

Personal Remarks

Essentially Optimization Challenges

Orthodox Way	
Yes	But
Beautiful Equation/Theory (SM, Einstein Equation)	Difficult to Solve
Perturbation Theory (ChiPT)	Limited Usage
Beyond Perturbation (LQCD)	Various Artifacts
…	…

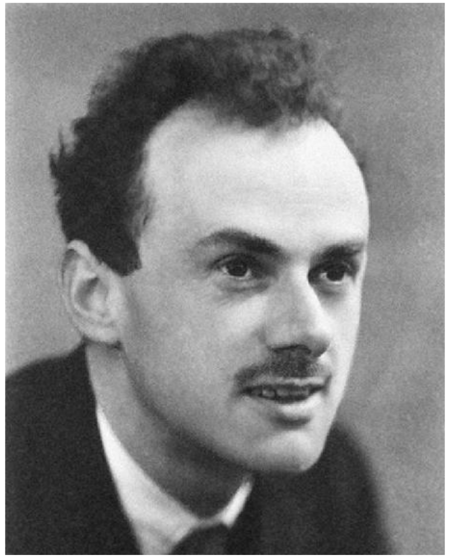
ML is a game changer
It peeks the correct answer!



AlphaFold2 [Nature 596, 583]

Huge Searching Space  Plenty Protein Data

Outlook



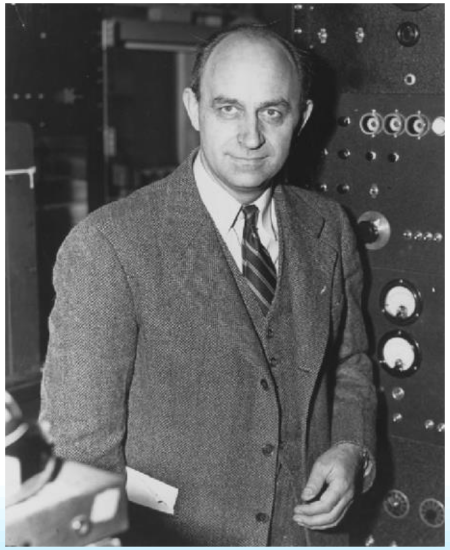
Sensible mathematics involves neglecting a quantity when it is small,

not neglecting it just because it is infinitely great and you do not want it!

➡ ★Renormalization

⬆

Schwinger, Feynman, Tomonaga; Wilson, ‘t Hooft, …



With four parameters I can fit an elephant, and with five I can make him wiggle his trunk

➡ More is different? 🤔

⬇

P.W. Anderson, J. Hinton & **You audience** 🤩

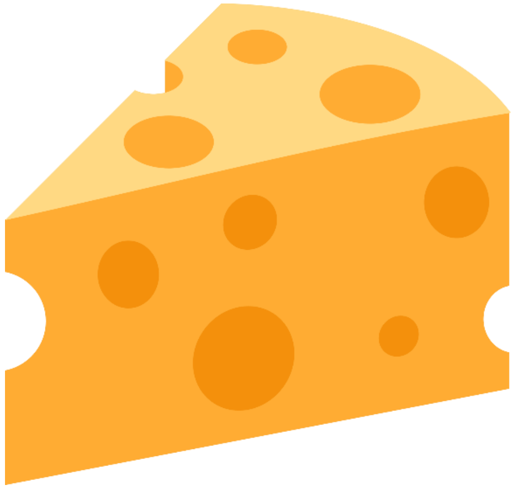
Thank you

Cheese Paradox

Bigger Cheese

Bigger Holes

Smaller Cheese



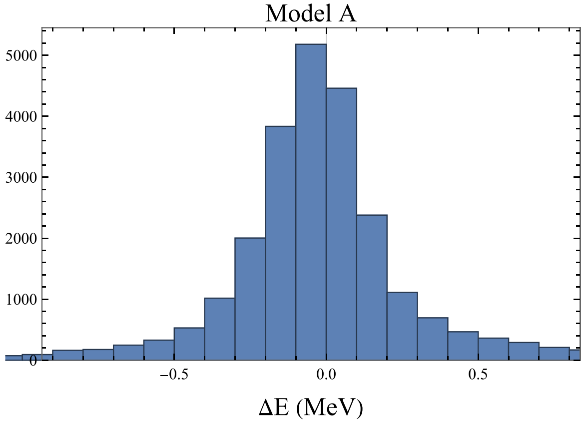
All models are wrong, but some are useful

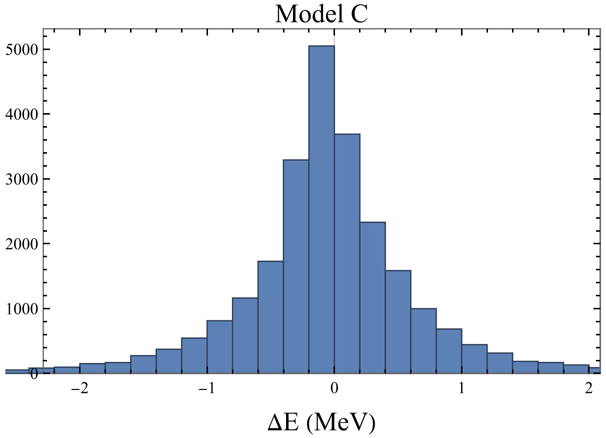
Huge Searching Space

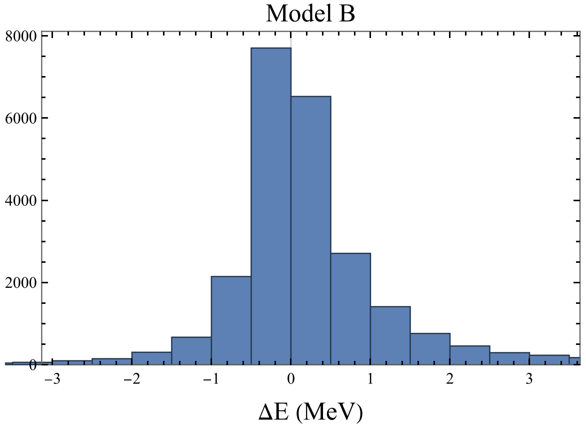
Plenty Protein Data

Orthodox Way		ML Way	
Yes	But	Yes	But
Beautiful Equation/Theory (SM, Einstein Equation)	Difficult to Solve	Simple Idea (Relatively)	Many GPU time To Train Resource Hungry
Perturbation Theory (ChiPT)	Limited Usage	It Works! (mostly)	Black Box + Adversarial Examples
Beyond Perturbation (LQCD)	Various Artifacts	Plenty of Data	Data Cleaning
…	…	…	…

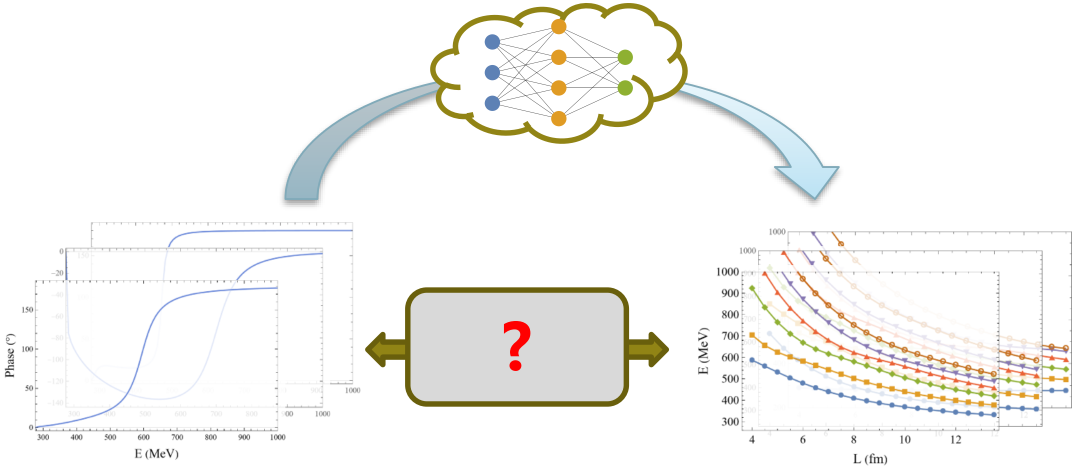
Result Analysis







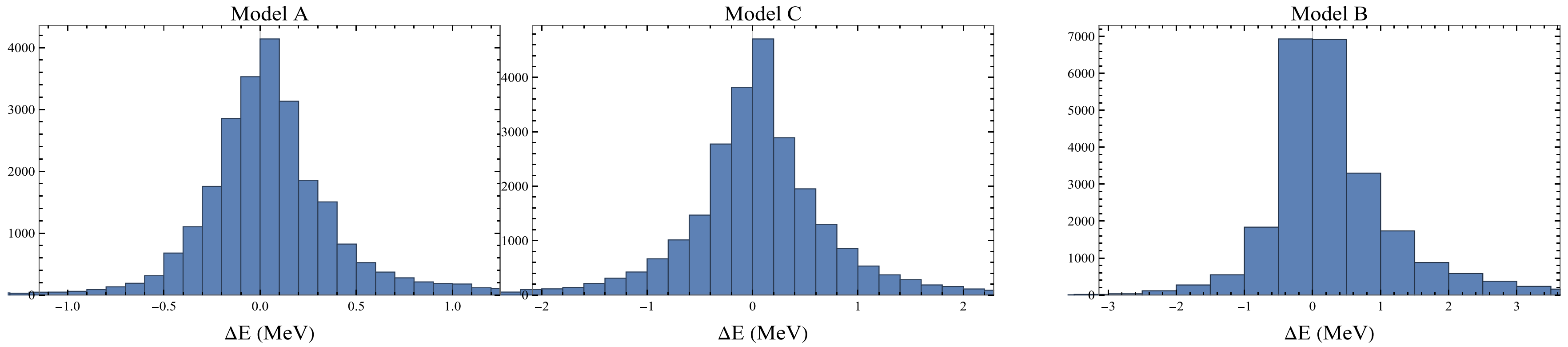
$$\Delta E \coloneqq E_{model} - E_{NN}$$

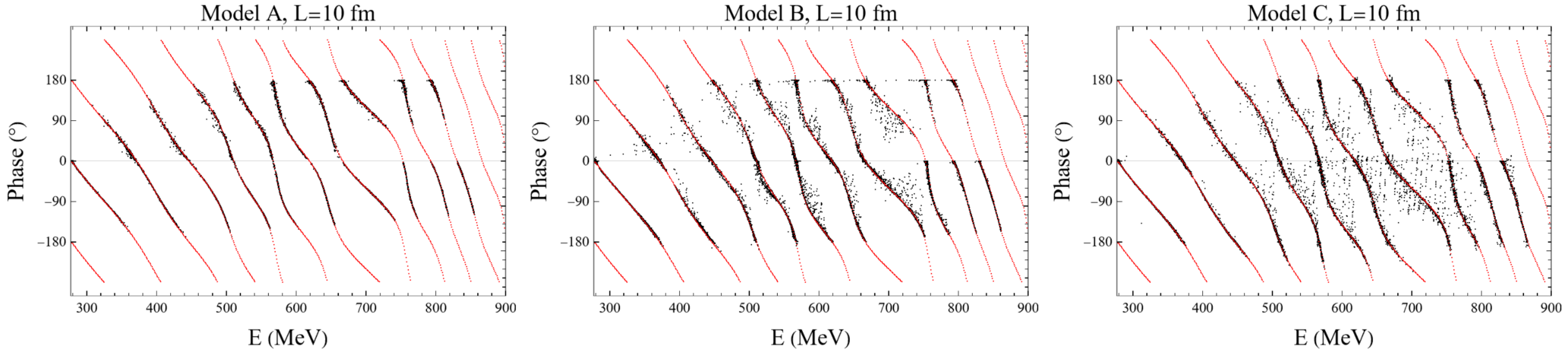


- Decently trained on model A, C
  - $\Delta E(L) < 1\text{MeV}, E(L) \sim 300 - 900$ MeV
- For model B,
  - as a test set, slightly worse
  - ΔE has heavier-tail on the right
- Signifies the existence of link
- Under the hood, LF is in charge
- -> Check against LF

BackUp

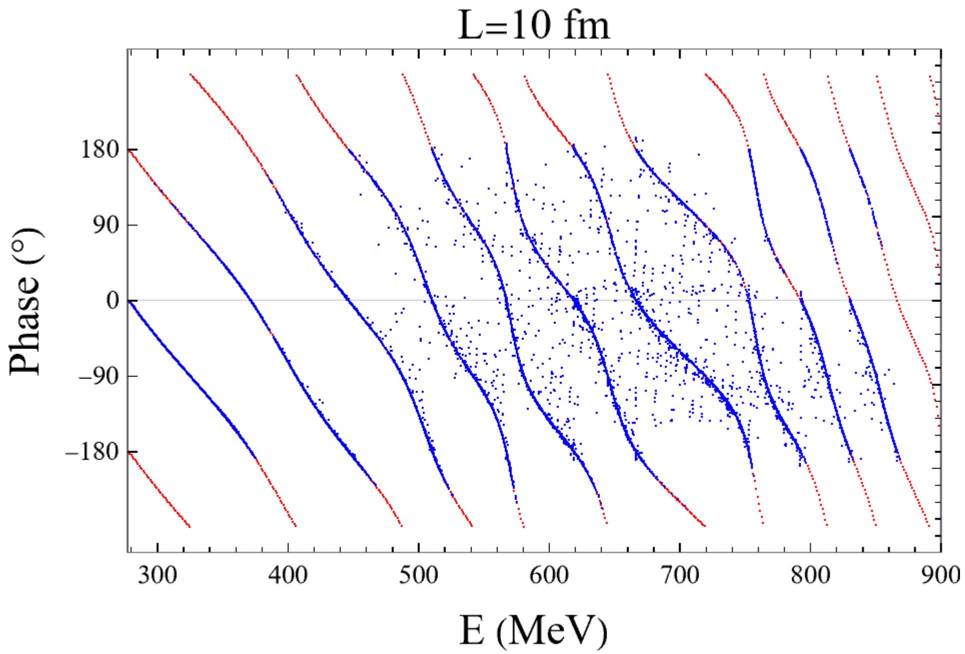
Results after level correction

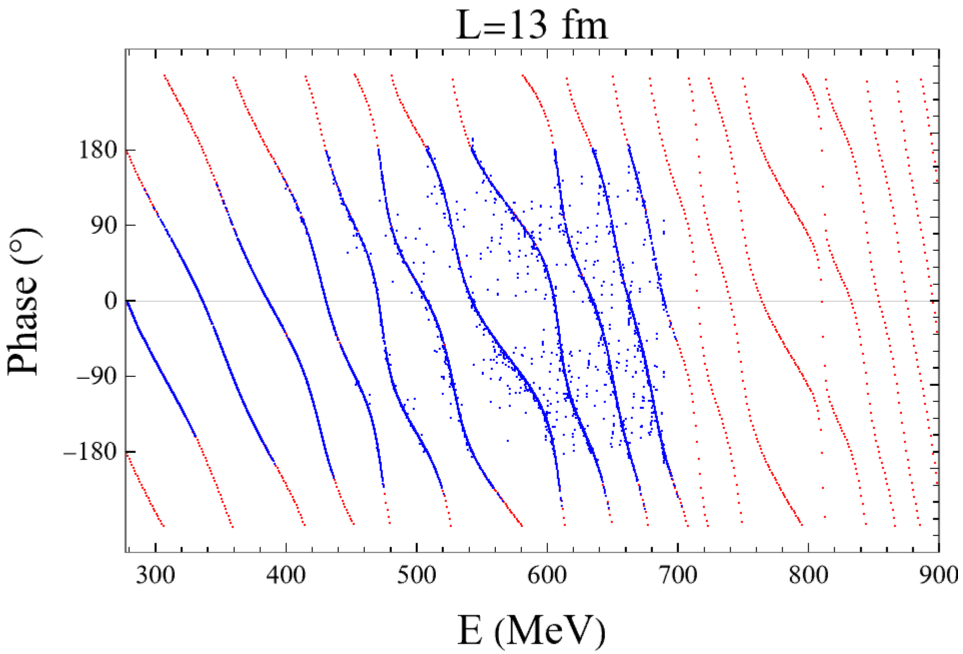




$\delta(E)$ is evenly sampled by 100 points in $[2m_\pi, 1\text{GeV}]$

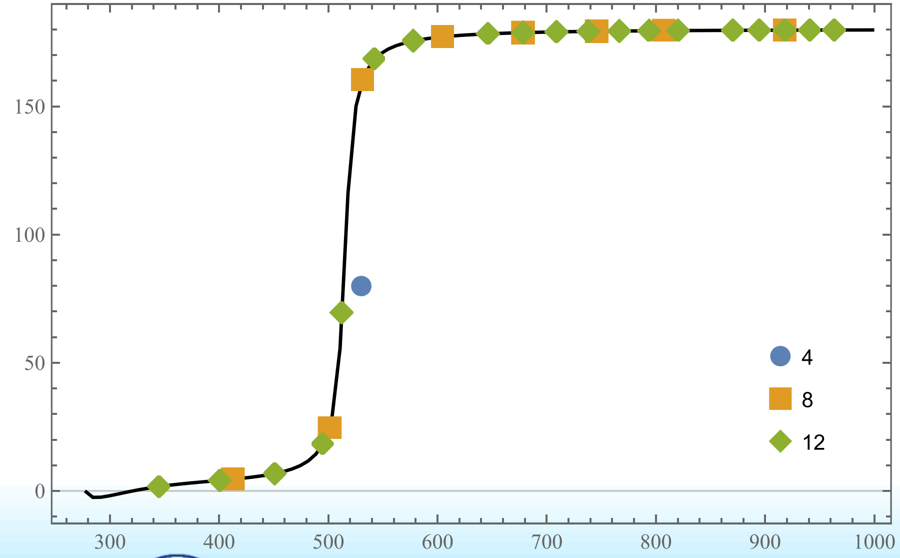
$m_\sigma \in [350,700], a \to c, d \in [0.5,2]$

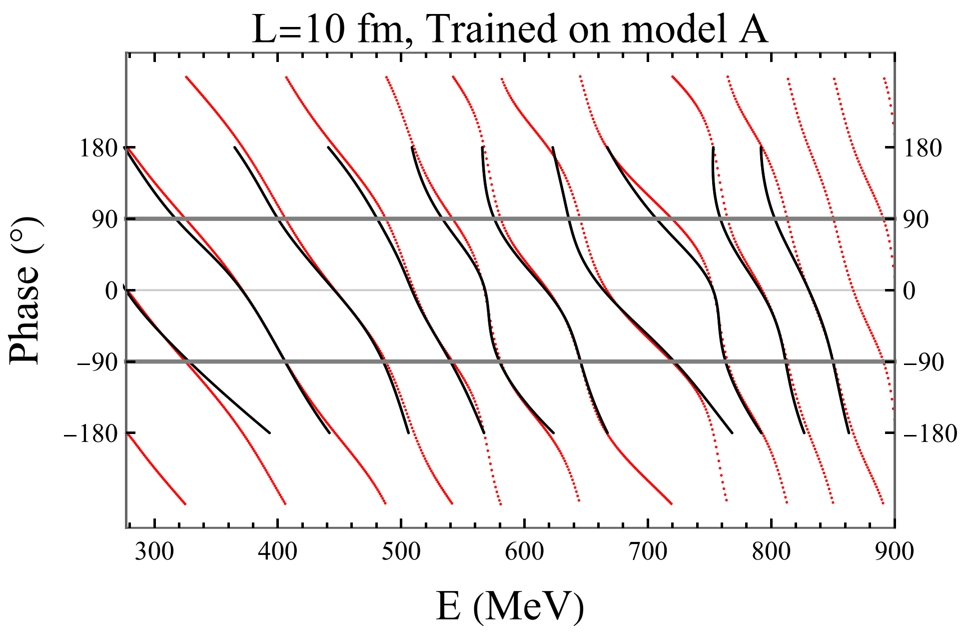




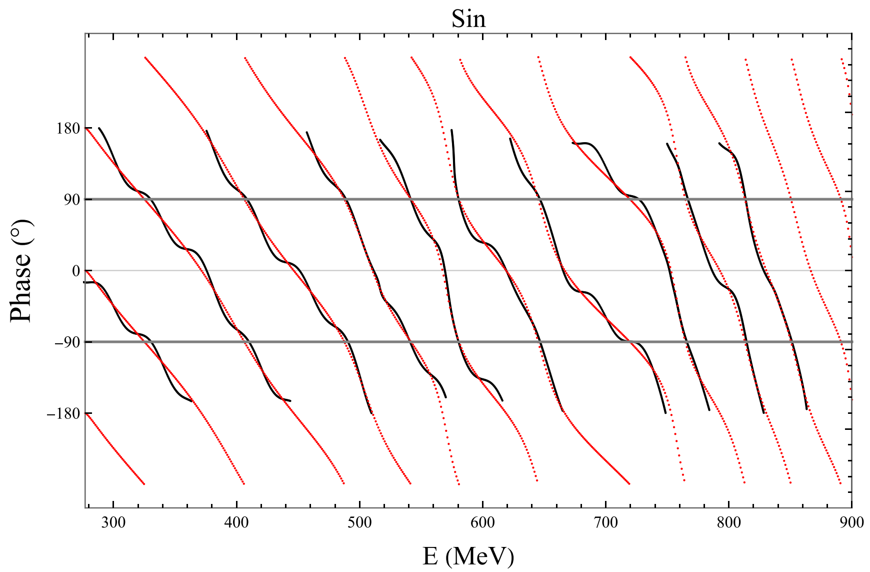
$$\delta(E) = \arctan\left(\mathrm{q}\frac{\pi^{3/2}}{\mathcal{L}_{00}(1;q^2)}\right) + n\pi$$

$$\mathcal{L}_{00}(1;q^2) = \frac{1}{\sqrt{4\pi}}\sum_{\vec{n}}\left(\vec{n}^2 - q^2\right)^{-1}$$





L=10 fm, Trained on model A
Phase (°)
E (MeV)



Sin
Phase (°)
E (MeV)