北京大學
PEKING UNIVERSITY

Machine Learning and its applications in nuclear physics

Junchen Pei (Peij@pku.edu.cn)

Contents

The Era of Big Data

- the era of Big Data—a term that refers to the explosion of available information
- massive amounts of very high-dimensional or unstructured data is continuously produced
- big data has been essentially changing and transforming the way we live, work, and think: *national development*, *industrial upgrades*, *scientific research*, *emerging interdisciplinary research*, *helping people better perceive the present* , *better predict the future*
- Examples: social media analysis, biomedical imaging, high-frequency finance, analysis of surveillance videos and retail sales, covid-19 analysis, US elections
- scientific advances are becoming more and more data-driven
- In big data research and applications, industry is ahead of academia.

- Infrastructure is developing very fast: computing, storage, communication of data

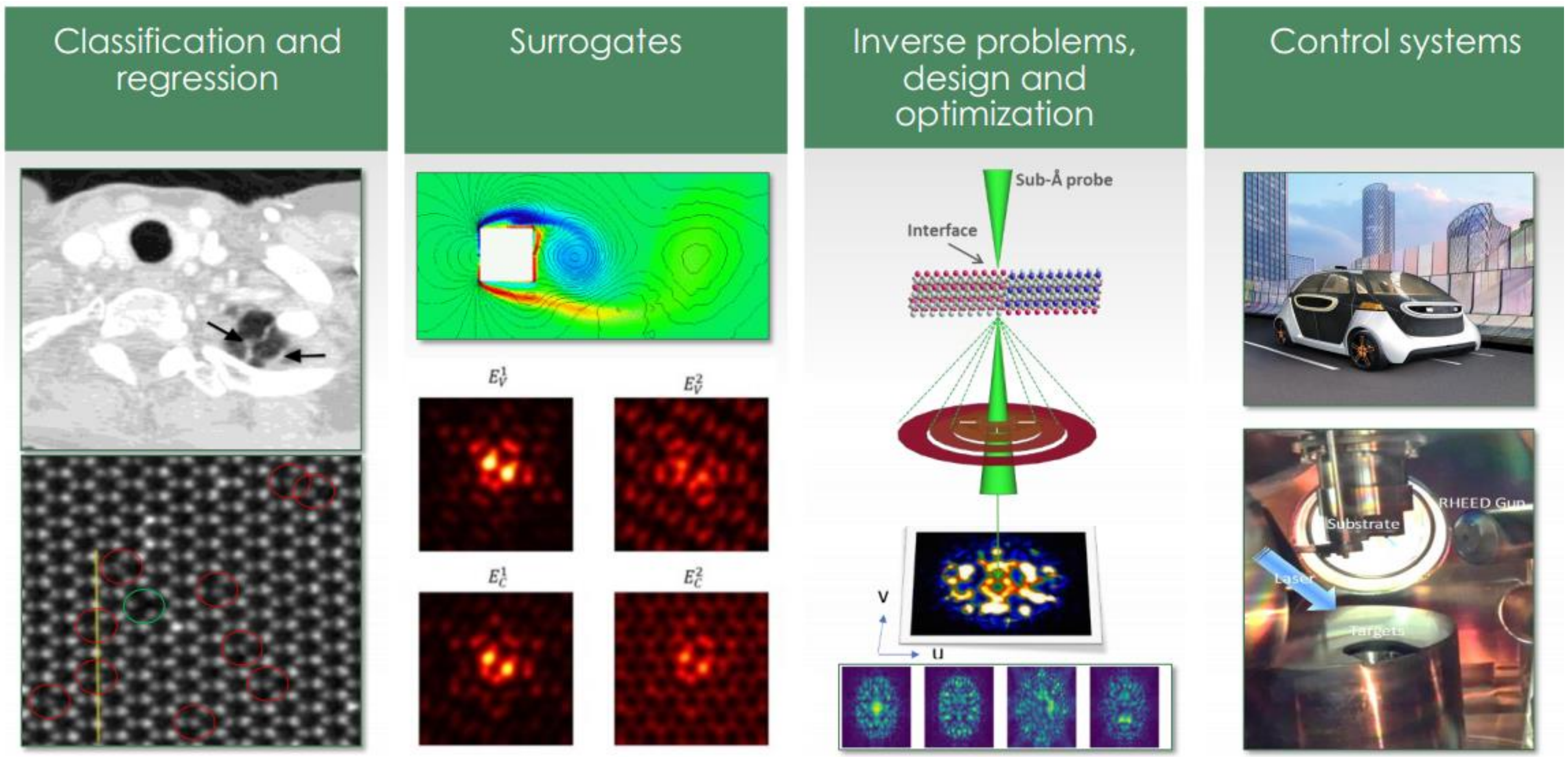
AI for science

- AI indeed is a game changer and will impact all aspects of our lives

  Alphago as a milestone (2016) , Alphafold2 (2020)

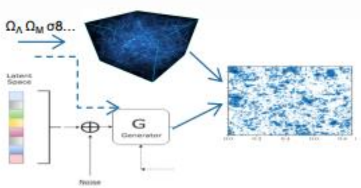
  AI won't replace the scientist, but scientists who use AI will replace those who don't.

- Applications: from Big data to information



Surrogates

- ML-created models
- Faster and/or higher fidelity models
- Generative networks
- Using ML to replace complicated physics
- Cosmology



Control

- ML-controlled experiments
- Efficient exploration of complex space
- Reinforcement Learning
- Use RL agent to control light source experiments
- Temperature control for Block Co-Polymer (BCP) experiments

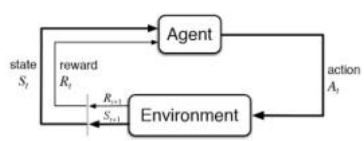
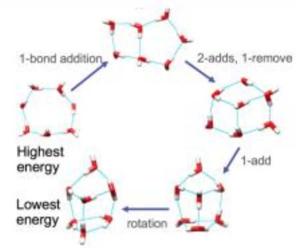


Image courtesy Sutton, Barto, Reinforcement Learning 2017

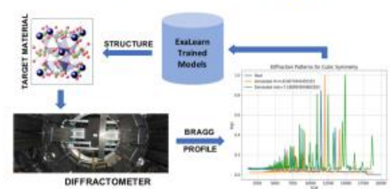
Design

- ML-created physical structures
- Optimized proposal for desired behavior of structure within complex design space
- Graph-Convnets
- Use Graph-CNN to propose new structures that respect chemistry
- Molecular Design



Inverse

- ML projection from observation to original form
- Back-out complex input structure from observed data
- Regression models
- Predicting crystal structure from light source imaging
- Material structure from neutron scattering

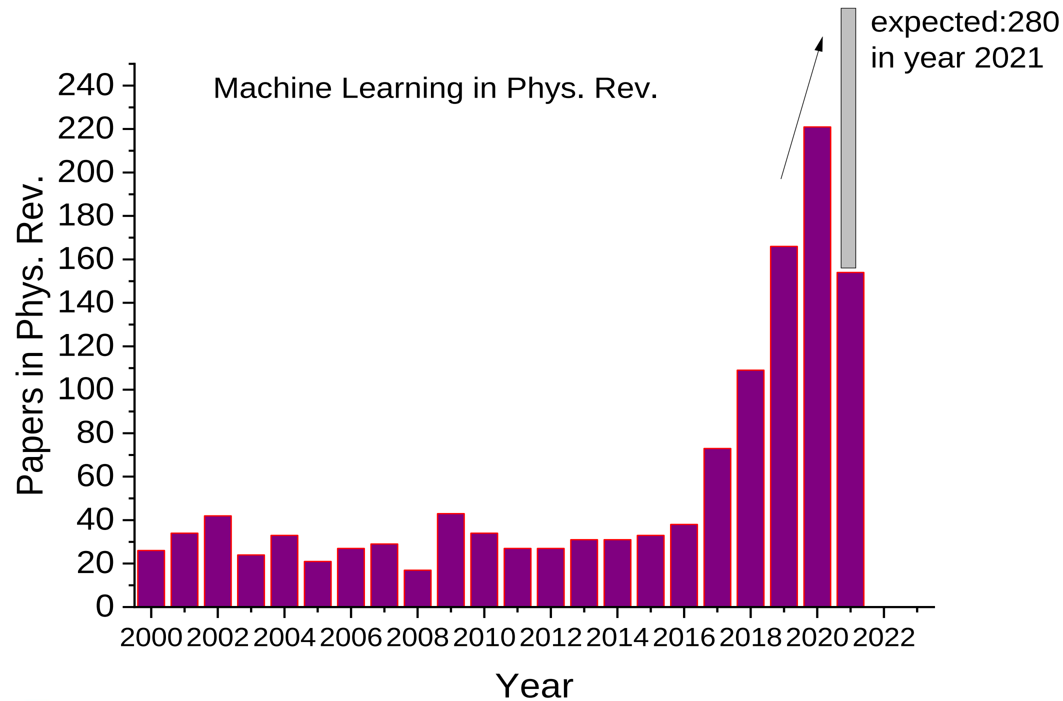


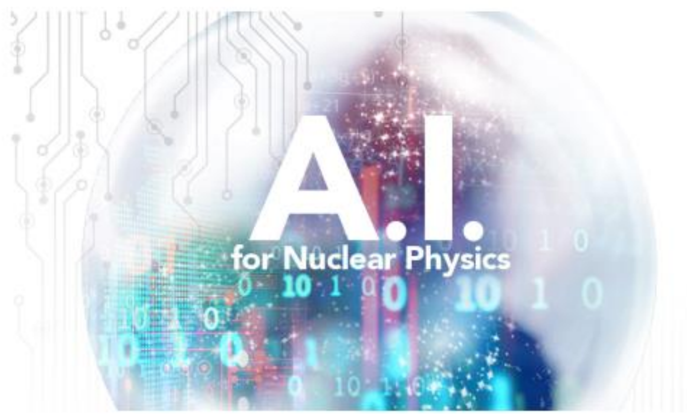
EXASCALE COMPUTING PROJECT

Machine Learning for Physics

Rapid developments in Physics:

ML papers in Phys. Rev.
Phys. Rev. Lett. (225)---2680
Phys. Rev. A (119)---2490
Phys. Rev. B (91)---5040
Phys. Rev. C (19)---991
Phys. Rev. D (110)---3696
Phys. Rev. E (608)---2037
1980-1990: 74
1990-2000: 265
2000-2010: 329
2010-2020: 674





AI for nuclear physics, arXiv:2006.05422, Jefferson Lab, March 4-6, 2020

AI for nuclear physics

- **Accelerator design and operations**
- **Holistic approach to experimentation**
- **Experiment design**
- **Improving simulation and analysis**

- Sign problem in LQCD:

  Extraction of physical observables:

  Propagator inversion in LQCD:

  Bayesian inference and global QCD analysis:

  Identifying rare events:

  Origin of elements:

  Nuclear fission:

  Quantified computations of heavy nuclei using realistic inter-nucleon forces:

  Discovering correlations and emergent phenomena:

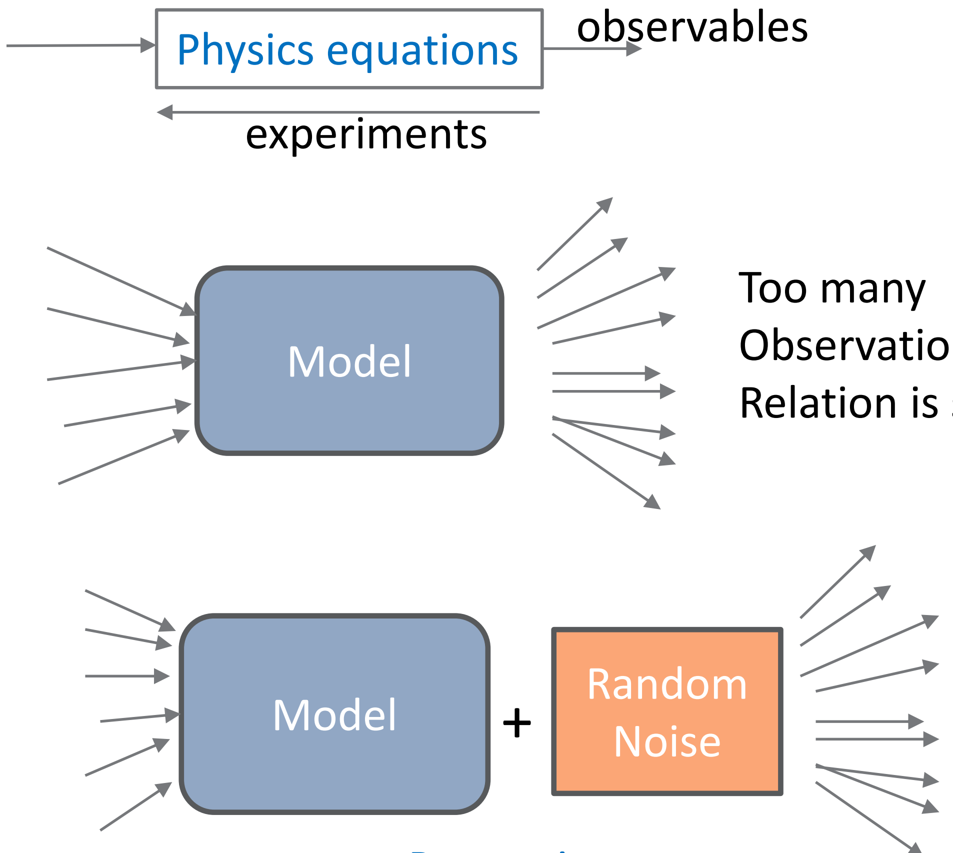
  Development of a spectroscopic-quality nuclear energy density functional:

  Neutron star and dense matter equation of state

AI for nuclear physics, arXiv:2006.05422, Jefferson Lab, March 4-6, 2020

Data science: regression

- 相同的目的：**prediction**



Physics model

Too many Observations.....
Relation is sparsely

Data model

More is different

Too many Observations.....

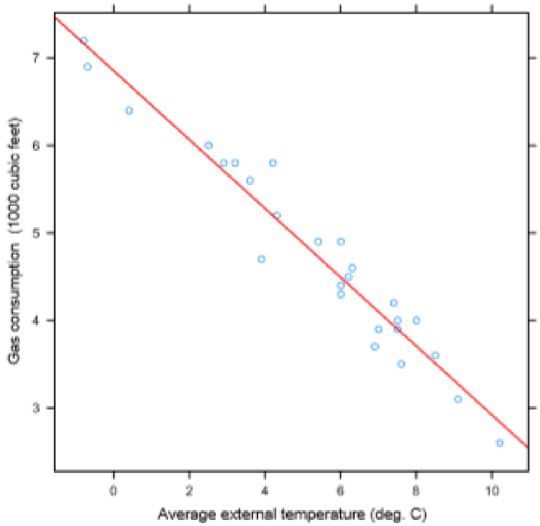
Regression

ill-Inverse, learn from data → infer + Uncertainty quantification

Future: hybrid physics + machine learning is expected

Linear regression

- Prediction is also referred to as regression.



$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

$$r(x) := E(Y \mid X = x) = \int y\, p(y|x) dx$$

$$Y = \beta_0 + \beta_1 x + \epsilon, \quad E(\epsilon|x) = 0, V(\epsilon|x) = \sigma^2$$

β_0 **intercept**
β_1 **slope**
σ^2 **variance**

$$\hat{\epsilon}_i := y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$$

$$\sum_{i=1}^{n} \hat{\epsilon}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

最小二乘法确定参数

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\hat{\sigma}^2 = \frac{1}{n-2} \sum_{i=1}^{n} \epsilon_i^2$$

线性回归不仅仅是线性拟合，还有误差可信度分析，是机器学习中的一个重要概念

$$Y = \beta_0 + \beta_1 x + \epsilon, \quad E(\epsilon|x) = 0, V(\epsilon|x) = \sigma^2$$

$$\epsilon|x \sim \mathcal{N}(0, \sigma^2)$$ 残余值分布-高斯形式

$$Y \sim \mathcal{N}(\beta_0 + \beta_1 X, \sigma^2)$$ 预测值的分布

θ应是一切可能取值中使 P(D|θ)达到最大的那一个

$$L_{\mathcal{D}}(\theta) := \prod_{i=1}^{n} \hat{p}(x_i, y_i \,|\, \theta)$$

似然函数L_D

p(X,Y |θ) be a joint probability density function for X and Y with parameters θ

maximum likelihood estimates (MLE)：**最大似然法确定参数**

$$L_{\mathcal{D}}(\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2) := \prod_{i=1}^{n} \hat{p}(x_i, y_i) = \prod_{i=1}^{n} \hat{p}(y_i \,|\, x_i) p(x_i) = \prod_{i=1}^{n} \hat{p}(y_i \,|\, x_i) \prod_{i=1}^{n} p(x_i)$$

$$\prod_{i=1}^{n} \hat{p}(y_i \,|\, x_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\hat{\sigma}} e^{-\frac{(y_i - \hat{y}_i)^2}{2\hat{\sigma}^2}} = \frac{1}{\sqrt{2\pi}^n \hat{\sigma}^n} e^{\frac{1}{-2\hat{\sigma}^2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

为正态分布时：最大似然法求一阶导数等同于最小二乘法

- 当有一系列变量时：

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \epsilon$$
$$= \langle \beta, X \rangle + \epsilon$$

$$\beta := \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad X := \begin{pmatrix} 1 \\ X_1 \\ \vdots \\ X_p \end{pmatrix},$$

X为长方矩阵
N行(1+p)列

$$\mathbf{X}^T\mathbf{X}\hat{\beta} = \mathbf{X}^T\mathbf{Y}.$$

求解线性方程组→β

SVD也可以求解

$$\hat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{n} \hat{\epsilon}_i^2 = \frac{1}{n-p}\sum_{i=1}^{n} (y - \hat{y})^2$$

P个变量，n个数据的方差

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

$$\hat{\beta} \sim \mathcal{N}(\beta, (X^T X)^{-1}\sigma^2)$$

$$V(\hat{\beta}) = (X^T X)^{-1}\sigma^2$$

Bayesian linear regression中参数也是分布项

多项式回归也可以依此类推,构建X矩阵

Gauss Process Regression

- Gauss process: 随机变量之间关系服从高斯分布，局域光滑nonlinear regression, with error bars, “smoothing devices”

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big).$$

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})],$$
$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))],$$

$$p(f|\mathcal{D}) = \frac{p(f)p(\mathcal{D}|f)}{p(\mathcal{D})}$$

$$k(x, x') = \sigma_f^2 \exp\left[\frac{-(x - x')^2}{2l^2}\right],$$ Kernel

$$y = f(x) + \mathcal{N}(0, \sigma_n^2),$$ 不用明确f(x)的形式，**very useful**

$$k(x, x') = \sigma_f^2 \exp\left[\frac{-(x - x')^2}{2l^2}\right] + \sigma_n^2 \delta(x, x'),$$

可以改进kernel函数

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix}$$

协方差矩阵

$$K_* = \begin{bmatrix} k(x_*, x_1) & k(x_*, x_2) & \cdots & k(x_*, x_n) \end{bmatrix} \qquad K_{**} = k(x_*, x_*).$$

- 预测点（x_*,y_*）

any finite subset of the range follows a multivariate Gaussian distribution

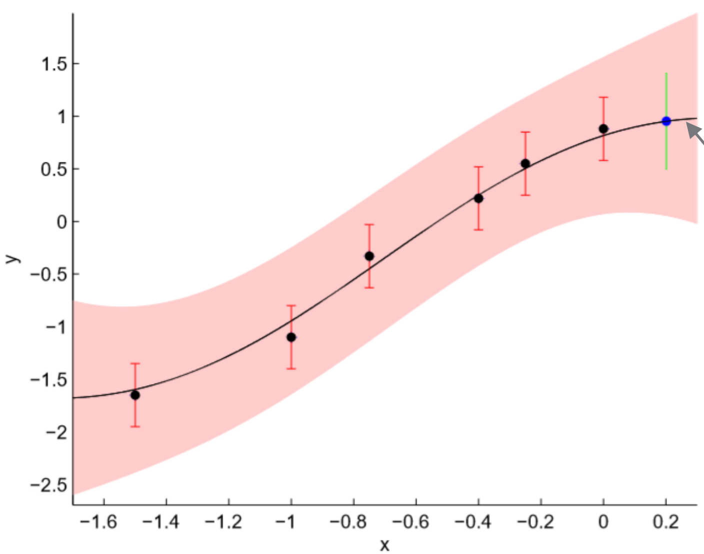
$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K & K_*^{\mathrm{T}} \\ K_* & K_{**} \end{bmatrix}\right),$$

Y的分布一致性

预测点x_*

$$y_*|\mathbf{y} \sim \mathcal{N}(K_*K^{-1}\mathbf{y},\ K_{**} - K_*K^{-1}K_*^{\mathrm{T}}).$$

$$\overline{y}_* = K_*K^{-1}\mathbf{y}, \qquad \mathrm{var}(y_*) = K_{**} - K_*K^{-1}K_*^{\mathrm{T}}.$$



X_*

maximum a posteriori $\log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$

得到高斯参数 σ_f l

hyperparameters

$$\log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^{\mathrm{T}}K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi.$$

$$\boldsymbol{\theta} = \{l, \sigma_f, \sigma_n\}$$

→ l = 1 and σ_f = 1.27 $\quad \sigma_n = 0.3$

- 例：

$$\mathbf{x} = \begin{bmatrix} -1.50 & -1.00 & -0.75 & -0.40 & -0.25 & 0.00 \end{bmatrix}.$$

预测点X*=0.2

$$K = \begin{bmatrix} 1.70 & 1.42 & 1.21 & 0.87 & 0.72 & 0.51 \\ 1.42 & 1.70 & 1.56 & 1.34 & 1.21 & 0.97 \\ 1.21 & 1.56 & 1.70 & 1.51 & 1.42 & 1.21 \\ 0.87 & 1.34 & 1.51 & 1.70 & 1.59 & 1.48 \\ 0.72 & 1.21 & 1.42 & 1.59 & 1.70 & 1.56 \\ 0.51 & 0.97 & 1.21 & 1.48 & 1.56 & 1.70 \end{bmatrix}$$

对角项为 $\sigma_f^2 + \sigma_n^2$

$\sigma_n = 0.3$

$$K_* = \begin{bmatrix} 0.31 & 0.68 & 0.92 & 1.25 & 1.38 & 1.54 \end{bmatrix}.$$

$K_{**} = 1.70$

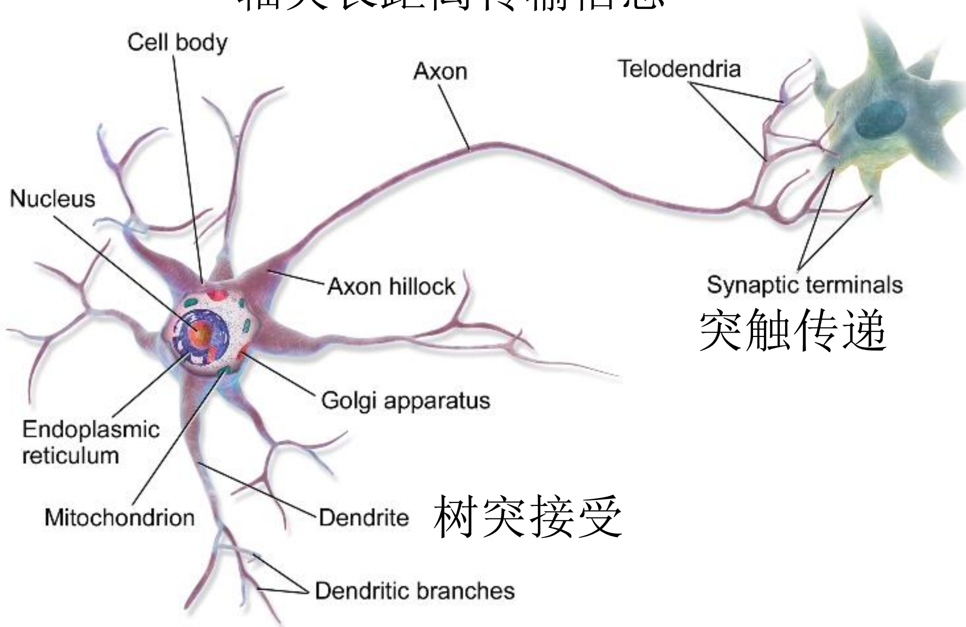
$\overline{y}_* = 0.95$ $\overline{y}_* = K_* K^{-1} \mathbf{y},$

$\text{var}(y_*) = 0.21$ $\text{var}(y_*) = K_{**} - K_* K^{-1} K_*^{\text{T}}.$

ftp://ftp.tuebingen.mpg.de/pub/ebio/chrisd/GPtutorial.pdf

生物神经元

轴突长距离传输信息



突触传递

树突接受

感知、记忆和思考

处理化学信号

细胞本体（soma）
树突（dendrites）
轴突丘（hillock）
轴突（axon）
突触（synapse）

- our brain contains about 86 billion neurons and more than a 100 trillion (or according to some estimates 1000 trillion) synapses (connections).
- In biological networks, neurons can fire asynchronously in parallel, have small-world nature with a small portion of highly connected neurons (hubs) and a large amount of lesser connected ones (the degree distribution at least partly follows the power-law).

- certain biological neurons can fire around 200 times a second on average. Signals travel at different speeds depending on the type of the nerve impulse, ranging from 0.61 m/s up to 119 m/s.

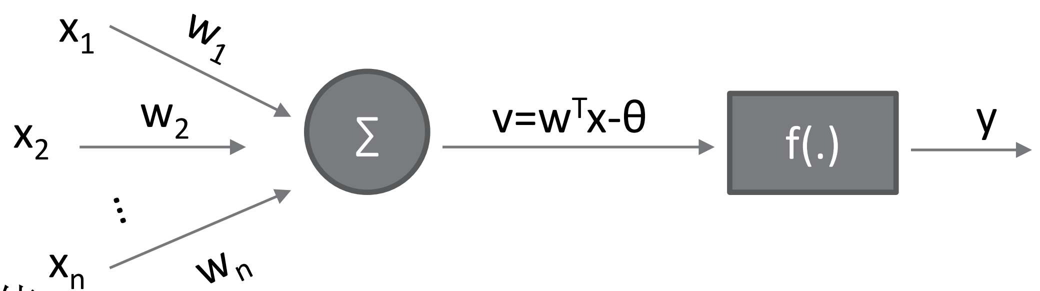
- biological neuron networks due to their topology are also fault-tolerant. Information is stored redundantly so minor failures will not result in memory loss.

- the brain consumes about 20% of all the human body's energy — despite it's large cut, an adult brain operates on about 20 watts (barely enough to dimly light a bulb) being extremely efficient. vs. 250w of GPU

- accepting binary inputs, applying weights to them and generating binary outputs depending on whether the sum of these weighted inputs have reached a certain threshold (also called a step function)

- Brain fibers grow and reach out to connect to other neurons, neuroplasticity allows new connections to be created or areas to move and change function

人工神经元

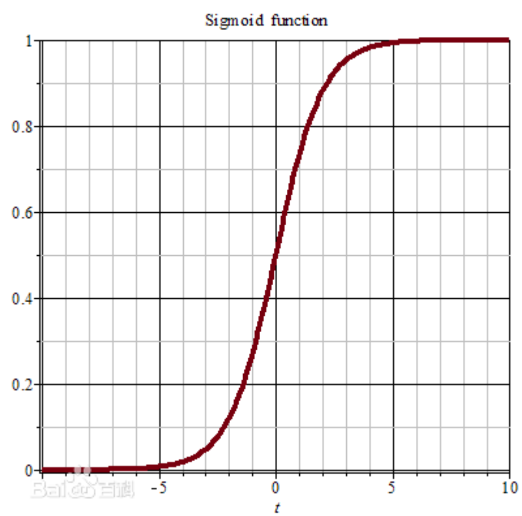


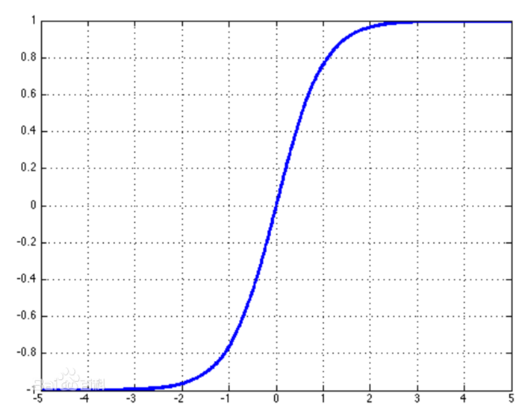
- 连接权值w
- 求和单元∑， θ为阈值
- 激活函数 f

一般为非线性映射，可以求导

Sigmoid 函数 $S(x) = \dfrac{1}{1+e^{-x}}$

$$tanhx = \frac{sinhx}{coshx} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$





Sigmoid函数收敛缓慢

Tanh函数实际效果可能会更好

Machine Learning 机器学习

- 监督学习：主要分为regression 和 classification（人脸识别，语音识别等等），前者输出的是连续变量，后者是离散变量

- 非监督学习：对数据不标注，寻找隐藏的结构和特征

  generative adversarial network 生成模型和判别模型的互相博弈学习（无监督）

- 强化学习：非监督学习，结果反馈
- 迁移学习Transfer Learning：训练的模型被重新应用

- 机器学习已广泛应用于数据挖掘、计算机视觉、自然语言处理、生物特征识别、搜索引擎、医学诊断、检测信用卡欺诈、证券市场分析、DNA序列测序、语音和手写识别、战略游戏和机器人等领域。

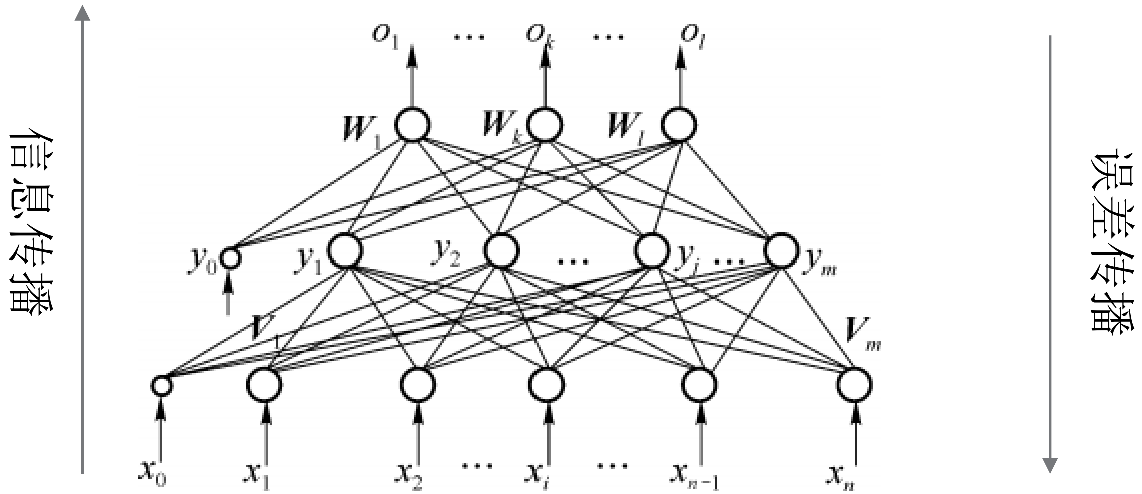
- 学习过程：过输出与期望值比较，调整权重因子

  非线性优化问题， 有众多参数

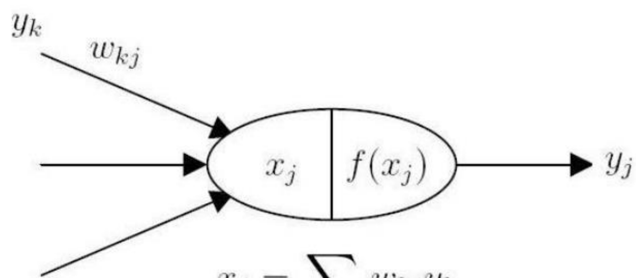
  梯度下降法，quasi-Newton-BFGS，模拟退火法，随机学习方法

BP-backward propagation of errors

- 误差反传学习算法--BP神经网络



$$o_k = f(\text{net}_k) \qquad k = 1, 2, \cdots, l$$



$$x_j = \sum_{k \in K_j} w_{kj} y_k,$$

$$y_j = f(x_j)$$

$$E := \frac{1}{2} \sum_{j=1}^{J} (t_j - y_j)^2.$$ 学习求极小值 cost function

$$\Delta w_{kj} = -\alpha \frac{\partial E}{\partial w_{kj}}.$$ 梯度下降法求导数

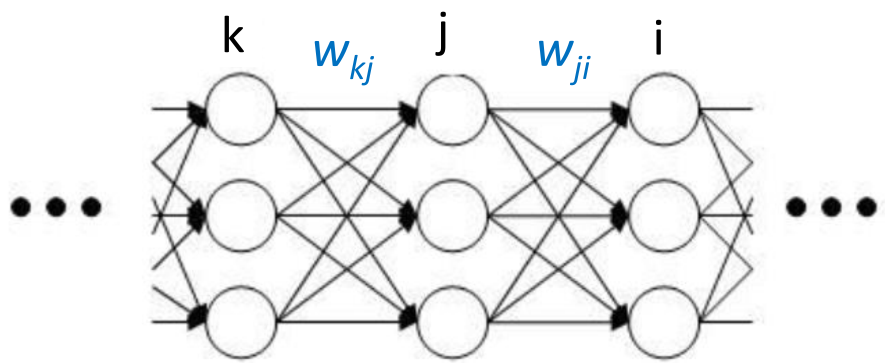
$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial x_j} \frac{\partial x_j}{\partial w_{kj}},$$

误差 $\delta_j := -\frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial x_j}.$ $\frac{\partial x_j}{\partial w_{kj}} = y_k.$

- $$\frac{\partial y_j}{\partial x_j} = y_j(1 - y_j).$$ $f(z) = \frac{1}{1+e^{-\gamma z}};$ sigmoid function

$$\frac{\partial E}{\partial y_j} = -(t_j - y_j).$$ j 为output时

→ $$\frac{\partial E}{\partial w_{kj}} = -(t_j - y_j)y_j(1 - y_j)y_k.$$ 单层output网络梯度



$$\frac{\partial E}{\partial y_j} = \sum_{i \in I_j} \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial x_i} \frac{\partial x_i}{\partial y_j}.$$

对于多层网络

j is a hidden layer, $\partial E/\partial y_j$ is not so simple

前两项为上一层的误差 δ_i

- $$\frac{\partial x_i}{\partial y_j} = w_{ji}. \qquad \frac{\partial E}{\partial y_j} = -\sum_{i \in I_j} \delta_i w_{ji},$$

→ $$\frac{\partial E}{\partial w_{kj}} = -\sum_{i \in I_j} (\delta_i w_{ji}) y_j (1 - y_j) y_k. \qquad \frac{\partial E}{\partial w_{kj}} = -\delta_j y_k.$$

error term

$$\delta_j := (t_j - y_j) y_j (1 - y_j);$$ ouput layer,

$$\delta_j := (\sum_{i \in I_j} \delta_i w_{ji}) y_j (1 - y_j).$$ a hidden layer[先计算后一层的误差] 误差反向传播

$$\Delta w_{kj}(n) = \alpha \delta_j y_k + \eta \Delta w_{kj}(n-1)$$

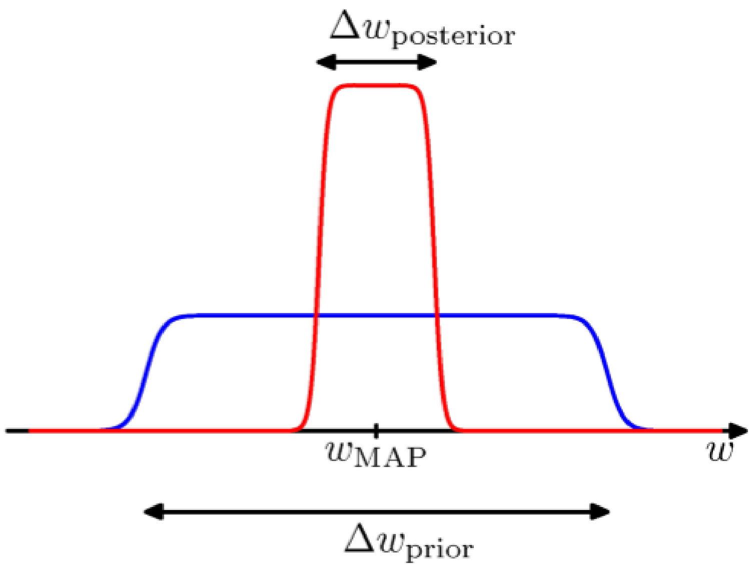
α is the learning rate, a real value on the interval (0,1]; η is the momentum

$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$ Update 权重系数，直到收敛完成学习

误差反向传播学习法可以应用于多种不同的神经网络，比如Bayesian BP, CNN.
标准的BP神经网络存在局部收敛，收敛慢，过度拟合等问题，有许多改进

Occam's Razor

-



$$posterior = \frac{likelihood * prior}{evidence}$$

marginal likelihood

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)\,\mathrm{d}w$$

$$\simeq \quad p(\mathcal{D}|w_{\text{MAP}})\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}$$

最优参数附近

Evidence: the probability that randomly selected parameters from the prior would generate D

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|w_{\text{MAP}}) + \underbrace{\ln\left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}\right)}_{\text{Negative}}.$$

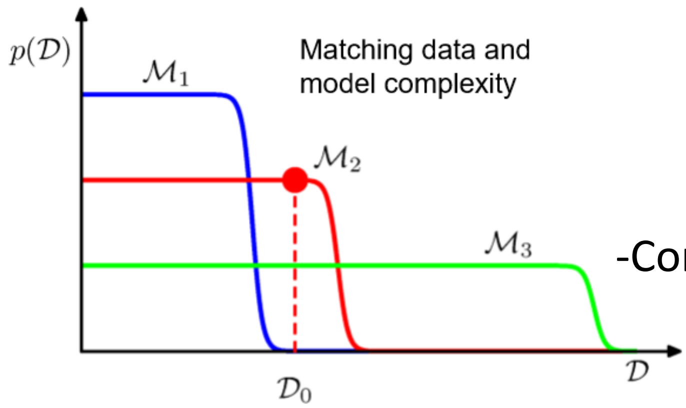
如果有*M*个参数

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\mathbf{w}_{\text{MAP}}) + \underbrace{M \ln\left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}\right)}_{\text{Negative and linear in } M}.$$

模型的竞争-balance：the first term will increase, whereas the second term will decrease with increasing M.

Occam's Razor

-



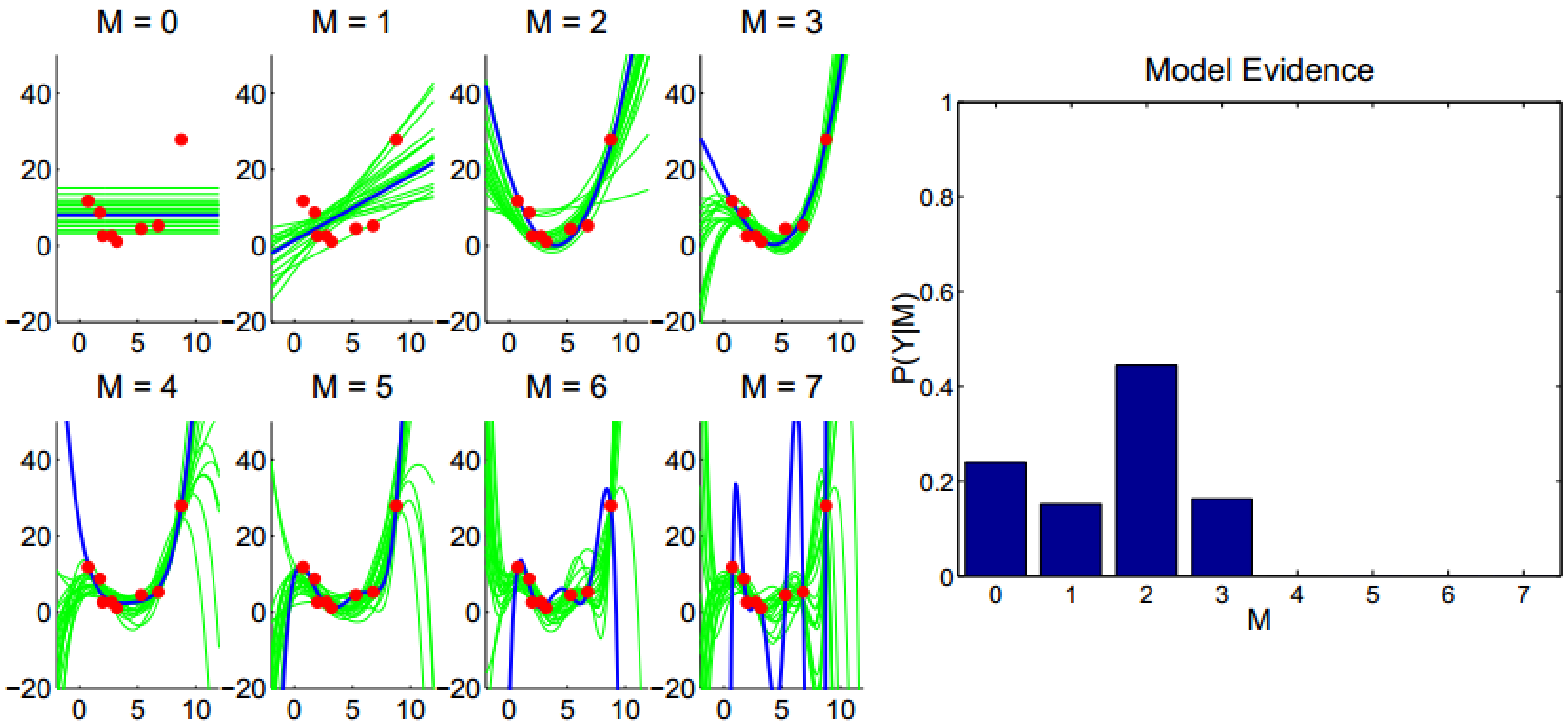
-Complex model

For the particular observed dataset D_0, the model M_2 with intermediate complexity has the largest *evidence*.

The simple model M1 cannot fit the data well, whereas the more complex model M3 spreads its predictive probability and so assigns relatively small probability to any one of them.
Be careful the choice of prior.

marginal Likelihood来看两个模型都能描述好，越简单的模型越好

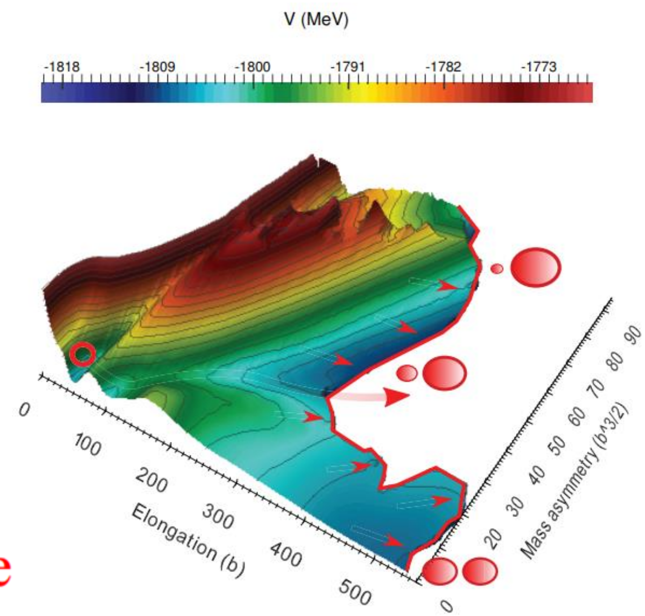
- 模型比较：



Bayesian evaluation of incomplete fission yields

- Fission data are key ingredients in many nuclear applications, need accurate and complete energy-dependent fission yields

- Fission is very complex:
  - Dynamics on Multi-dimensional PES;
  - Energy dependent fission;
  - Post-neutron fission fragments;



- **Problem:** Evaluations in major nuclear data libraries (ENDF, JENDL, JEFF, CENDL), energies only available at thermal, 0.5, 14 MeV

Limited neutron resources:14 MeV $^{3}H(d,n)^{4}He$

PHYSICAL REVIEW LETTERS **123**, 122501 (2019)

Bayesian Evaluation of Incomplete Fission Yields

Zi-Ao Wang, Junchen Pei,* Yue Liu, and Yu Qiang
State Key Laboratory of Nuclear Physics and Technology, School of Physics, Peking University, Beijing 100871, China

Evaluation methods of fission yields

- Measurements are difficult, incomplete, mostly thermal U235 and Pu239, evaluations are always needed
- Multiple Gaussian approach (5 Gaussians, 7 parameters, 1960's)

$$Y(A) = \frac{N_1}{\sigma_1\sqrt{2\pi}}\left[e^{-\left(\frac{(A-\bar{A}-D_1)^2}{2\sigma_1^2}\right)} + e^{-\left(\frac{(A-\bar{A}+D_1)^2}{2\sigma_1^2}\right)}\right] + \frac{N_2}{\sigma_2\sqrt{2\pi}}\left[e^{-\left(\frac{(A-\bar{A}-D_2)^2}{2\sigma_2^2}\right)} + e^{-\left(\frac{(A-\bar{A}+D_2)^2}{2\sigma_2^2}\right)}\right] + \frac{N_3}{\sigma_3\sqrt{2\pi}} e^{-\left(\frac{(A-\bar{A})^2}{2\sigma_3^2}\right)}$$

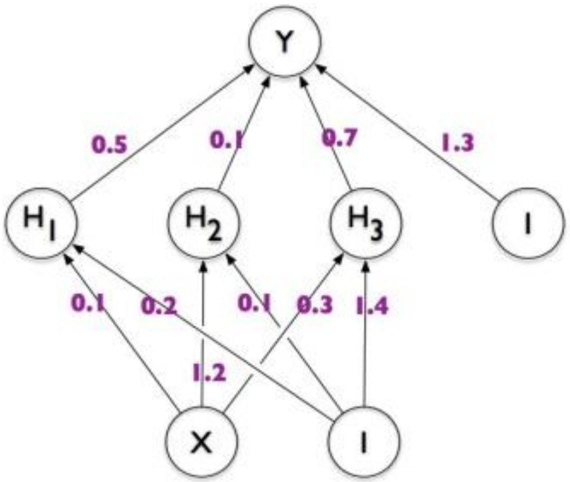
- Brosa model with random neck rupture(liquid-drop model+shell) (Phys.Rep., 1990)
- GEF model (~50 parameters, including physics inputs: barriers, level densities, fragment shells.....) (Rep.Prog.Phys., 2018)
- Machine learning for next-generation evaluations

Key issues： energy dependence, uncertainty propagation

Bayesian Neural Network

•

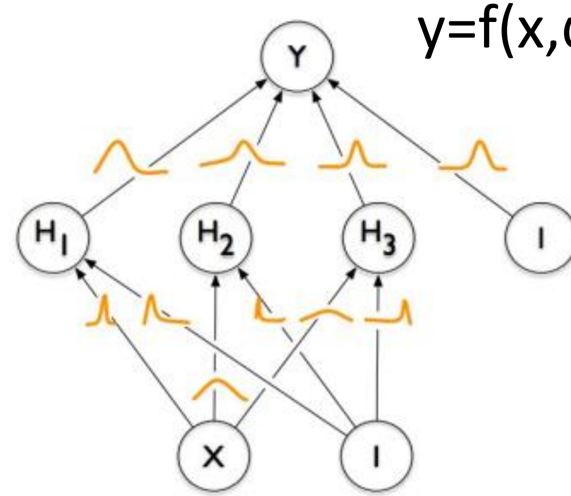
Standard Neural Network



Bayesian Neural Network

$y=f(x,\omega)+\sigma\varepsilon$

Hidden layer



Learning： minimize the cost

By adjust the connection weights

$$p(\omega|D)=\frac{p(D|\omega)p(\omega)}{p(D)}\propto p(D|\omega)p(\omega),$$

Connection weights are distribution function

Learning： maximize likelihood $P(\mathcal{D} \mid \mathbf{w})$

parameter uncertainty

structure uncertainty

Computing is very costly

Bayesian statistics

统计学意义 $P(h|D) = P(D|h)\,P(h) / P(D)$

posterior = likelihood x prior / evidence

- P(D) : prior probability of the data D, *evidence*, independent of hypothesis
- P(h) : prior probability of the hypothesis *h* before we know the data, ***prior***， reflect some background knowledge
- P(h|D) : posterior probability of the hypothesis given the data D, *posterior*,
- P(D|h) : probability of observing data D given the hypothesis h , *likelihood*

- By observing the data D we can convert the prior probability P(h) to the a posteriori probability (posterior) P(h|D); Given data D, update P(h) by P(h|D).
- The posterior is probability that h holds after data D has been observed. reflects the influence of D on our confidence. Probability measures degree of belief. Inference is conditional on the observed data
- The evidence P(D) can be viewed merely as a scale factor that guarantees that the posterior probabilities sum to one

- 模型，参数评估优化 (模型，参数，数据三者关系)

$$P(\text{parameters} \mid \text{data}) = \frac{P(\text{parameters})\, P(\text{data} \mid \text{parameters})}{P(\text{data})}$$

$$P(\theta|\mathcal{D}, m) = \frac{P(\mathcal{D}|\theta, m)P(\theta|m)}{P(\mathcal{D}|m)}$$

$P(\mathcal{D}\|\theta, m)$	likelihood of parameters θ in model m
$P(\theta\|m)$	prior probability of θ
$P(\theta\|\mathcal{D}, m)$	posterior of θ given data $\mathcal{D}$

posterior distribution for the model parameters given the observed data

Prediction:

$$P(x|\mathcal{D}, m) = \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta$$

新数据

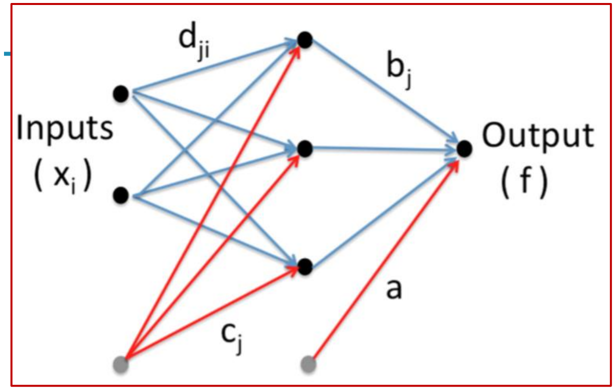
Model Comparison:

$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}$$

$$P(\mathcal{D}|m) = \int P(\mathcal{D}|\theta, m)P(\theta|m)\, d\theta$$

- Net function:

$$f(x,\omega) = a + \sum_{j=1}^{H} b_j \tanh(c_j + \sum_{i=1}^{l} d_{ij} x_i)$$



Likelihood:

$$p(D|\omega) = \exp(-\chi^2/2), \quad \chi^2 = (t_i - f(x_i,\omega)^2)/\Delta t_i^2$$

Posterior:

$$p(\omega|D) = \frac{p(D|\omega)p(\omega)}{\int p(D|\omega)p(\omega)d\omega}$$

Input： Z, N, $A_{fragment}$, E*
Output: Yields

Average infer:

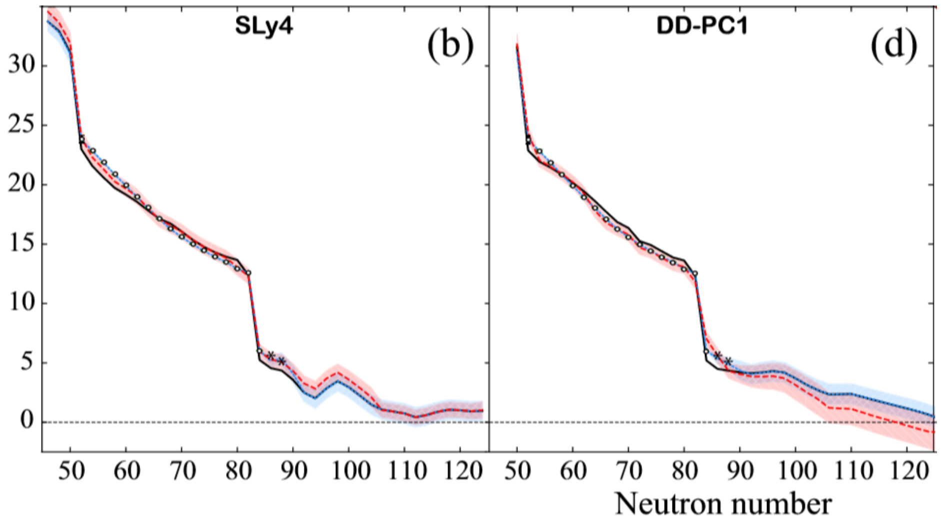
$$< f(x_n,\omega) >= \int f(x_n,\omega)p(\omega|D)d\omega$$

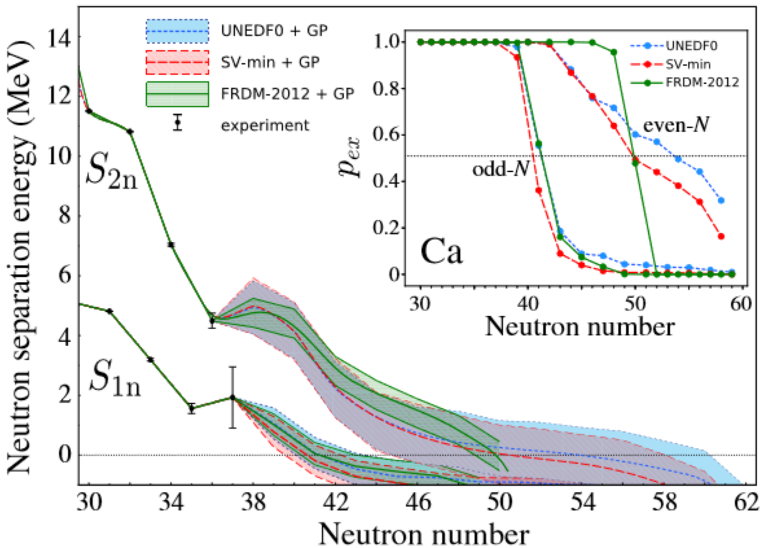
Markov-chain Monte Carlo integral

$$p(\omega|D, D_n) = \frac{p(D_n|\omega)p(\omega|D)}{\int p(D_n|\omega)p(\omega|D)d\omega}$$

Evaluation： infer after take into account new data

Bayesian NN applications in nuclear physics





GP regression (only 3 parameters) is excellent compared to BNN, BNN could be unstable due to many parameters

$$w_k := p(\mathcal{M}_k|^{52}\mathrm{Cl}, {}^{53}\mathrm{Ar}, {}^{49}\mathrm{S}\ \mathrm{exist})$$

^{68}Ca has an average posterior probability ≈76% to be bound to 2n emission

J. McDonnell, N. Schunck, D. Higdon, J. Sarich, S.M. Wild, W. Nazarewicz, PRL. 114, 122501(2015)

R. Utama, J. Piekarewicz, and H.B. Prosper, Phys. Rev. C 93, 014311 (2016);

L. Neufcourt, Y. Cao, W. Nazarewicz, and F. Viens, Phys. Rev. C 98, 034318 (2018).

Z. Niu and H. Liang, Phys. Lett. B 778, 48 (2018)

L. Neufcourt ,Phys. Rev. Lett. 122, 062502(2019)

Hai Fei Zhang, et al., J. Phys. G: Nucl. Part. Phys. 44 045110 (2017)

X. H. Wu and P. W. Zhao Phys. Rev. C 101 051301 (2020)

Bayesian NN applications in nuclear physics

- (Spallation cross sections) CW Ma, D Peng, HL Wei, ZM Niu, YT Wang, Chinese Phys. C, 2020
- (Charge radius) D Wu, CL Bai, H Sagawa, HQ Zhang - arXiv:2006.09677, 2020
- (Charge radius), Yunfei Ma, Chen Su, Jian Liu, Zhongzhou Ren, Chang Xu, and Yonghao Gao, Phys. Rev. C 101, 014304 (2020)
- (alpha decays) B. Cai, G.s. Chen, J.y. Xu, Cenxi Yuan, Chong Qi, and Yuan Yao, Phys. Rev. C 101, 054304 (2020)
- (beta decay) Z. M. Niu,H. Z. Liang ,B. H. Sun,W. H. Long ,and Y. F. Niu,Phys. Rev. C 99, 064307 (2019) - Published 5 June 2019
- Excitation energies(R-D. Lasseri, Phys. Rev. Lett. 124, 162502 (2020), deuteron wavefunctions (J. W. T. Keeble, A. Rios, arXiv:1911.13092), PES, model space truncations (W. G. Jiang, G. Hagen, and T. Papenbrock, Phys. Rev. C 100 054326 (2019))……
- High energy heavy-ion collisions, nuclear reactions and studies of EoS…

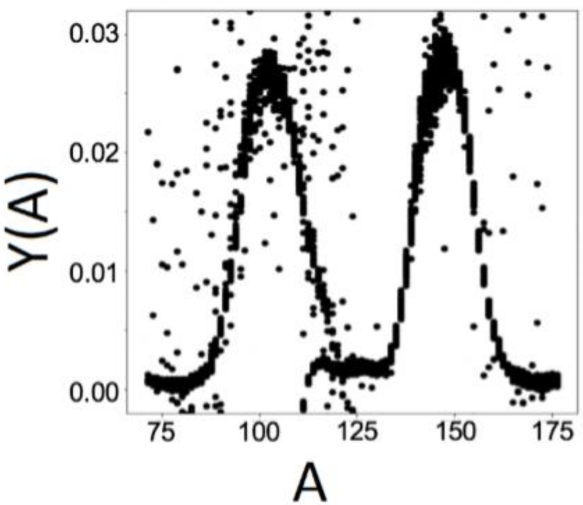
Other neural networks for fission yields

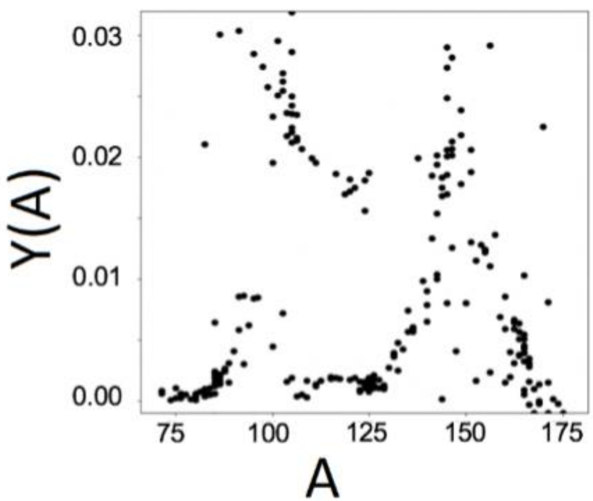
- A hot topic in nuclear data conference

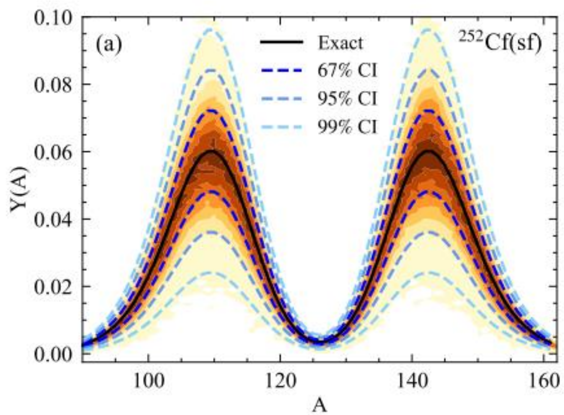
the Mixture Density Network (MDN)

$$f(x) = \alpha_1 \mathcal{N}(\mu_1, \sigma_1) + ... + \alpha_n \mathcal{N}(\mu_n, \sigma_n).$$

a mixture of Gaussians,



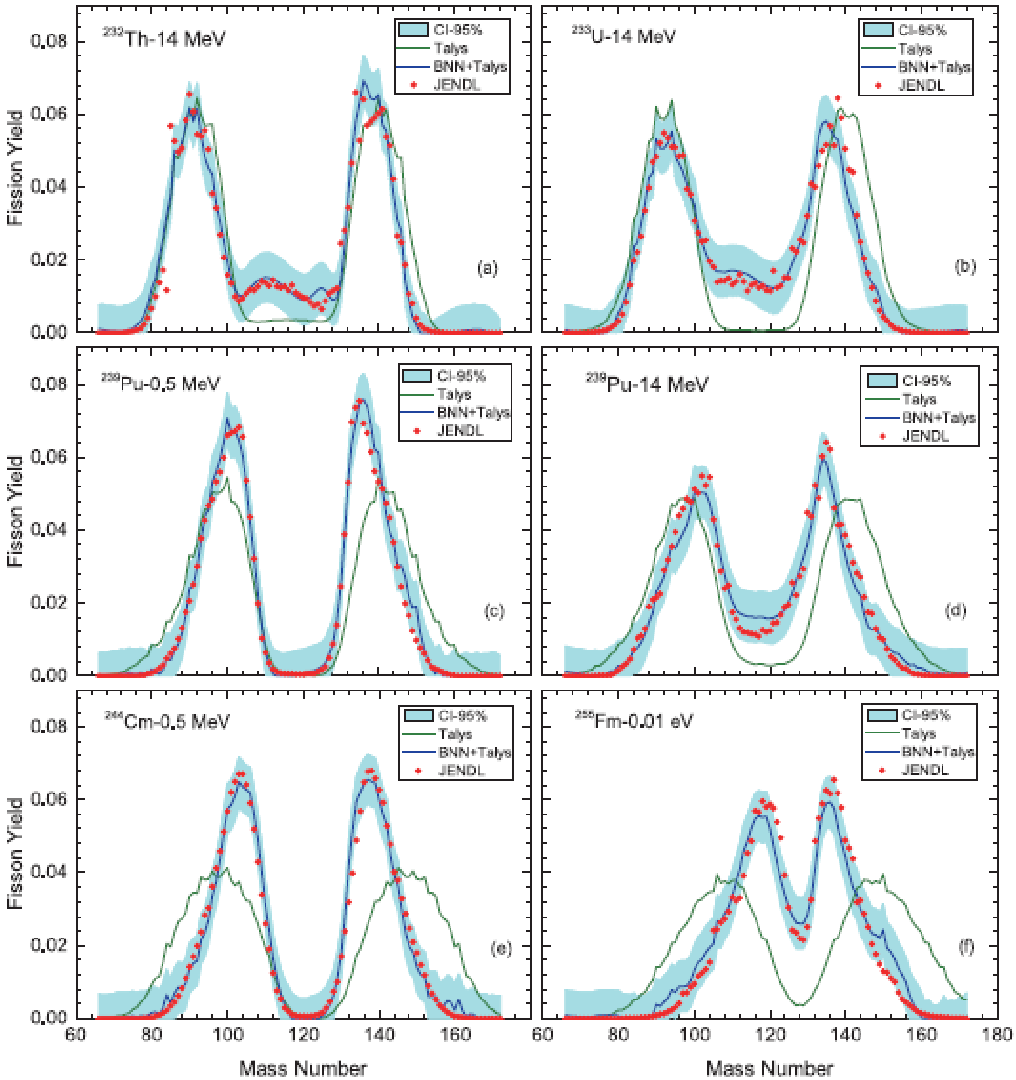




A. Lovell, A. Mohan, P. Talou and M. Chertkov, EPJ Web Conf. 211, 04006 (2019)
A.E. Lovell, A.T. Mohan, P. Talou, arXiv:2005.03198, J.Phys.G

Bayesian evaluation of incomplete fission yields

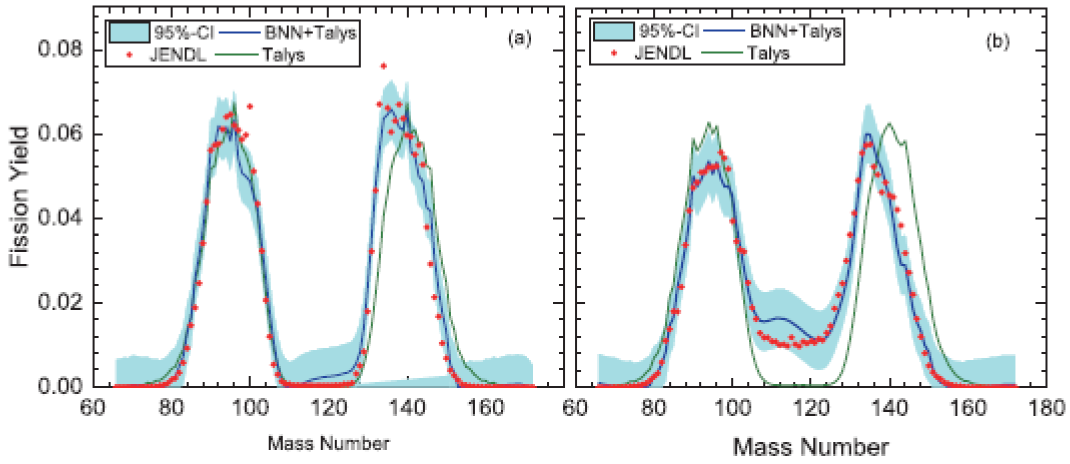
- Z.A. Wang, J.P., Y.Liu, Y.Qiang, **PRL 123,122501 (2019)**



Learning of fission yields of 30 nuclei using BNN+Talys to learn residuals

Previously BNN+HFB for nuclear mass residual predictions

Validation of BNN



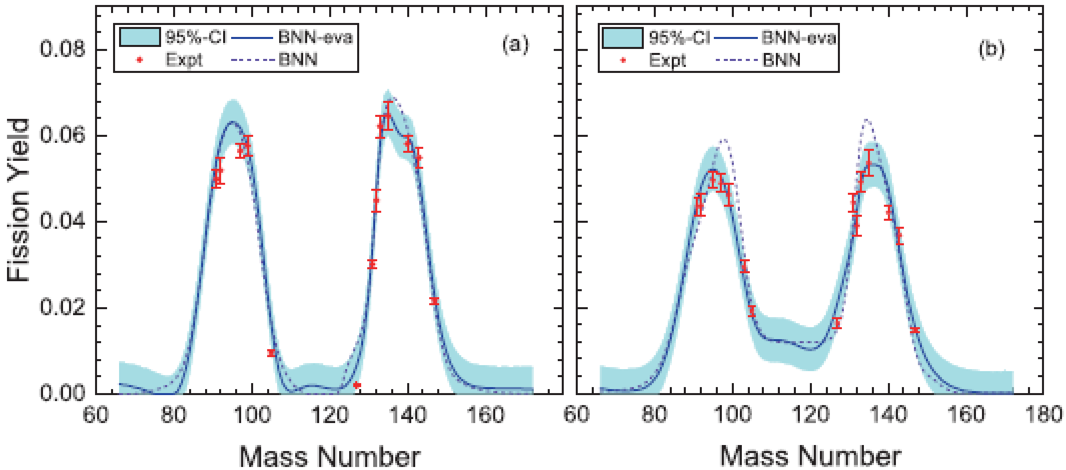
Validation of n+^{235}U

Normalization close to ~2.0 within 2%

Models with different network configurations

Models	learning χ^2_N	validation χ^2_N
BNN-32	1.574	1.317
BNN-40	1.616	1.640
BNN-32-resample	1.314	1.179
BNN-32+Talys	1.902	1.302
BNN-40+Talys	1.139	1.134
BNN-32-resample+Talys	1.547	1.419
Talys(pre-n)	17.56	8.964
Talys	16.79	8.334

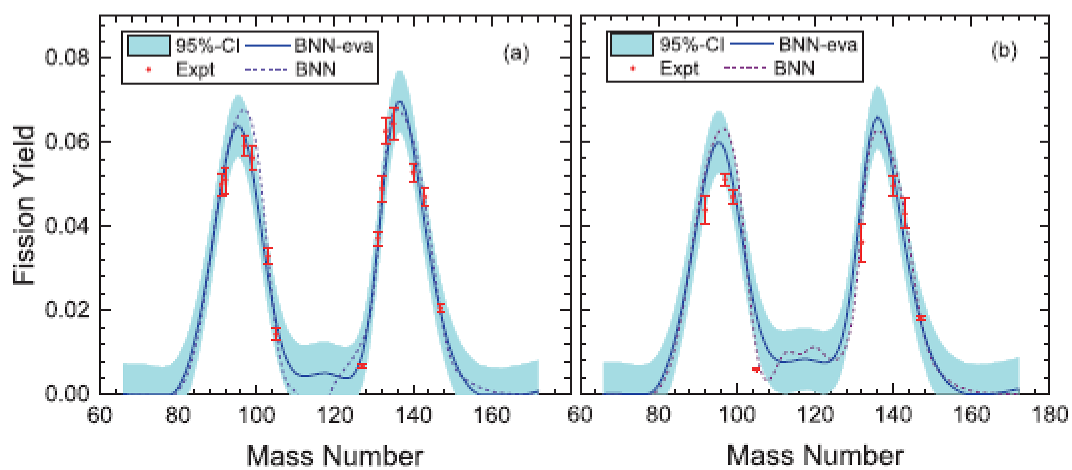
Key motivation: evaluation of incomplete FPY



Evaluation at 1.37 and 14.8 MeV

$$p(\theta|D, D_n) = \frac{p(D_n|\theta)p(\theta|D)}{\int p(D_n|\theta)p(\theta|D)d\theta}$$

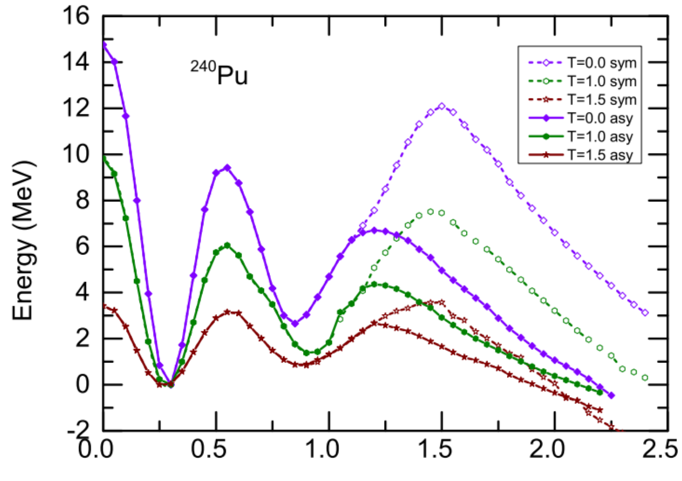
M.E. Gooden, et al., Nucl. Data Sheets 131, 319(2016)



Evaluation at 4.49 and 8.9 MeV

Z. Wang, J.P., et al arXiv:1906.04485
PRL 123,122501 (2019)

Energy dependence of fission yields

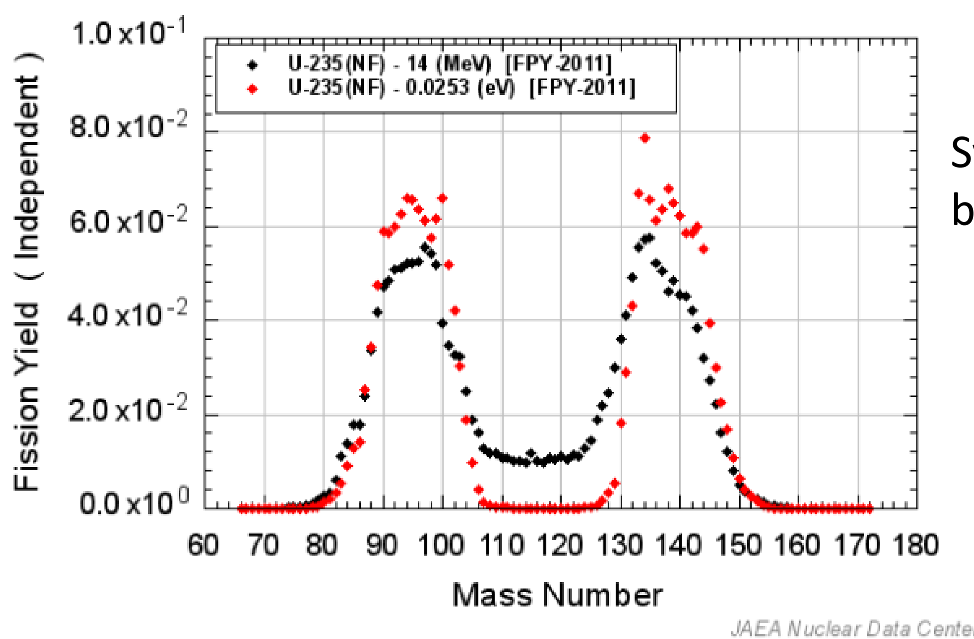


Pei, Zhu, 2017 Nucl. Phys.Rev

Quantum effects fades away

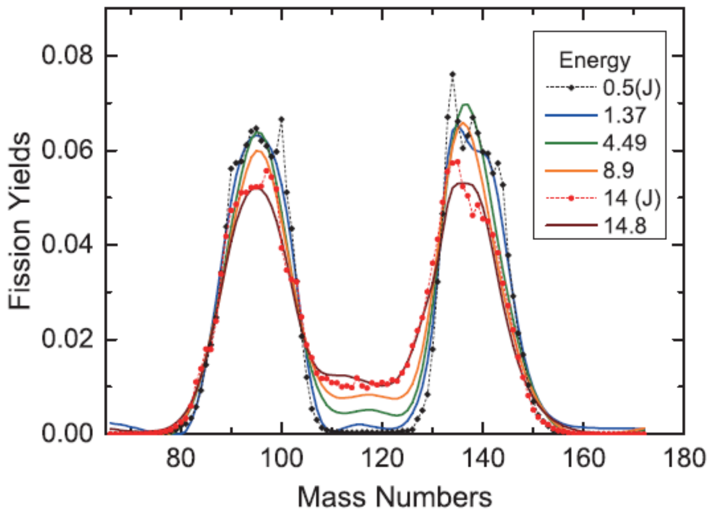
$$B_f = B_{LD} - \delta W e^{-\gamma_D E^*}$$

U-235 Neutron-induced Fission Yields



Symmetric fission becomes dominated

- **Energy dependence can be captured by BNN**

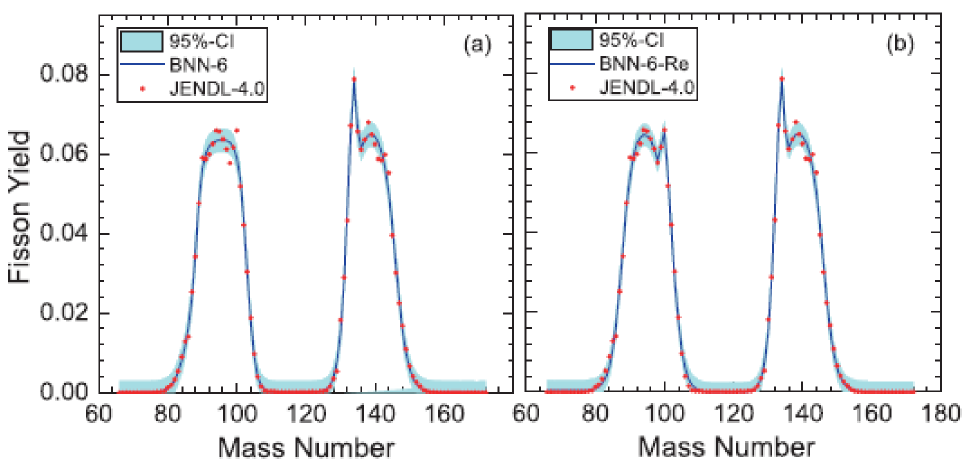


Symmetric fission will increase as excitation energy increase, as shell effects fades in compound nuclei

J. Randrup et al. PRC 88, 064606 (2013).

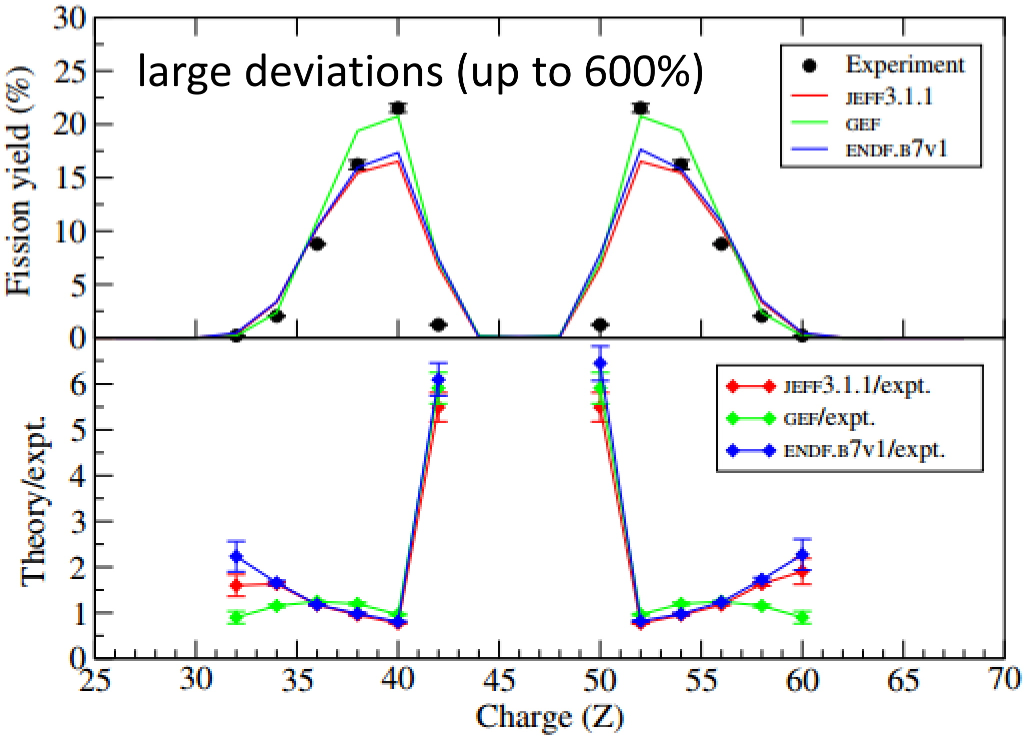
J. Zhao,et al. PRC 99, 014618 (2019)

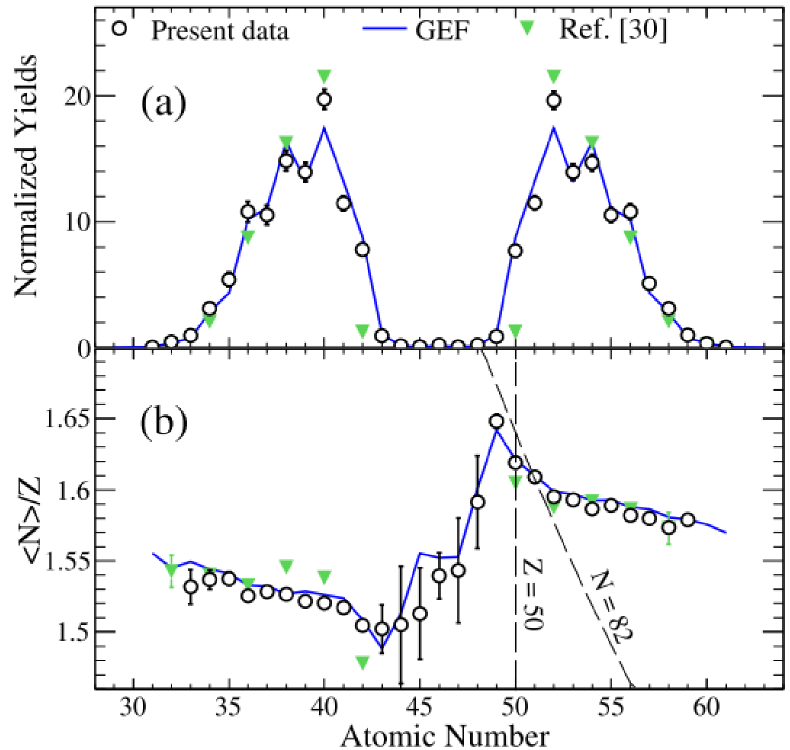
J. Sheikh,W.Nazarewicz, J.P, PRC (2009)



- Reinforcement Learning is useful
- Still need to improve our approach to modeling quantitive fission data

Evaluations of charge yields

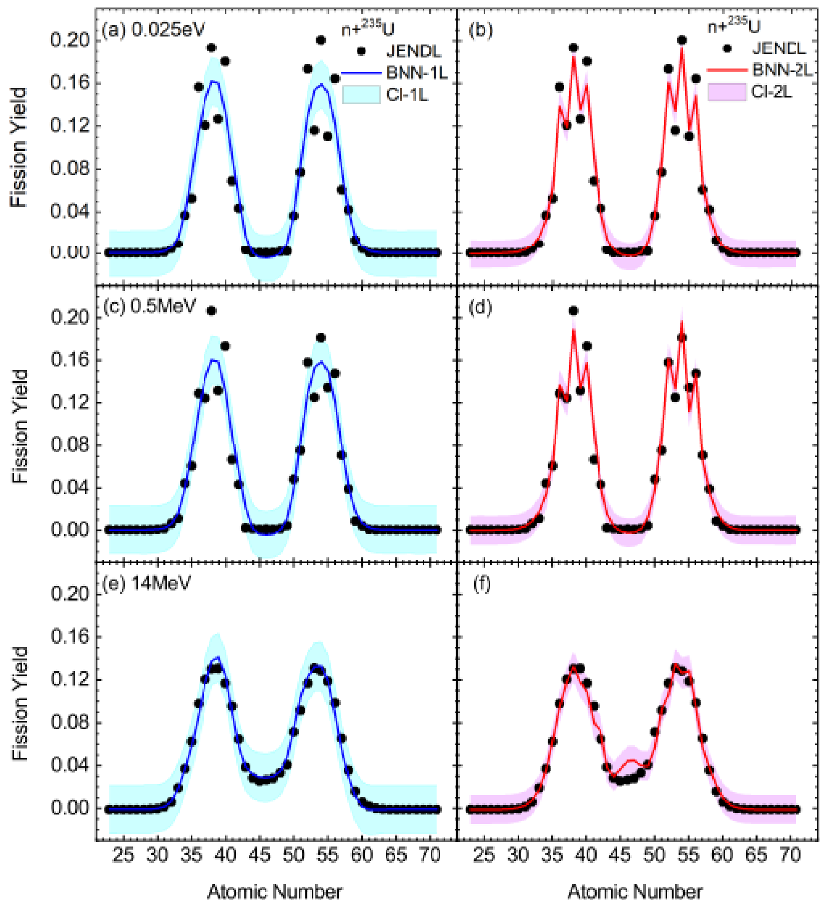




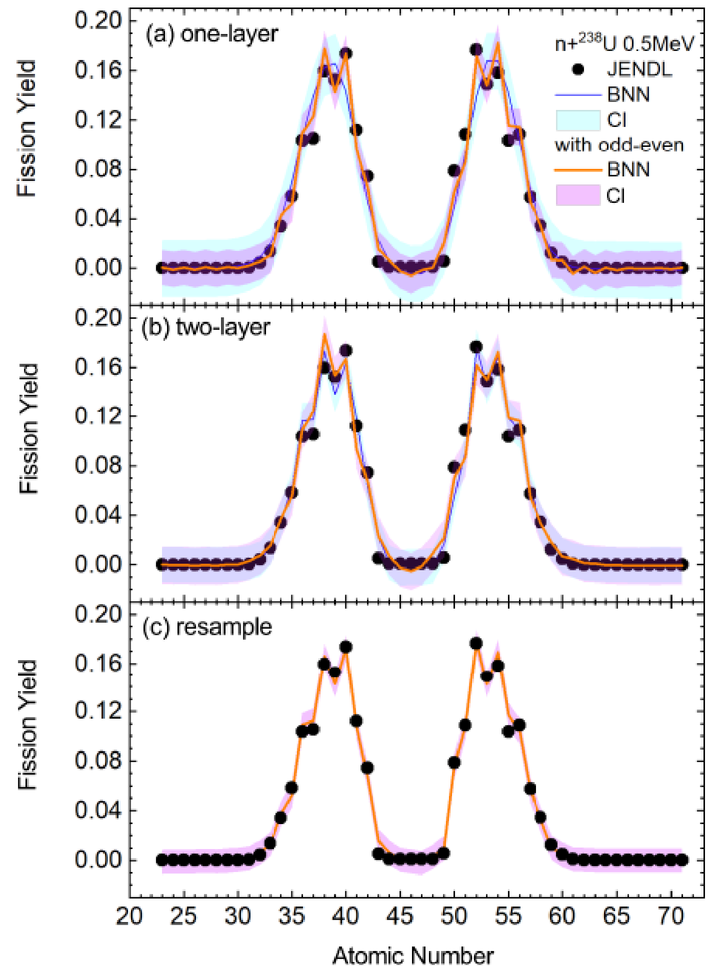
Two experiments on fission of 239U
Phys. Rev. Lett. 118, 222501(2017)
Phys. Rev. Lett. 123, 092503(2019);

inverse kinematics + magnetic spectrometer
Recent great progress in fission experiments: measurements of full fragment yields

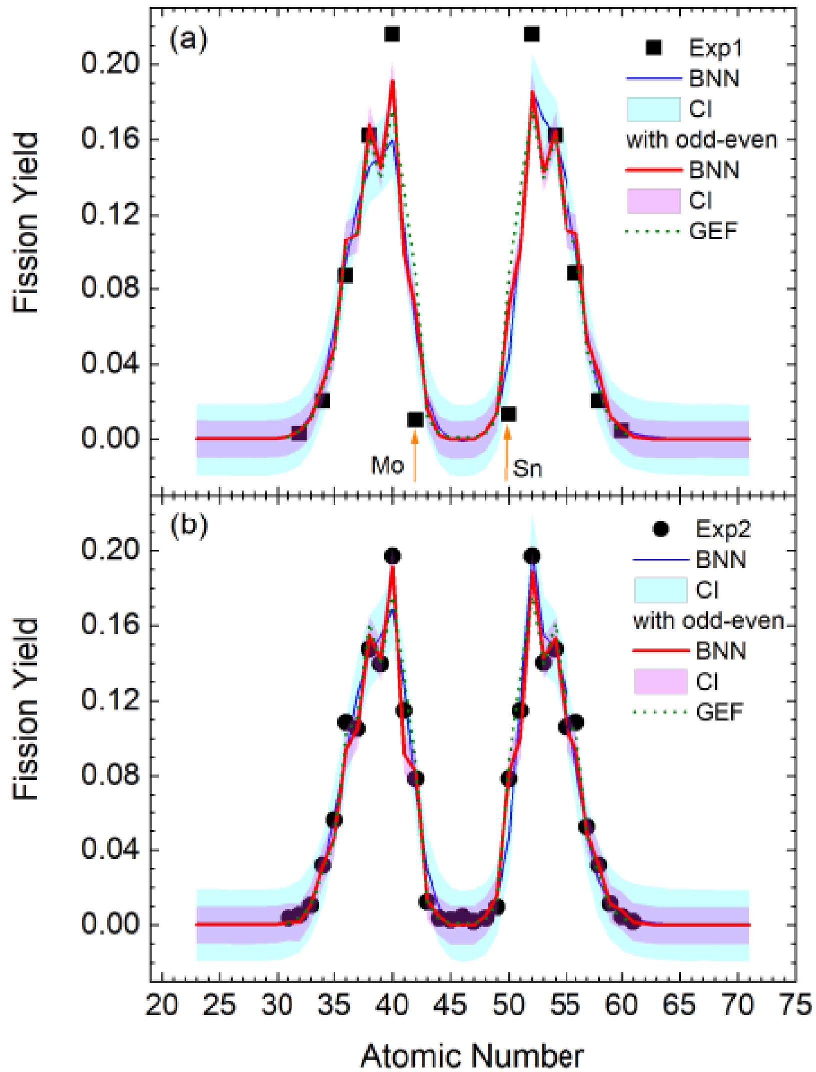
Evaluations of charge yields



One layer-32 Double layer 16-16



Odd-even effects



实验在Sn， Mo的电荷产额有很大分歧

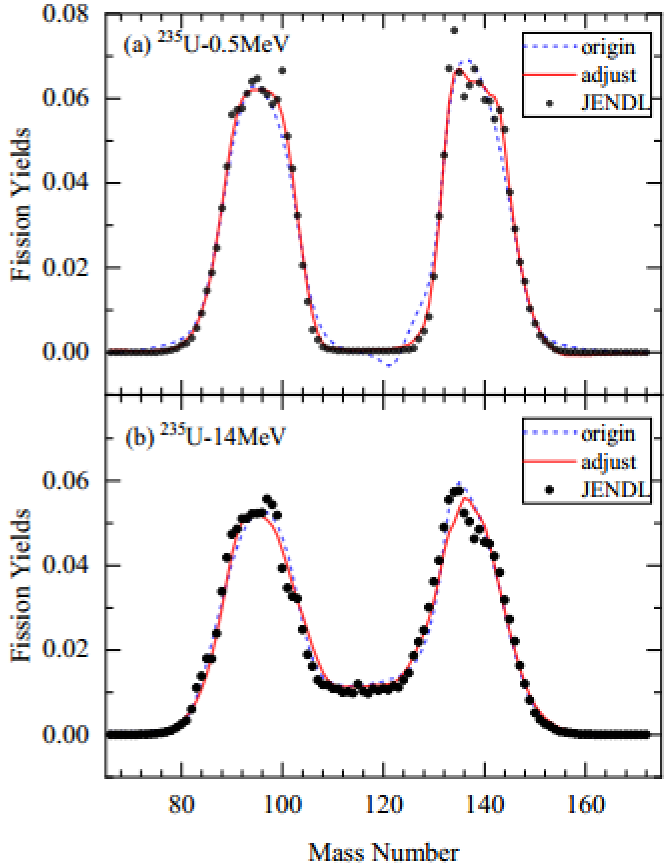
Expt1: PRL 118, 222501(2017)
Expt2: PRL 123, 092503(2019);

我们BNN评价结果支持2019年的实验

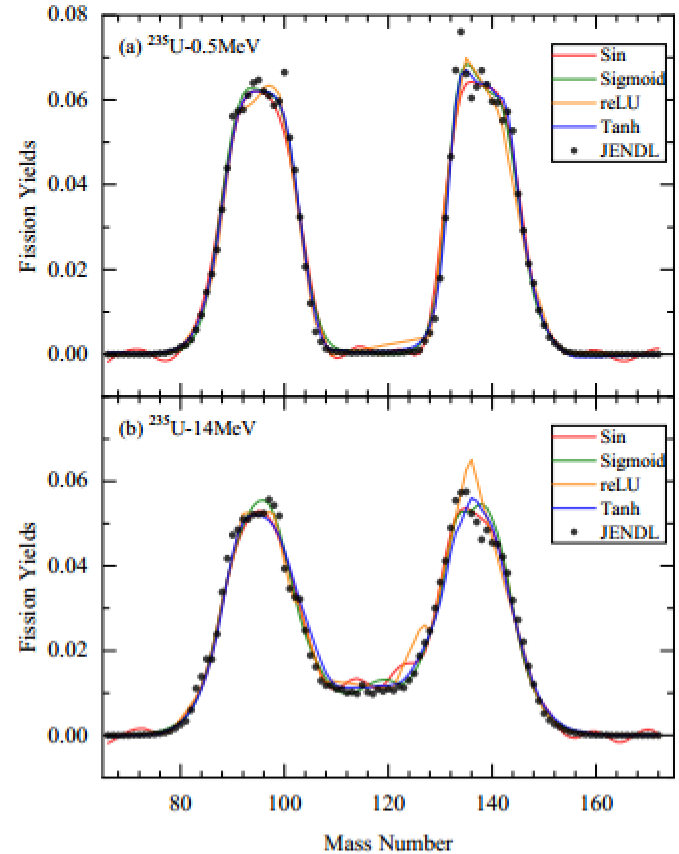
C. Y. Qiao, J. C. Pei, Z. A. Wang, Y. Qiang, Y. J. Chen, N. C. Shu, and Z. G. Ge， Phys. Rev. C 103, 034621 (2021)

Optimization of BNN

- adjustment of data

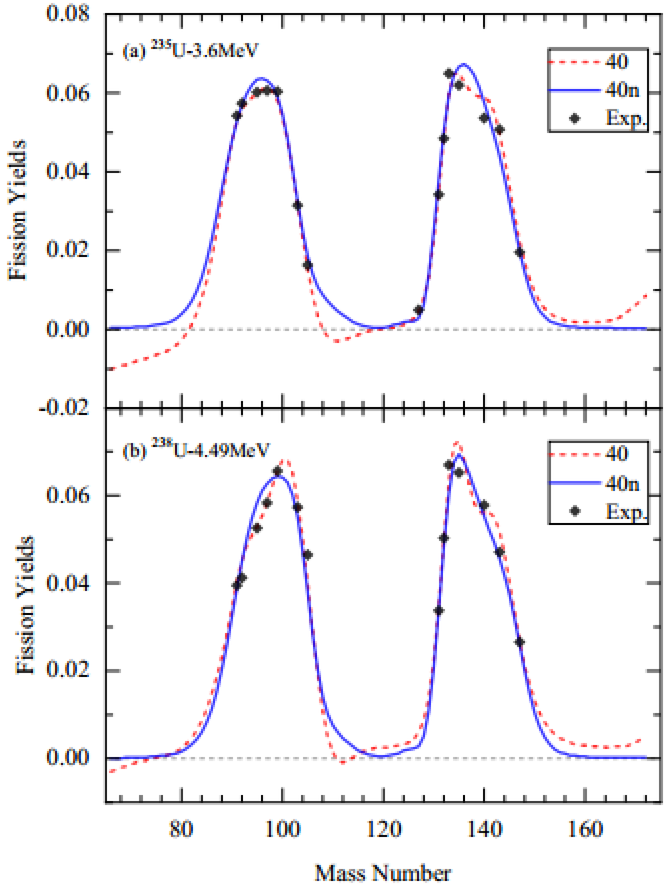


adjustment of active function

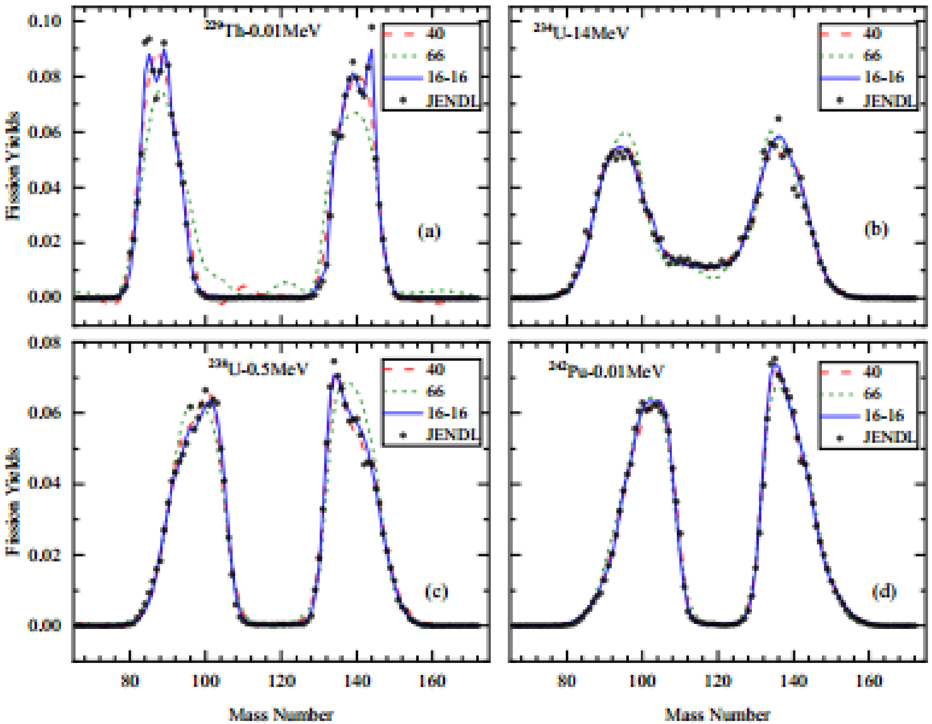


Bayesian machine learning is a promising tool for the evaluation of nuclear data but its potential capability has not been fully realized.

- Penalty on negative values

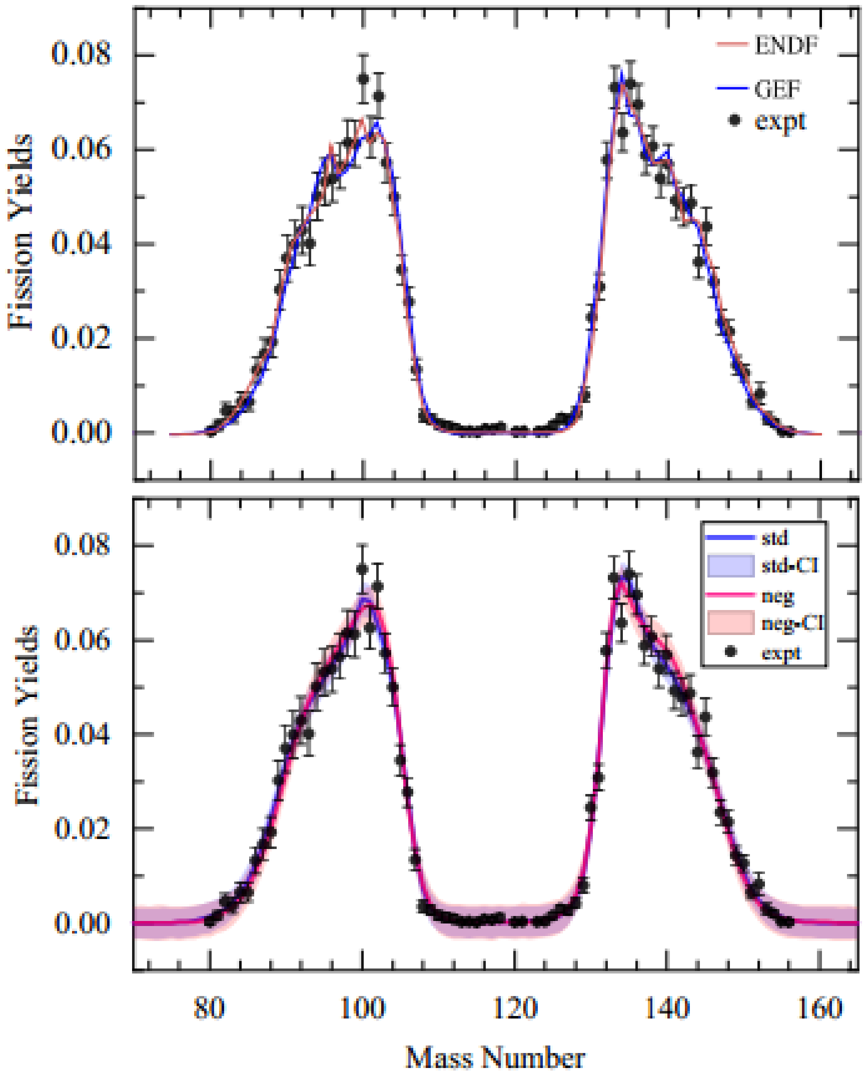


Structure of network



Shadow or deeper neural networks?

66	16-16	11-12-12	9-10-10-10	9-9-8-8-9	8-8-8-8-7-7	7-7-7-7-7-7-8
4.35×10^{-6}	3.43×10^{-6}	4.07×10^{-6}	4.99×10^{-6}	4.64×10^{-6}	5.05×10^{-6}	4.94×10^{-6}

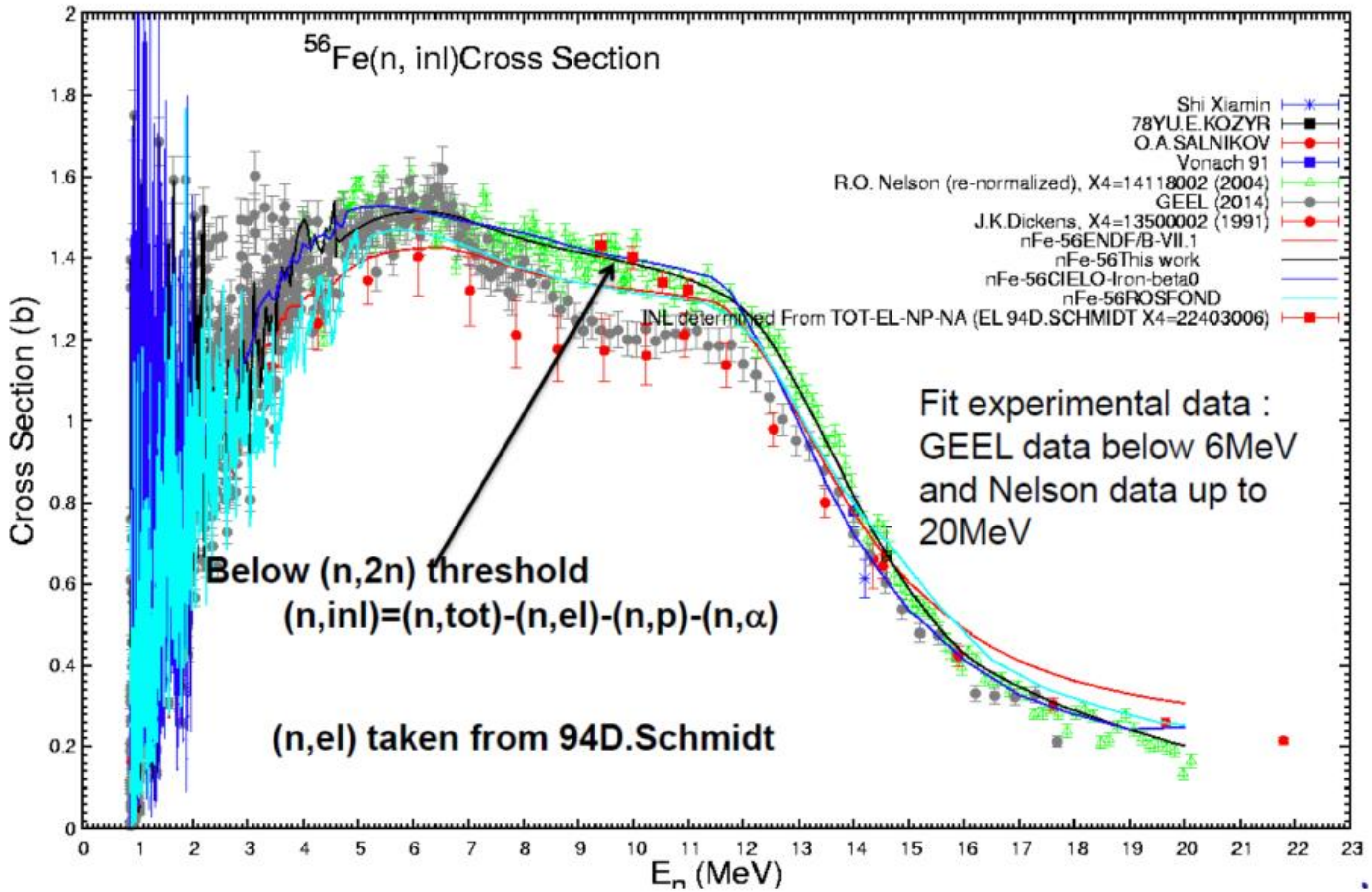


D.Ramos et al., Phys. Rev. C 101, 034609(2020)

Z.A. Wang, J.C.Pei, arXiv:2106.11746

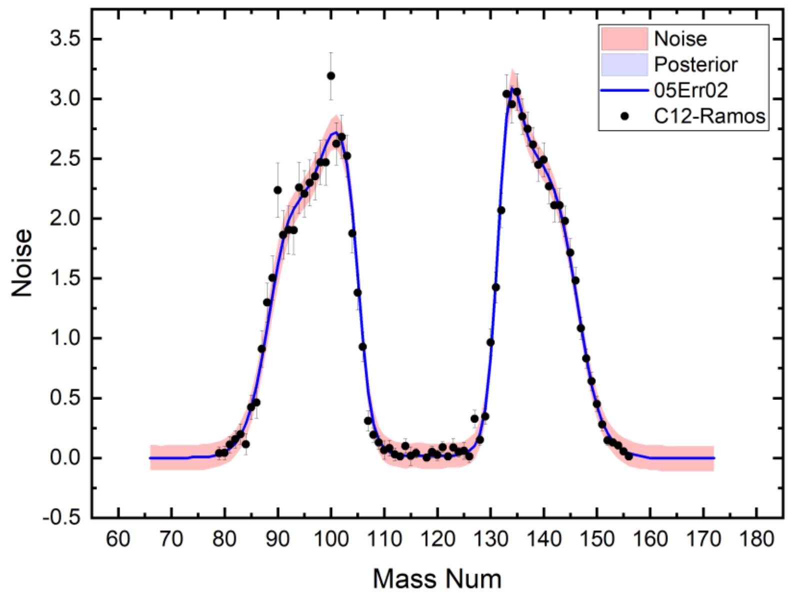
Machine Learning for data fusion (Heterogeneous)

- Evaluations of incomplete, divergent data, Noisy data, or heterogeneous data

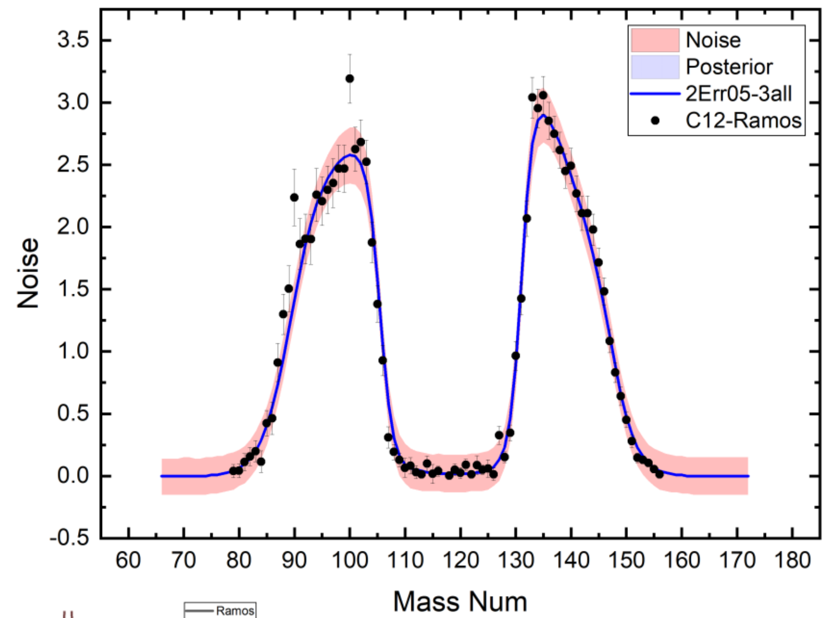


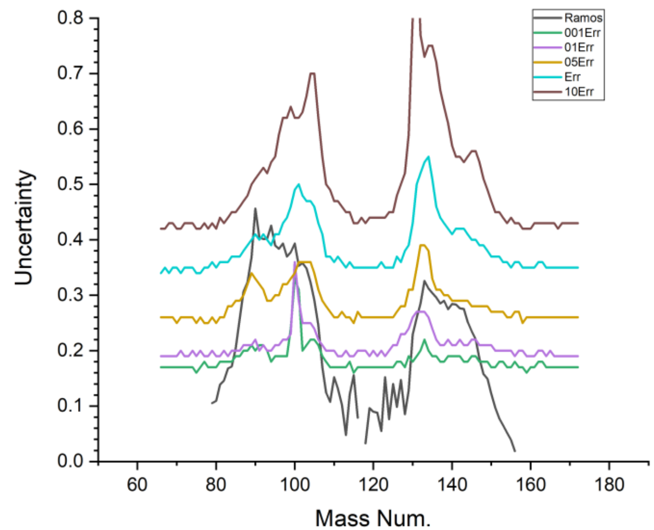
Machine Learning for data fusion (Heterogeneous)

- Propagation of uncertainties



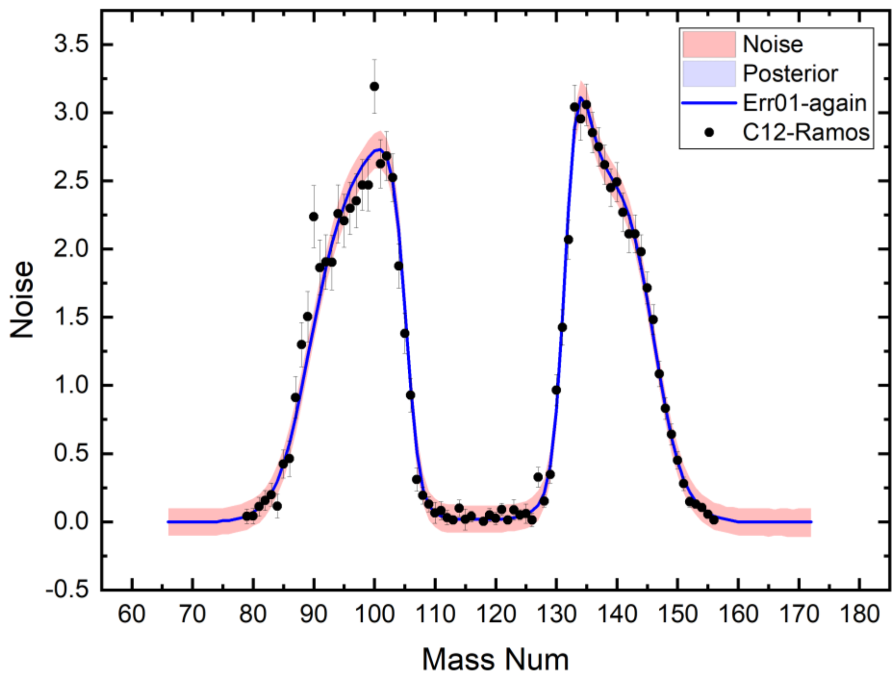
2-times expt uncertainties

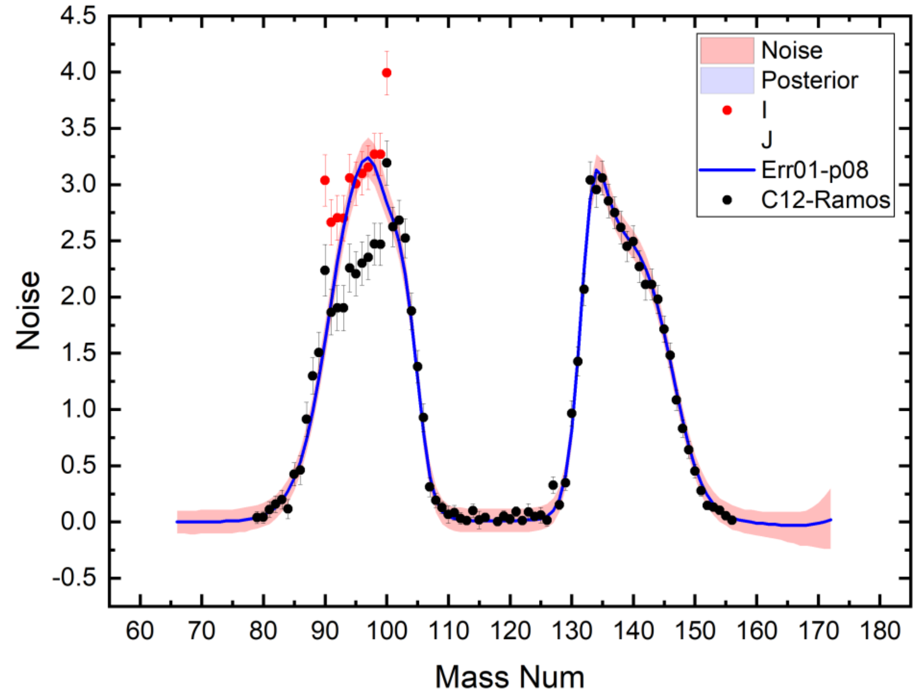




Z.A.Wang, in preparation

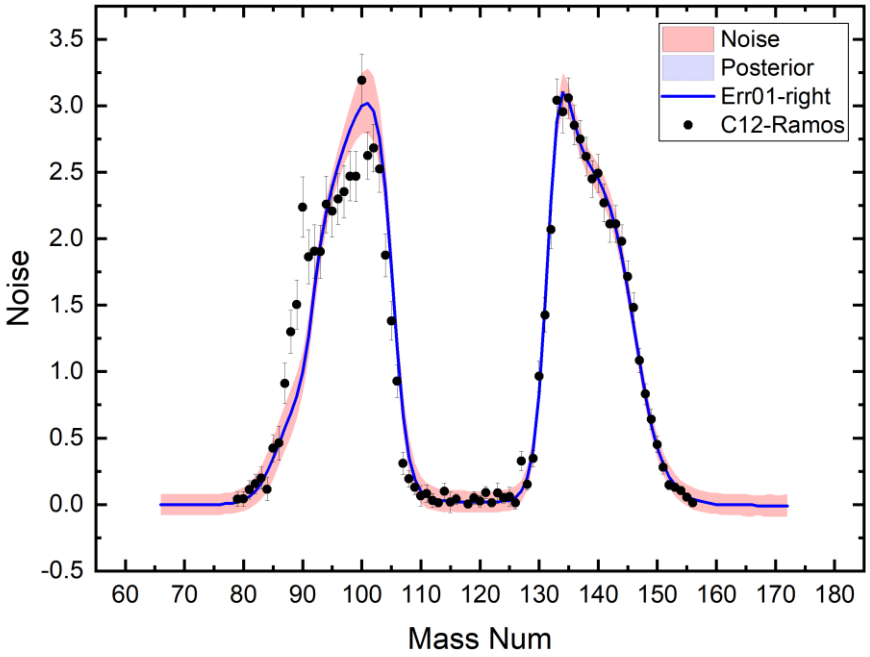
- evaluation with divergent data



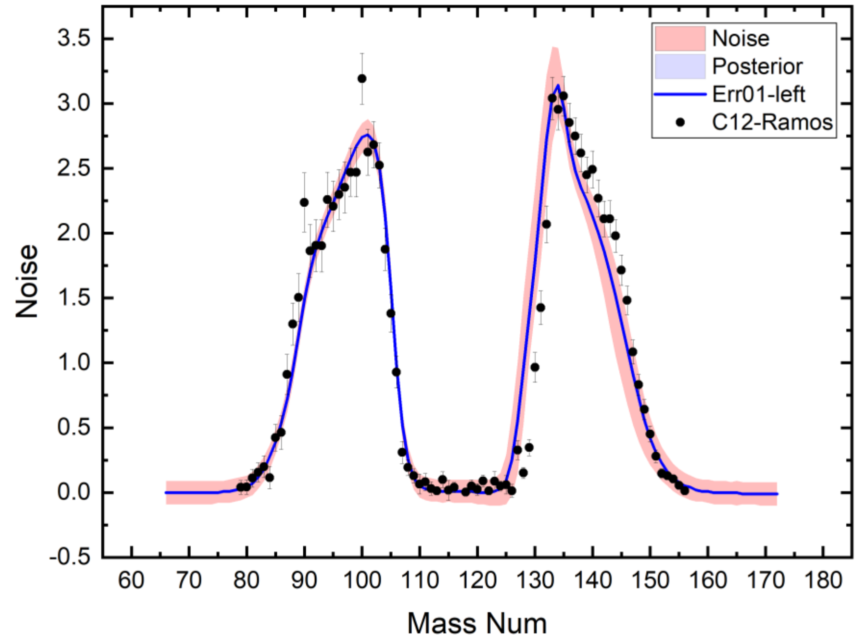


- Evaluation with incomplete data

左峰缺失

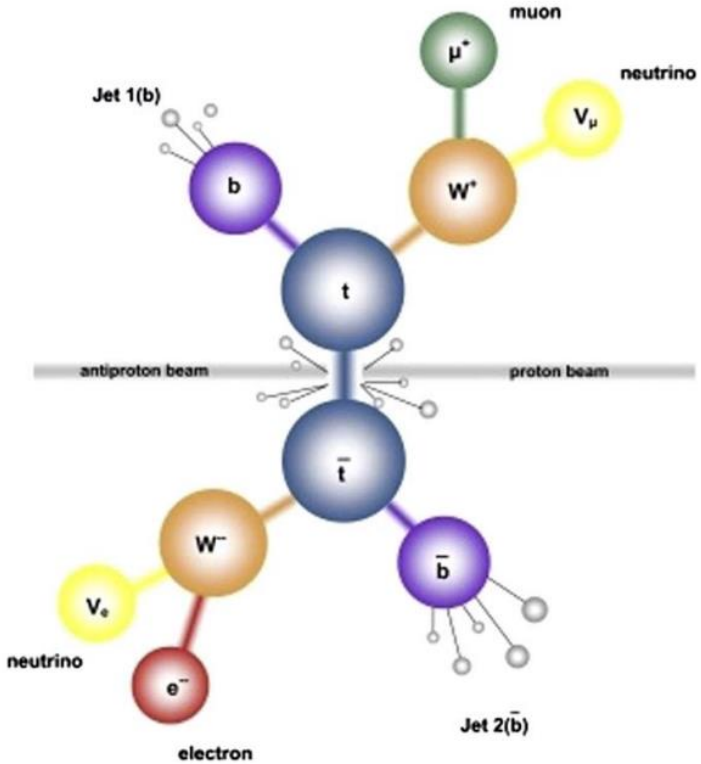


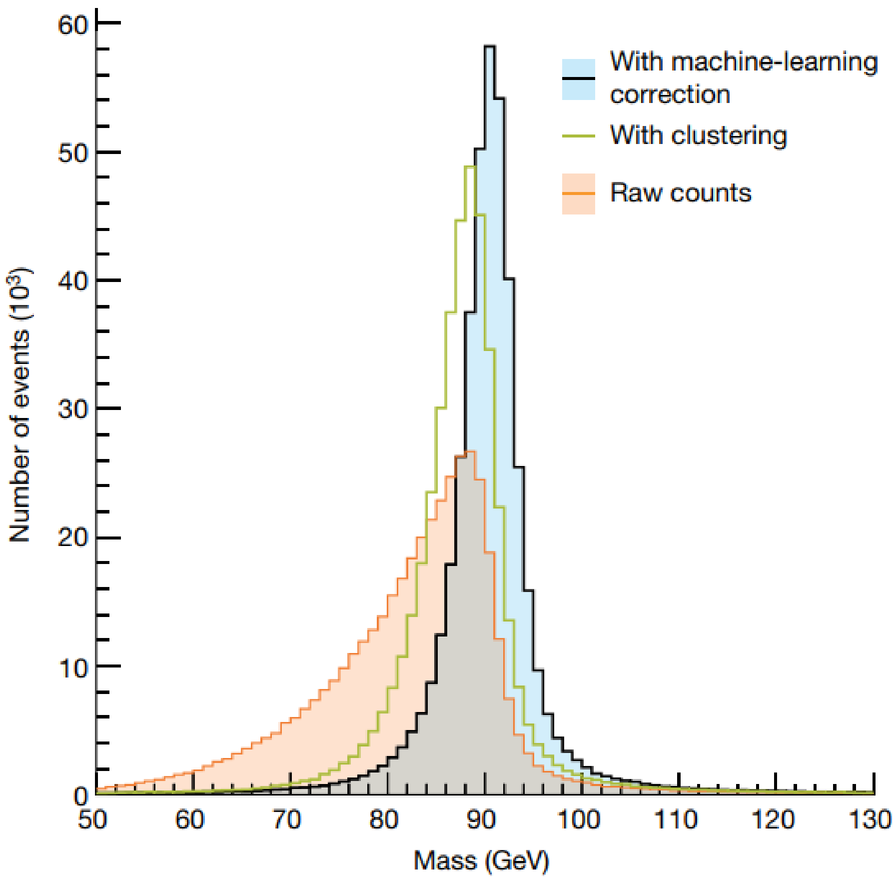
右峰缺失



Other applications

- train particle selection in high energy physics





A. Radovic, 41 | NATURE | VOL 560 | 2018

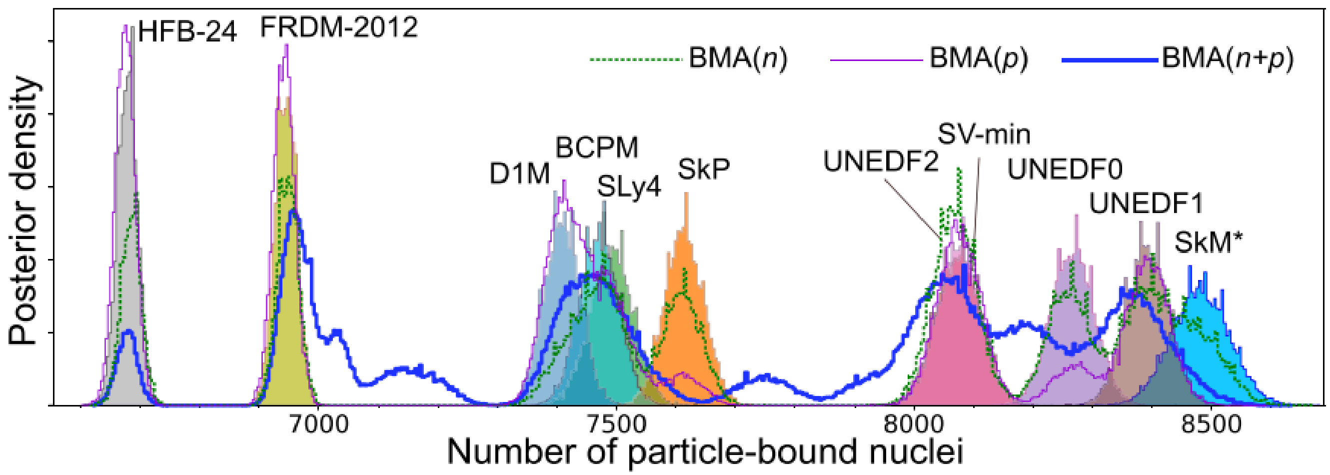
- **Bayesian model mixing for nuclear mass**

$$p(\mathcal{M}_k|y) = \frac{p(y|\mathcal{M}_k)\pi(\mathcal{M}_k)}{\sum_{\ell=1}^{K} p(y|\mathcal{M}_\ell)\pi(\mathcal{M}_\ell)}$$

Prediction:

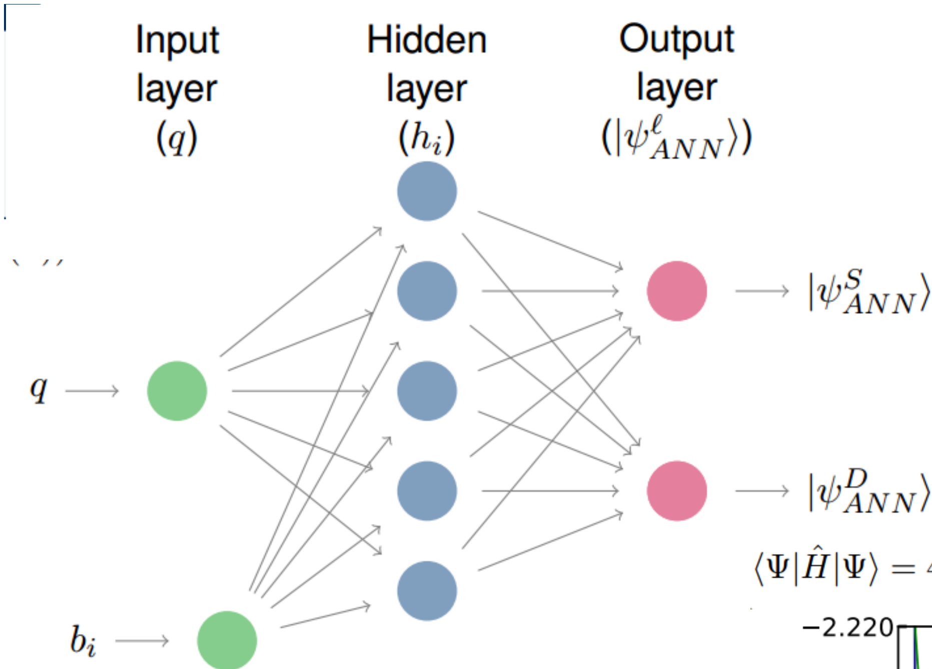
$$p(y^*|y) = \sum_{k=1}^{K} p(y^*|y, \mathcal{M}_k)p(\mathcal{M}_k|y)$$

unknown data



Phys. Rev. C 101, 044307 (2020)

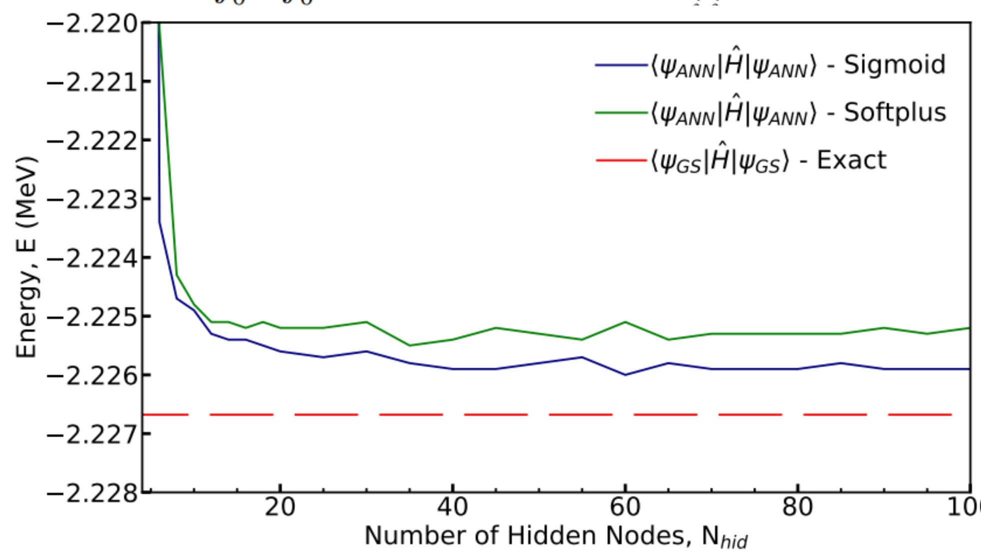
- **Machine learning of wavefunction of deutron**



Keeble, J. W. T., and A. Rios.
"Machine learning the deuteron."
PLB 809 (2020): 135743

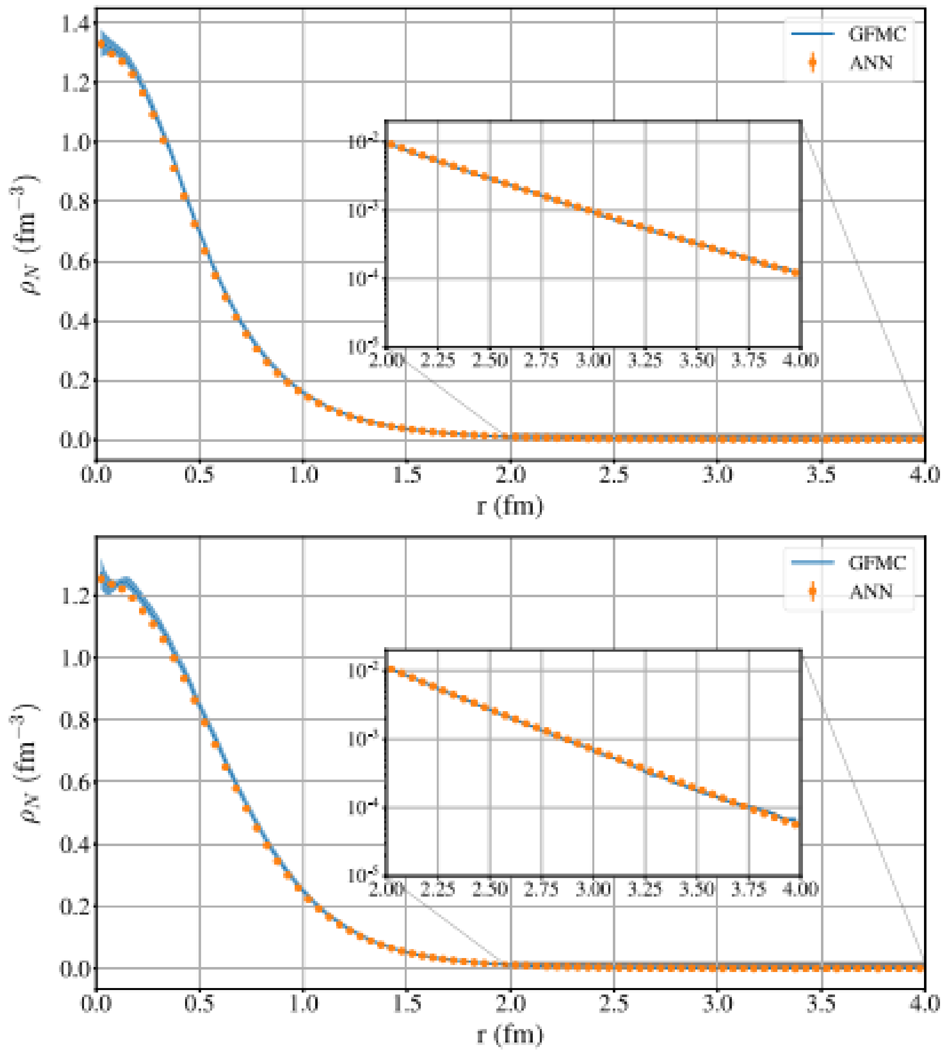
$$\langle\Psi|\hat{H}|\Psi\rangle = 4\pi \int_0^{\infty}\int_0^{\infty} q^2\Psi\hat{H}q'^2\Psi\, dq\, dq' \approx 4\pi \sum_{i,j} w_i q_i^2 \Psi_i \hat{H}_{ij} w_j q_j'^2 \Psi_j$$

$$E^{\mathscr{W}} = \frac{\langle \Psi^{\mathscr{W}}_{\rm ANN} | \hat{H} | \Psi^{\mathscr{W}}_{\rm ANN} \rangle}{\langle \Psi^{\mathscr{W}}_{\rm ANN} | \Psi^{\mathscr{W}}_{\rm ANN} \rangle} .$$

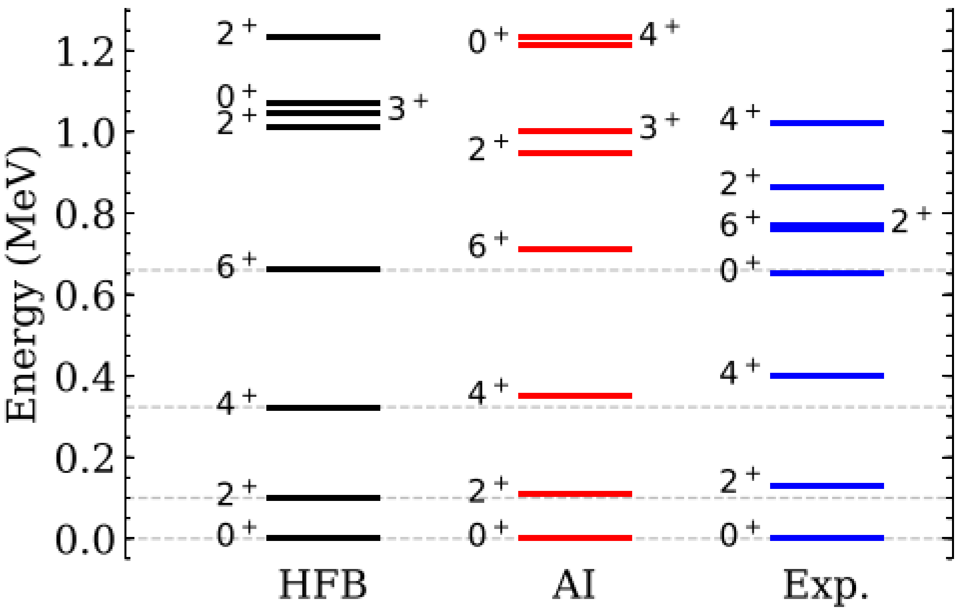


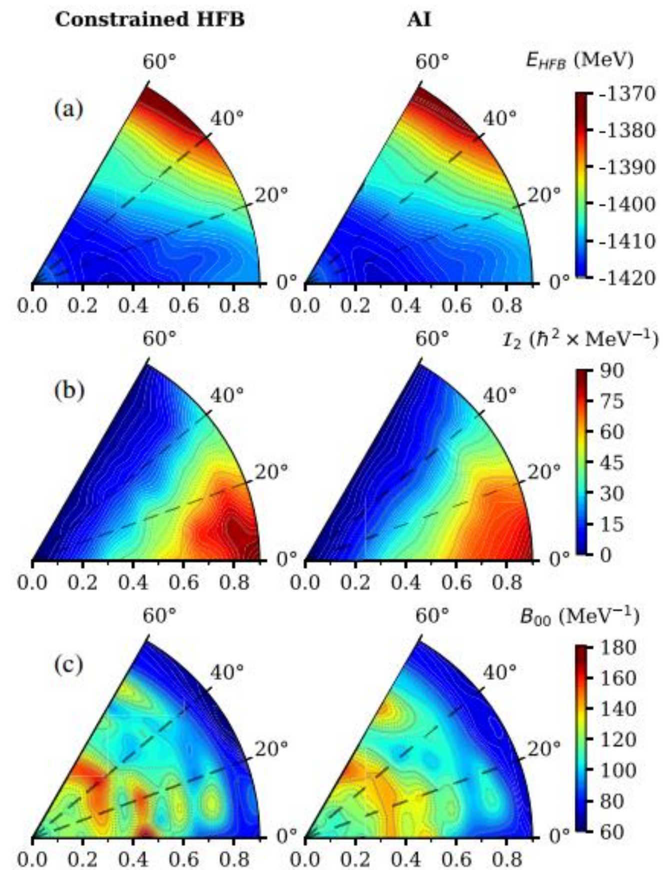
- **Variational Monte Carlo Calculations of A ≤ 4 Nuclei with an Artificial Neural-Network Correlator Ansatz**

PRL 127, 022502 (2021)



- **Learning different aspects of structures from DFT**



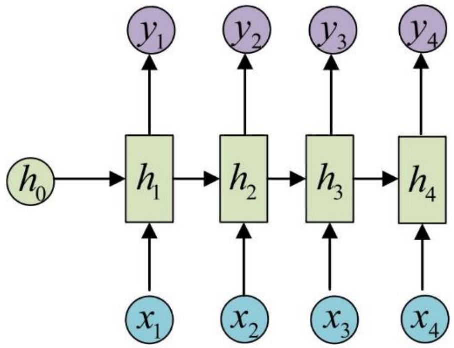


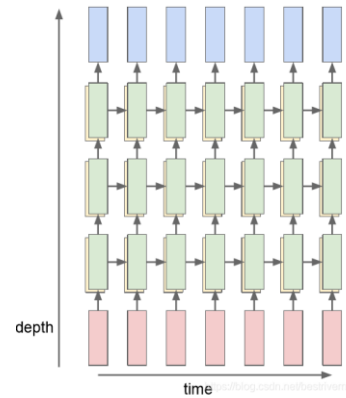
PRL 124, 162502 (2020)

Other networks

- RNN: Recurrent Neural Network 循环神经网络

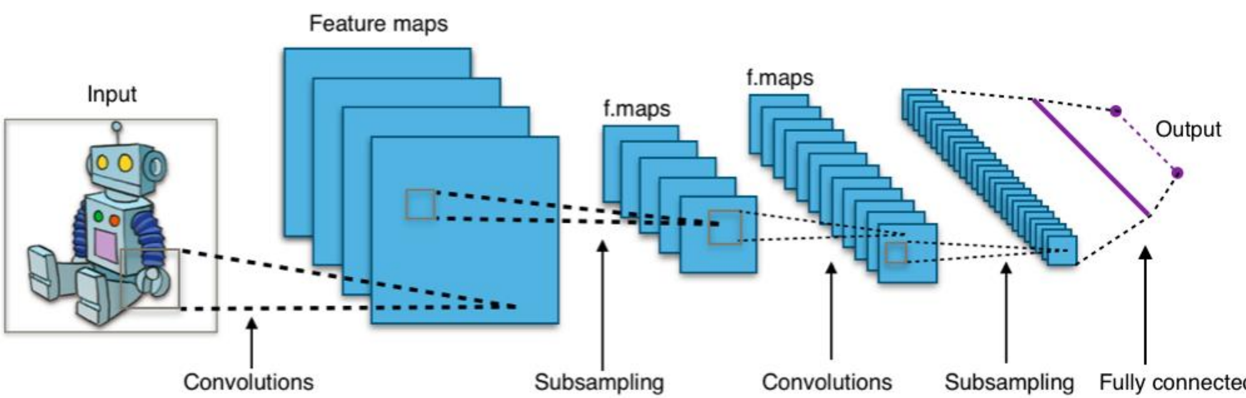
网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点有连接的，序列相关数据





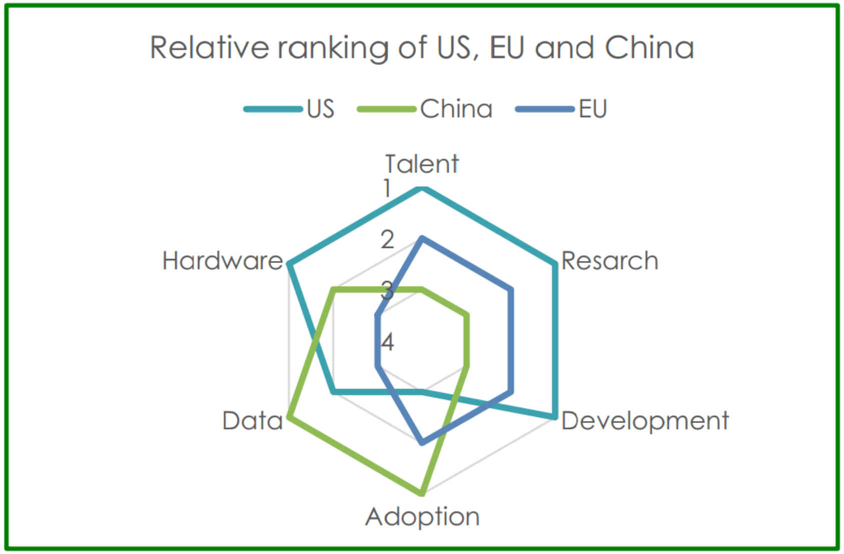
- CNN：卷积神经网络Convolutional Neural Networks， 网格化数据

图像的特征提取

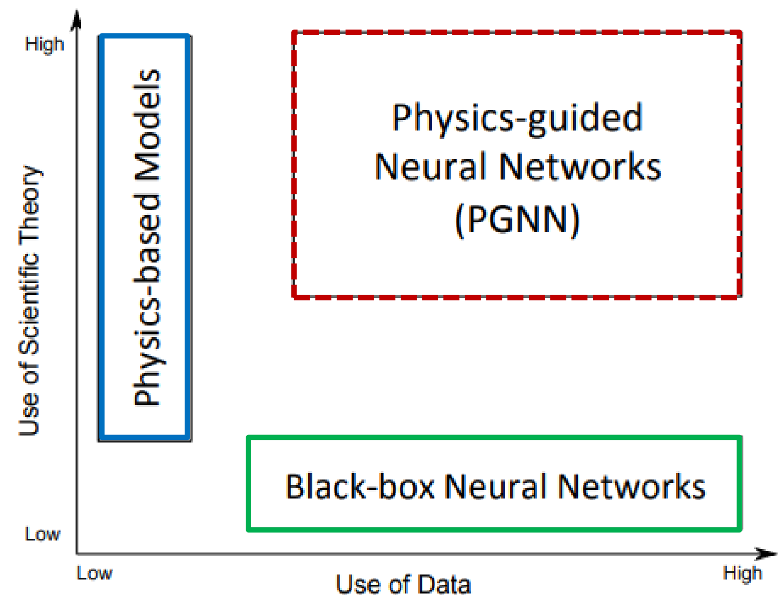


$$\int_{-\infty}^{\infty} f(\tau)g(x-\tau)d\tau$$

- **Challenges**



To merge with physics

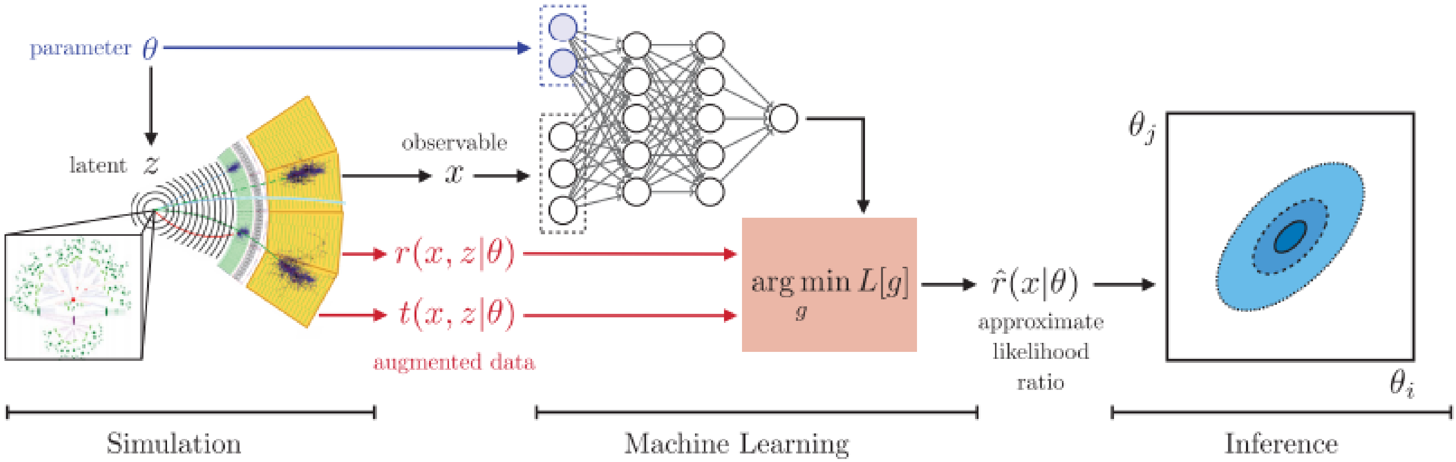


A. Karpatne, W. Watkins, J. Read, V. Kumar, arXiv:1710.11431

Build physics into priors

$$p(\mathbf{Q}|\mathbf{D}) \propto p(\mathbf{D}|\mathbf{Q},\mathbf{I})p(\mathbf{Q}|\mathbf{I})p(\mathbf{I}) \propto p(\mathbf{D}|\mathbf{Q})p(\mathbf{Q}|\mathbf{I})p(\mathbf{I}).$$

- Machine learning assisted simulation, simulation coupled with machine learning



Rev. Mod. Phys 91, 2019, 045002

Summary

- AI and machine learning impact all aspects of our lives
- Uncertainty evaluation, inference and interpolation
- Emulation for big simulations
- Bayesian model mixing
- Data fusion
- Inverse problems
- event selection and identification
- Identifying crucial experimental data , Providing meaningful input to measurements
- Experimental design: Efficient exploration of complex space

- Model reduction, model calibration, Parameter estimation
- Discovery with unsupervised learning

Outlook:

- Build physics-informed or physics-guide, physics-constrained machine learning
- Discover new physics from data
- Be careful of false correlation and Noise accumulation
- Chinese community for AI research in nuclear physics

Thank you for your attention!