

# 机器学习基本知识和基本工具

李 刚

中国科学院高能物理研究所

第一届机器学习和量子计算冬季学校

2025.01.12-18 南开大学

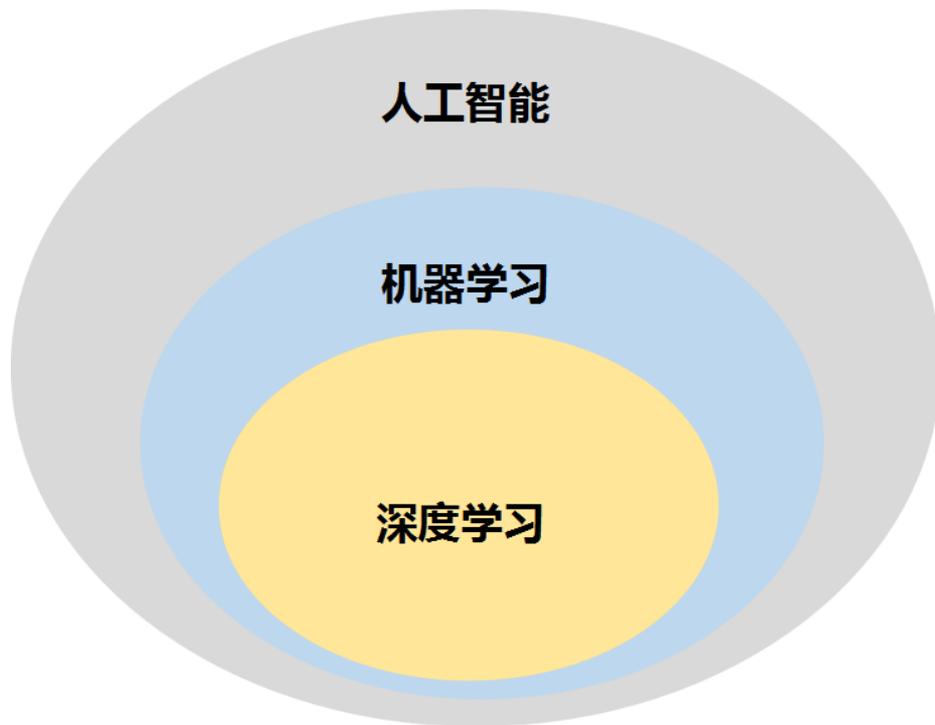
# Disclaimers

- This is a very personal review, highly biased
- Lots of thing not covered

# 提纲

- 什么是机器学习
- 关于机器学习我们应该知道的
- 机器学习和深度学习
- 机器学习算法和工具
- 两个例子
- 总结

# 什么是机器学习？



- 一套规则
- 无需编程
- 可以学习数据

- ✓ 统计学习
- ✓ 高维问题, 降维
- ✓ 问题的特殊性: 对称性

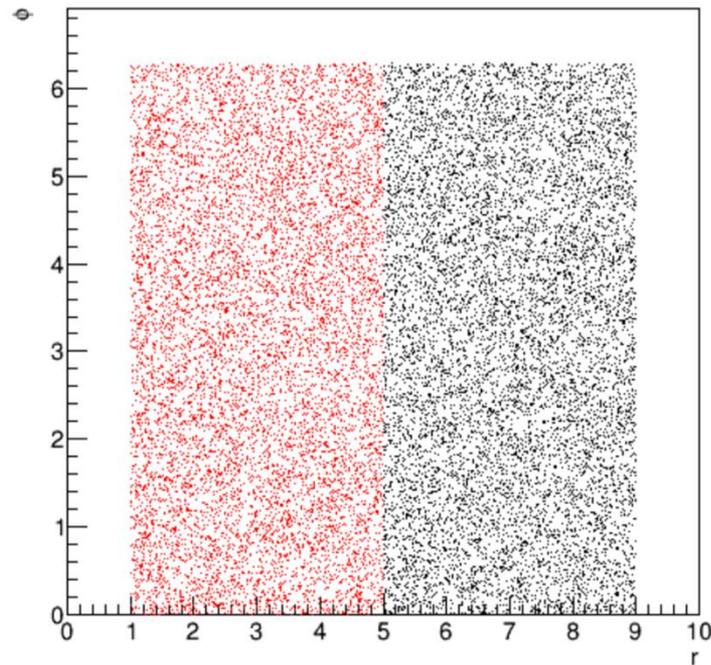
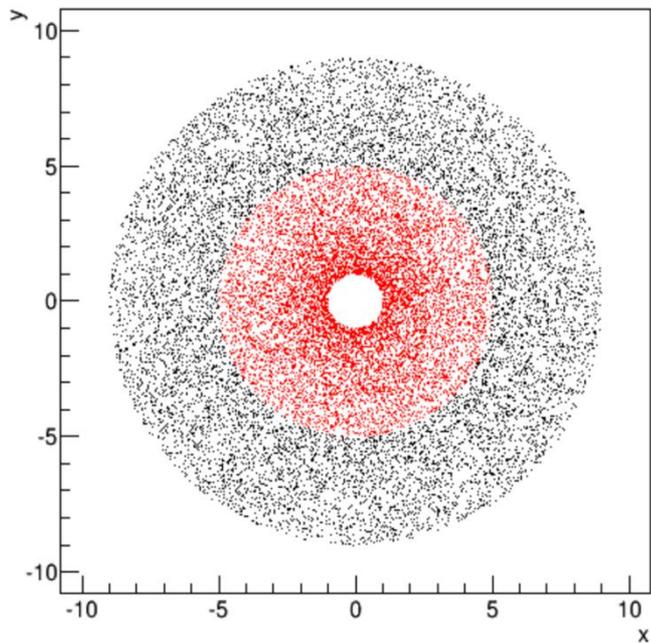
- ✓ A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$
- ✓ Machine learning is a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data or other outcomes of interest

# 什么是机器学习？

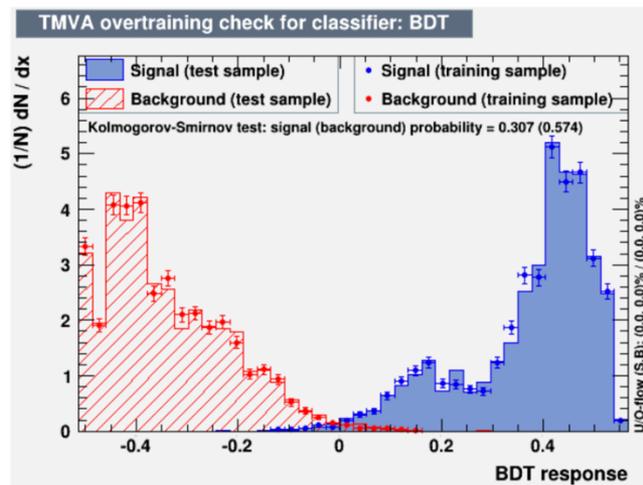
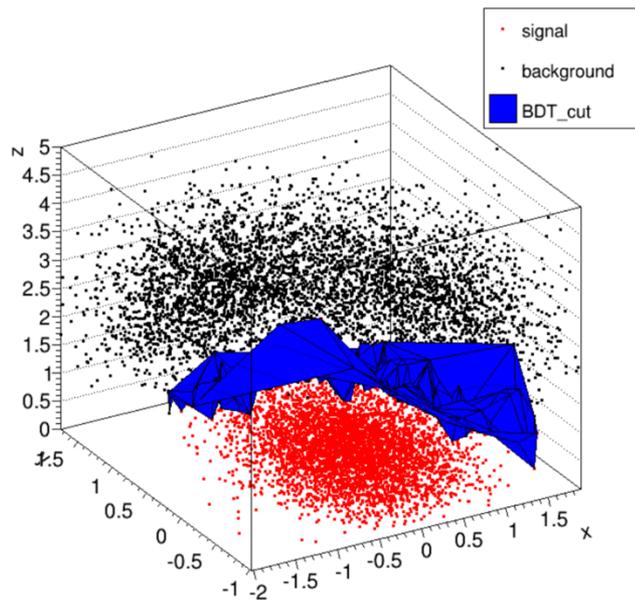
- 机器学习：不需要编程，或者只需要极少控制代码，可以通过数据学习（训练）“经验”并应用到新的数据上（测试、应用）。
- 可以用来做：分类，回归，统计推断，异常探测等
- 高能物理的机器学习叫“MVA”，采用的软件工具叫做“TMVA”，包含了大部分传统机器学习算法：BDT, SVM, 神经网络 ...
- 深度学习是机器学习的一部分：特征是网络模型的层数和复杂度大幅增加，模型的表达能力也大幅提升， ...
- 写代码是一种高级脑力劳动，如果能避免或者减少，将会大大提高效率和效果 ...

# 例子：简单分类问题

## 线性 到 非线性

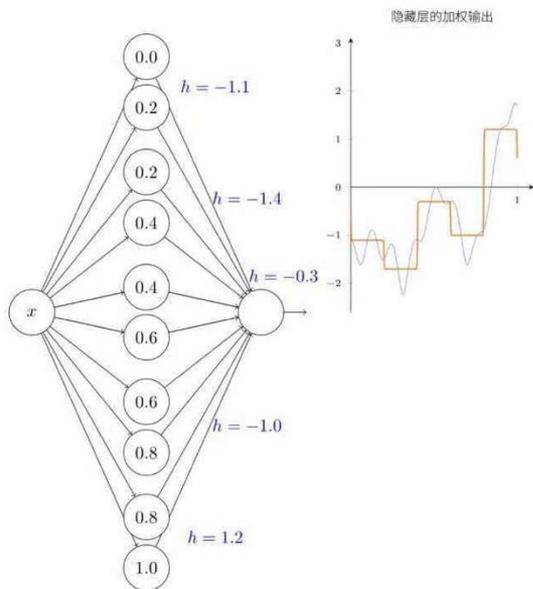


# 粒子：分类问题

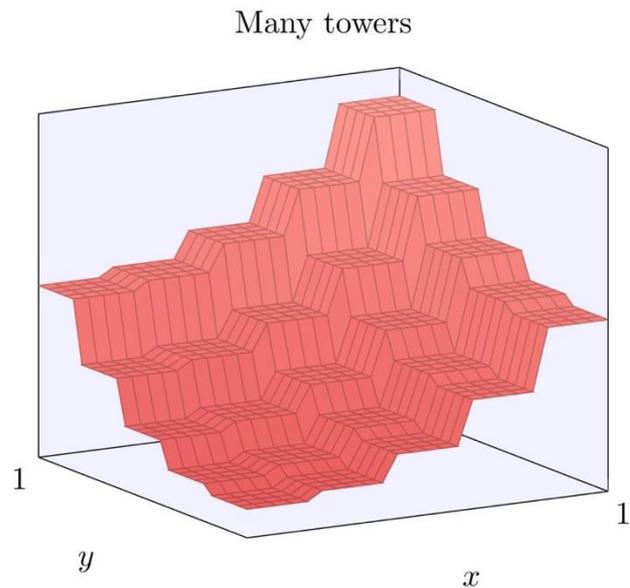


关于机器学习，我们需要知道什么？

# 事实 1: 为什么能? Neural network as universal function approximator



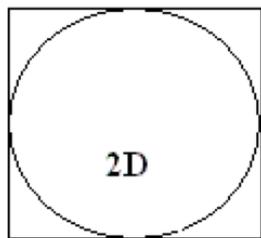
1D



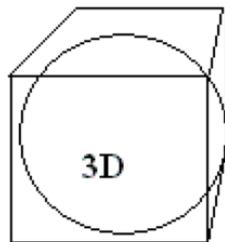
2D

A notable fact about neural networks is that they can approximate a continuous function to any desired level of precision, provided that there are enough neurons in the hidden layers.

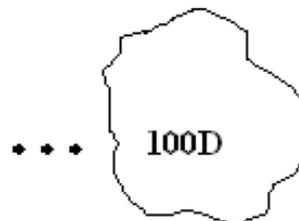
## 事实 2 :为什么难? Curse of dimensionality



ratio:  $4/\pi = 1.27$

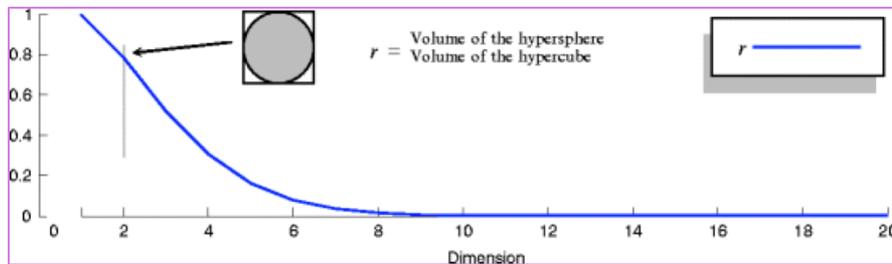


ratio:  $6/\pi = 1.91$



ratio:  $4.2 \cdot 10^{39}$

$$\frac{A_{circle}}{A_{square}} = \frac{\pi}{4} \text{ for } d = 2$$
$$\frac{V_{sphere}}{V_{cube}} = \frac{\pi}{6} \text{ for } d = 3$$
$$\frac{V_{hypersphere}}{V_{hypercube}} = \frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)} \rightarrow 0 \text{ as } d \rightarrow \infty$$

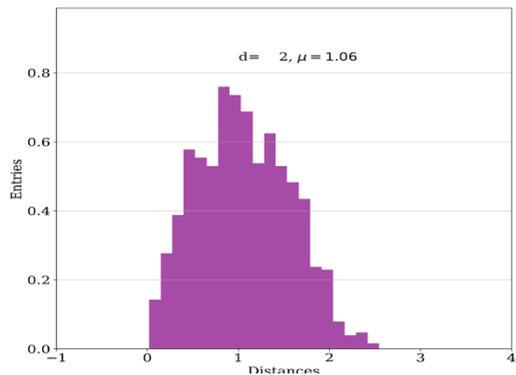


- When  $D=1$ : 100 evenly distributed points can sample a unit interval with a distance no greater than 0.01;
- When  $D=10$ : it requires  $10^{20}$  sampling points to achieve the same sampling rate.
- Almost all points in high- $D$  are isolated

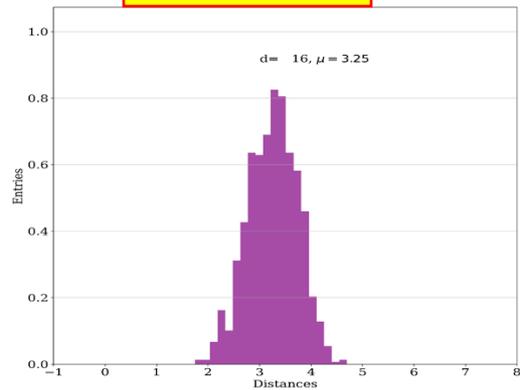
On one hand: fortunately, most specific problems can be reduced in dimensionality!

# (高维) 空间中的两点距离

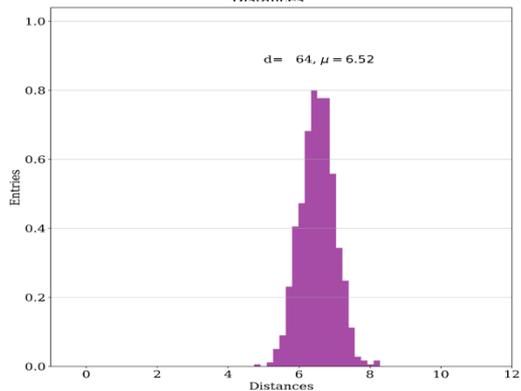
$D=2, d=1.06$



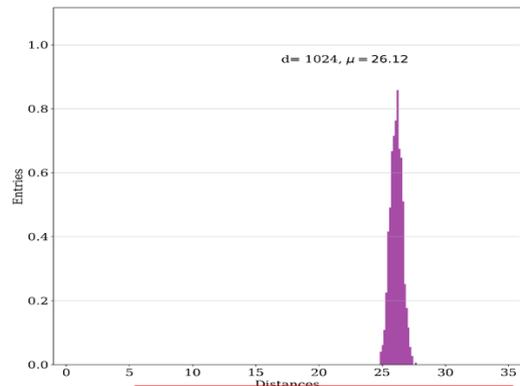
$D=16, d=3.25$



$D=64, d=6.5$



$D=1024, d=26.12$



# 推荐大家涉猎一点高维统计和高维概率

- ✓ 高维空间异常空旷，各个位置之间非常 isolated
  - Lost in the immensity of high dimensional spaces
- ✓ 统计涨落会累积
- ✓ 稀有事例的积累会变得不再“稀有”
- ✓ 高维空间的概率具有“concentration”的现象
- ✓ 高维度带来计算复杂性
- ✓ ... ..

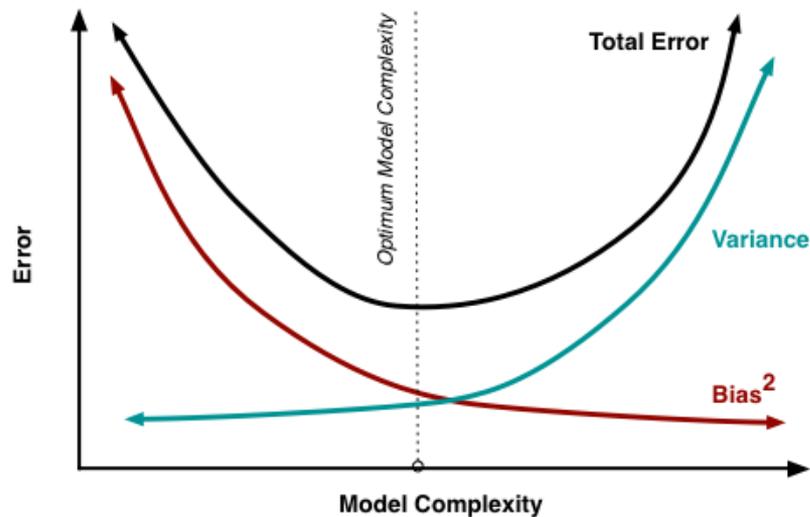
Neural networks have demonstrated their ability to effectively address the dimension problem!

## 事实 3:为什么要选择算法? bias-variance tradeoff

- 机器学习模型的目标是最小化预测误差。
- 预测误差可以分解为偏差、方差和不可减少的误差。
- Bias: 模型预测值与真实值之间的差异。高偏差导致模型在训练集上的表现不佳。
- Variance :模型在不同数据集上的预测结果的波动。高方差导致模型在新数据上的泛化能力差

$$\text{Total Error}^2 = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}^2$$

- ✓ 理解偏差和方差的关系对于构建有效模型至关重要。
- ✓ 持续调整模型以找到最佳平衡点。



## 事实 4: 算法没有好坏 !?

No free lunch theorem ( <http://www.no-free-lunch.org> )

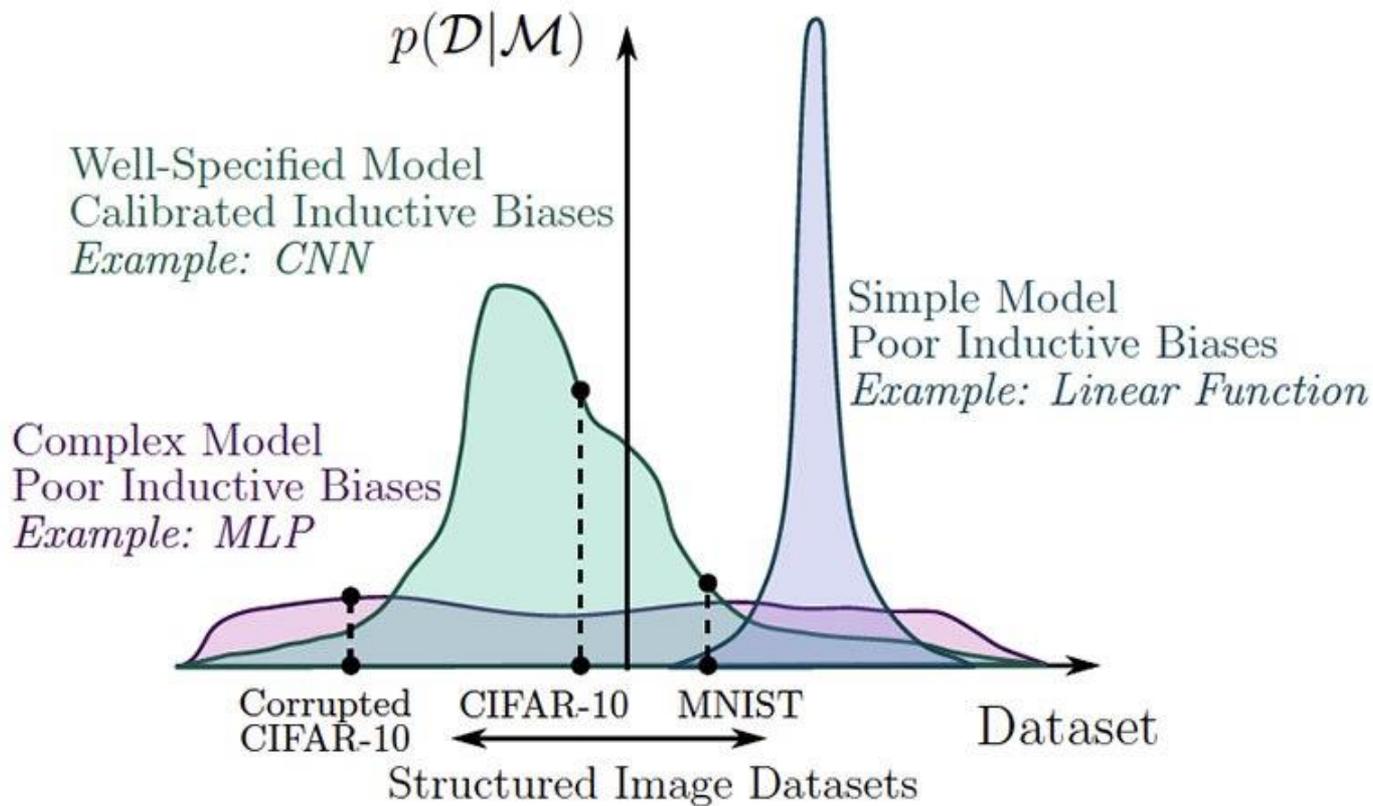
There is no single algorithm that is universally the best for all problems

Performance of a learning algorithm is problem-specific

算法没有好坏，但对具体问题有“偏好 (bias) ”

- 这是个严肃的数学定理
- 有证明
- 假设：数据集是全集
- 但具体问题的数据一个子集
- 关键：合适的算法



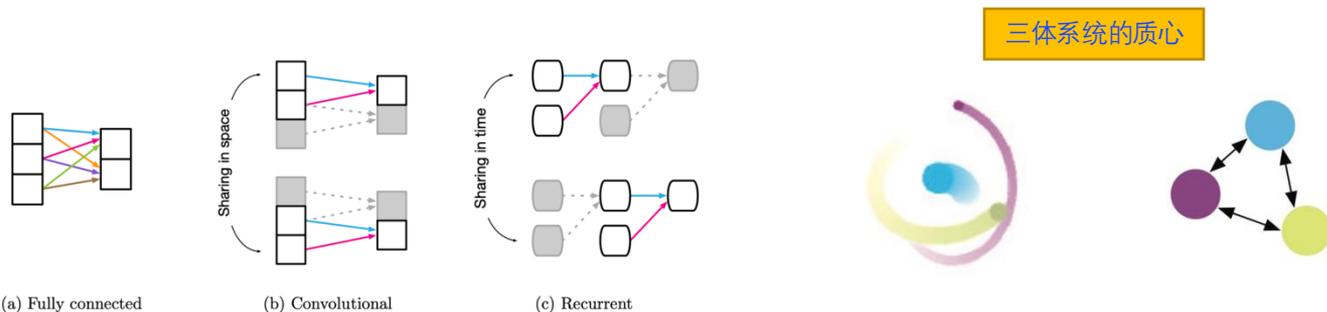


Why do some models perform well on certain datasets? Inductive bias

## 选择合适的算法

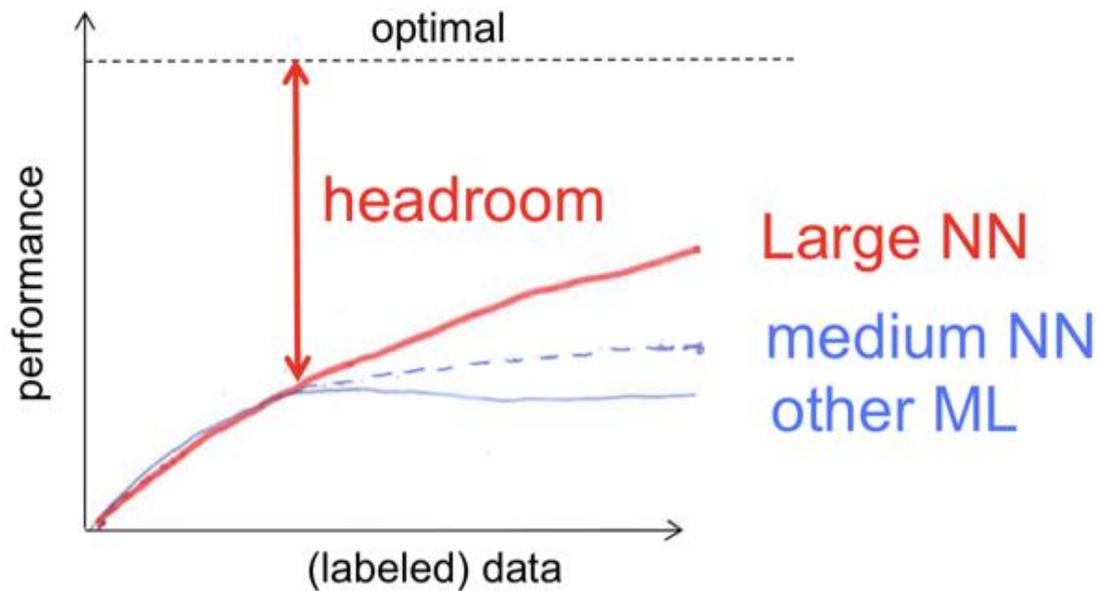
- ✓ NFL定理指出学习是不可能的，除非有先验知识。
- ✓ 通常情况下我们不知道上帝函数，但猜测它属于一个较小的假设类别。
- ✓ 这种基于先验知识对目标模型的判断就是 inductive bias —— 归纳偏好。归纳偏好所做的事情就是将无限可能的目标函数约束在一个有限的假设类别当中。
- ✓ 如果给出更加宽松的模型假设，也即用更弱的 inductive bias，那我们更有可能得到强力模型 —— 更接近目标函数  $F$ ，但是训练变得非常困难，乃至不可能。
- ✓ 学习者的归纳偏好是一组额外的假设，保证其归纳推理接近演绎的推论。

# 不同算法的归纳偏好比较

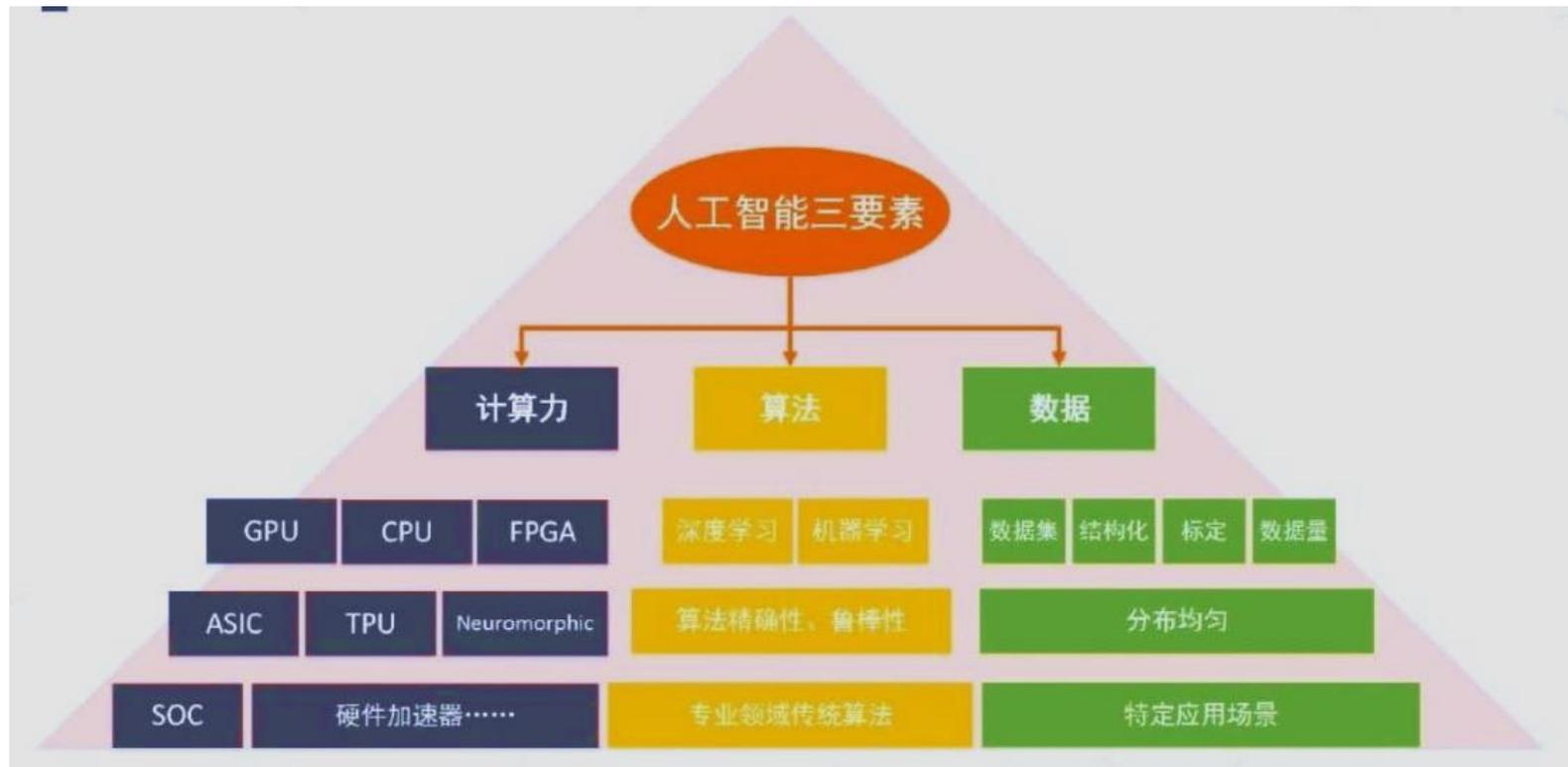


类型	输入形式	关系	偏好	变换	评论
DNN	单位	All-to-all	弱	---	信息无重用，无孤立
CNN	均匀像素	局域	取决于局域性	空间平移	局域特征和平移不变性
RNN	均匀时间步长	序列	取决于序列性	时间平移	信息重用+时间平移不变
GNN	节点	边	变化大	点、边的交换	顺序无关，IB来自某种东西的“无” (absence)

# 机器学习和深度学习



# 三要素：数据、算法、算力



# 数据

## ■ 数据依赖

- 机器学习：对数据量要求相对较少，较小的数据集即可进行有效训练和模式识别。
- 深度学习：需要大量的数据来训练，数据量越大，模型性能越好，因为其通过大量数据自动学习特征。

## ■ 特征工程

- 机器学习：需要手动进行特征工程，由领域专家提取和选择特征，以简化数据复杂性并使模式更明显。
- 深度学习：自动学习和提取特征，通过多层神经网络架构直接从原始数据中学习。

# 算法

## ■ 学习方法

- 机器学习：基于规则和统计方法，通常将问题分解为多个子任务，分别用不同算法解决后再组合结果。
- 深度学习：基于神经网络的学习方法，采用端到端的方式，使用单一模型从输入到输出。

## ■ 输出结果

- 机器学习：通常产生单一输出。
- 深度学习：可以产生多个输出或层次化的表示。

## ■ 算法与可解释性

- 机器学习：算法相对简单，可解释性强，如决策树等模型的逻辑容易理解。
- 深度学习：算法复杂，可解释性差，常被视为“黑箱”模型，内部工作机制不清晰。

## ■ 复杂任务表现

- 机器学习：在处理高度复杂的任务时可能会遇到困难。
- 深度学习：在复杂领域如计算机视觉和自然语言处理中表现出色，能够取得最先进的成果。

## ■ 可扩展性

- 机器学习：在可扩展性方面可能存在一定限制。
- 深度学习：具有很高的可扩展性，能够有效处理大规模和复杂的问题。

# 算力

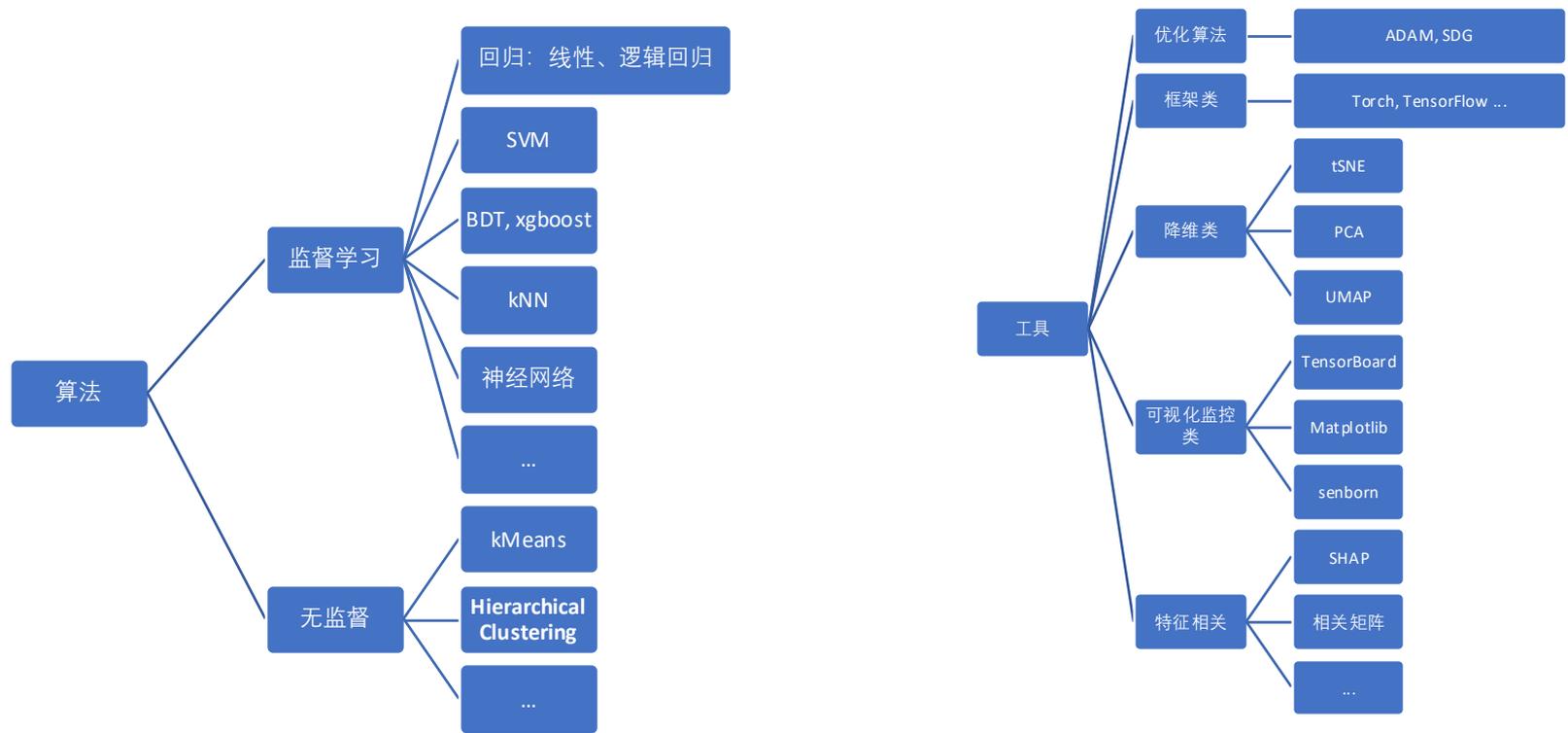
## ■ 硬件需求

- 机器学习：对硬件要求较低，通常可以在标准计算机上高效运行。
- 深度学习：计算密集型，通常需要高性能的硬件，如GPU、TPU等，以加速训练过程。

## ■ 训练时间

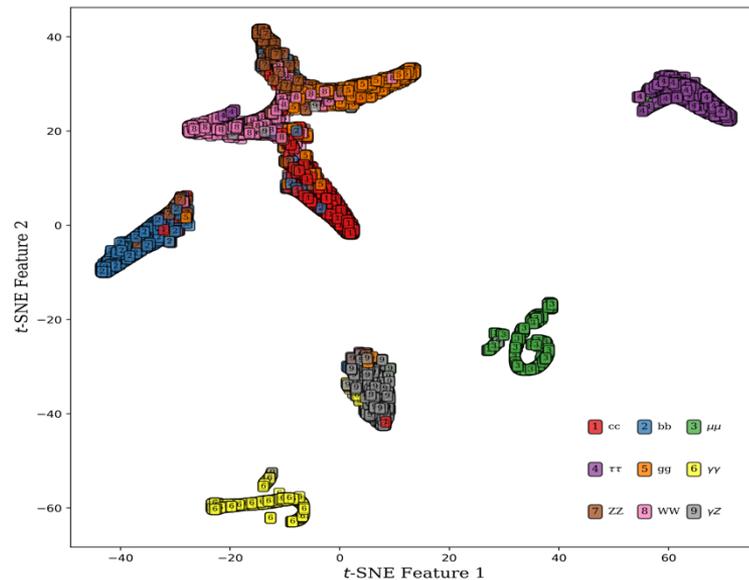
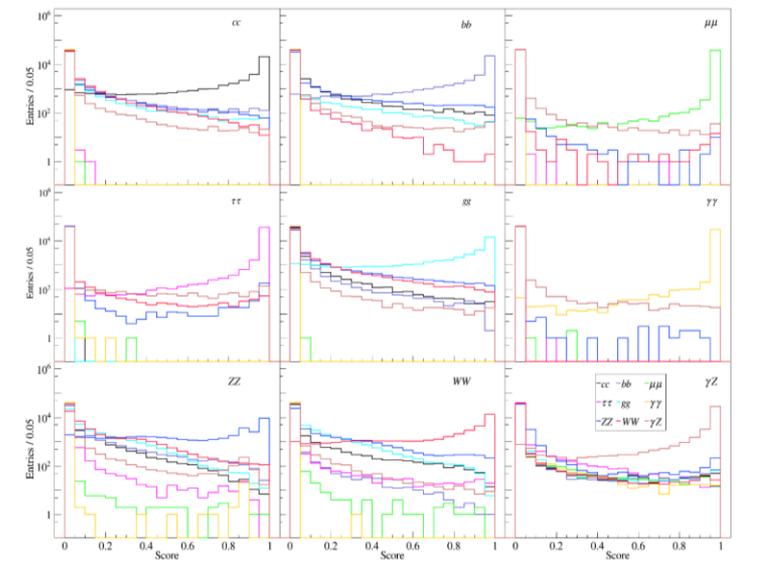
- 机器学习：训练时间相对较短，通常在几分钟到几小时内完成。
- 深度学习：由于模型复杂且数据量大，训练时间较长，可能需要数天甚至数周。

# 机器学习算法和工具



# tSNE 降维的例子

9 维  $\rightarrow$  2 维



# 关于“confusion matrix” and beyond

- Data sample containing instances of two classes:  $N_{tot} = S_{tot} + B_{tot}$ 
  - HEP: signal  $S_{tot} = S_{sel} + S_{rej}$
  - HEP: background  $B_{tot} = B_{sel} + B_{rej}$
- Discrete binary classifiers assign each instance to one of the two classes
  - HEP: classified as signal and selected  $N_{sel} = S_{sel} + B_{sel}$
  - HEP: classified as background and rejected  $N_{rej} = B_{rej} + S_{rej}$

	<u>true class</u> : Positives + (HEP: signal)	<u>true class</u> : Negatives - (HEP: background)
<u>classified as</u> : positives (HEP: selected)	<b>True Positives (TP)</b> (HEP: selected signal <b>Ssel</b> )	<b>False Positives (FP)</b> (HEP: selected bkg <b>Bsel</b> )
<u>classified as</u> : negatives (HEP: rejected)	<b>False Negatives (FN)</b> (HEP: rejected signal <b>Srej</b> )	<b>True Negatives (TN)</b> (HEP: rejected bkg <b>Brej</b> )

T. Fawcett, *Introduction to ROC analysis*, Pattern Recognition Letters 27 (2006) 861. doi:10.1016/j.patrec.2005.10.010

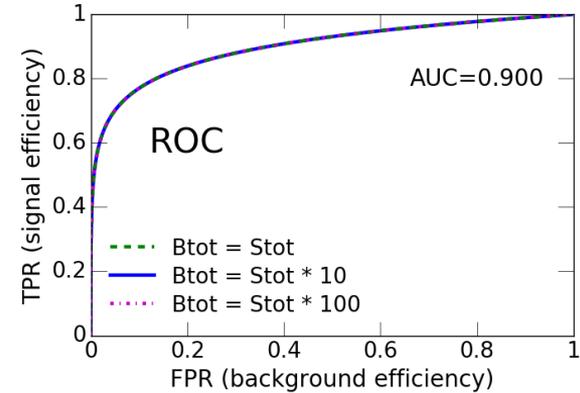
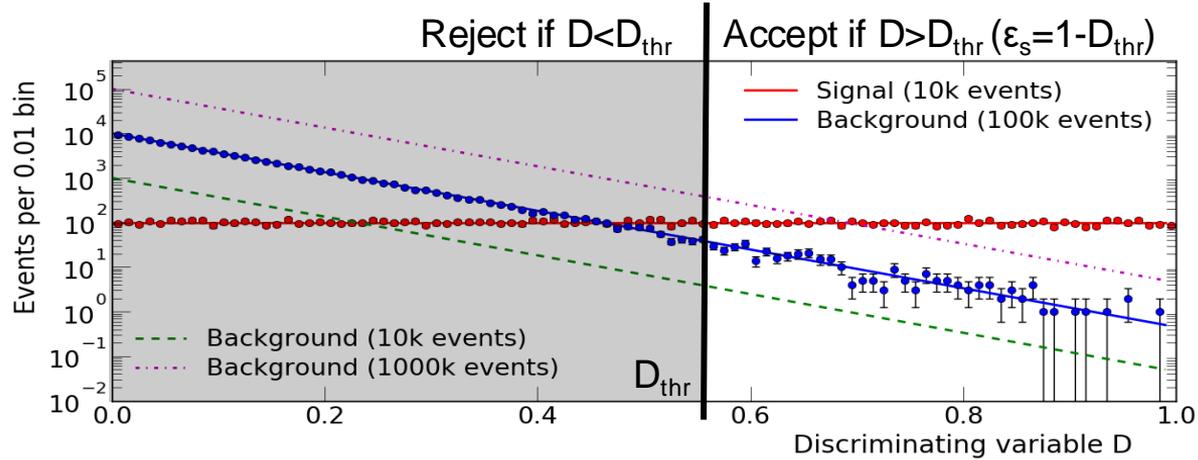
# The confusion about the confusion matrix...

Different domains → focus on different concepts → different terminologies

<table border="1"> <tr> <td>TP (<math>S_{sel}</math>)</td> <td>FP (<math>B_{sel}</math>)</td> </tr> <tr> <td>FN (<math>S_{rej}</math>)</td> <td>TN (<math>B_{rej}</math>)</td> </tr> </table>	TP ( $S_{sel}$ )	FP ( $B_{sel}$ )	FN ( $S_{rej}$ )	TN ( $B_{rej}$ )	<table border="1"> <tr> <td>TP (<math>S_{sel}</math>)</td> <td>FP (<math>B_{sel}</math>)</td> </tr> <tr> <td>FN (<math>S_{rej}</math>)</td> <td>TN (<math>B_{rej}</math>)</td> </tr> </table>	TP ( $S_{sel}$ )	FP ( $B_{sel}$ )	FN ( $S_{rej}$ )	TN ( $B_{rej}$ )	<table border="1"> <tr> <td>TP (<math>S_{sel}</math>)</td> <td>FP (<math>B_{sel}</math>)</td> </tr> <tr> <td>FN (<math>S_{rej}</math>)</td> <td>TN (<math>B_{rej}</math>)</td> </tr> </table>	TP ( $S_{sel}$ )	FP ( $B_{sel}$ )	FN ( $S_{rej}$ )	TN ( $B_{rej}$ )
TP ( $S_{sel}$ )	FP ( $B_{sel}$ )													
FN ( $S_{rej}$ )	TN ( $B_{rej}$ )													
TP ( $S_{sel}$ )	FP ( $B_{sel}$ )													
FN ( $S_{rej}$ )	TN ( $B_{rej}$ )													
TP ( $S_{sel}$ )	FP ( $B_{sel}$ )													
FN ( $S_{rej}$ )	TN ( $B_{rej}$ )													
$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$	$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}$	$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} = 1 - \text{FPR}$												
<p>HEP: “efficiency”</p> $\epsilon_s = \frac{S_{sel}}{S_{tot}}$	<p>HEP: “purity”</p> $\rho = \frac{S_{sel}}{S_{sel} + B_{sel}}$	<p>HEP: “background rejection”</p> $1 - \epsilon_b = 1 - \frac{B_{sel}}{B_{tot}}$												
<p>IR: “recall”</p>	<p>IR: “precision”</p>	<p>—</p>												
<p>MED: “sensitivity”</p>	<p>—</p>	<p>MED: “specificity”</p>												

Information Retrieval: 信息检索  
MED: 医疗

# Discrete vs. Scoring classifiers – ROC curves



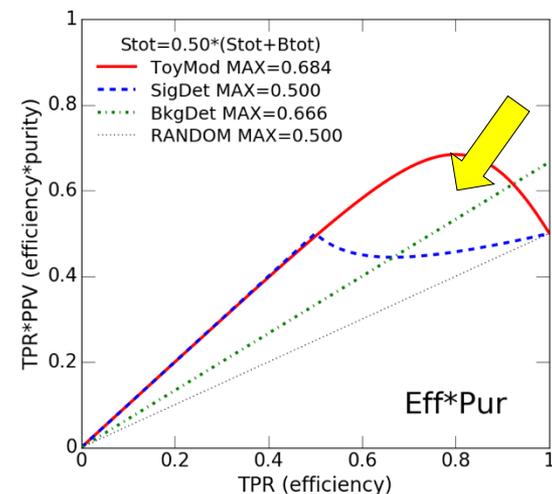
- Discrete classifiers → either select or reject → confusion matrix
- Scoring classifiers → assign score  $D$  to each event (e.g. BDT)
  - ideally related to likelihood that event is signal or background
  - from scoring to discrete: choose a threshold → classify as signal if  $D > D_{thr}$
- ROC curves describe how  $FPR(\epsilon_b)$  and  $TPR(\epsilon_s)$  are related when varying  $D_{thr}$

# A simple HEP example

- *Measurement of a total cross-section  $\sigma_s$  in a counting experiment*
- To minimize statistical errors: **maximise  $\epsilon_s * \rho$**  (well-known since decades)
  - *global efficiency  $\epsilon_s = S_{sel}/S_{tot}$  and global purity  $\rho = S_{sel}/(S_{sel} + B_{sel})$  – “1 single bin”*

$$\frac{1}{(\Delta\sigma_s)^2} = \frac{1}{\sigma_s} \mathcal{L} \epsilon_s \rho = \frac{1}{\sigma_s^2} S_{tot} \epsilon_s \rho$$

- To compare classifiers (red, green, blue, black):
  - in each classifier  $\rightarrow$  vary  $D_{thr}$  cut  $\rightarrow$  vary  $\epsilon_s$  and  $\rho$   
 $\rightarrow$  find maximum of  $\epsilon_s * \rho$  (choose “working point”)
  - chose classifier with maximum of  $\epsilon_s * \rho$  out of the four
- $\epsilon_s * \rho$ : metric between 0 and 1
  - qualitatively relevant: the higher, the better
  - numerically: fraction of Fisher information ( $1/error^2$ ) available after selecting
  - **correct metric only for  $\sigma_s$  by counting!**



# Different HEP problems → Different metrics

## Binary classifiers for HEP event selection (signal-background discrimination)

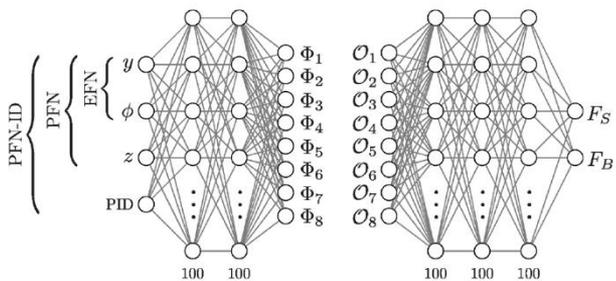
<b>Statistical error minimization</b>  <b>(or statistical significance maximization)</b>	<b>Cross-section (1-bin counting)</b>	2 variables: global $\epsilon_s, \rho$ (given $S_{tot}$ )	Maximise $S_{tot} * \epsilon_s * \rho$ (at any $S_{tot}$ )
	<b>Searches (1-bin counting)</b>	Simple – 2 variables: global $S_{sel}, B_{sel}$ (or equivalently $\epsilon_s, \rho$ )	Maximise $\frac{S_{sel}}{\sqrt{S_{sel} + B_{sel}}}$ (i.e. $\sqrt{S_{tot} * \epsilon_s * \rho}$ )
		HiggsML – 2 variables: global $S_{sel}, B_{sel}$	Maximise $\sqrt{2((S_{sel} + B_{sel}) \log(1 + \frac{S_{sel}}{B_{sel}}) - S_{sel})}$
		Punzi – 2 variables: global $\epsilon_s, B_{sel}$	Maximise $\frac{\epsilon_s}{A/2 + \sqrt{B_{sel}}}$
	<b>Cross-section (binned fits)</b>	2 variables: local $\epsilon_{s,i}$ and $\rho_i$ in each bin (given $s_{tot,i}$ in each bin)	Maximise $\sum_i s_{tot,i} * \epsilon_{s,i} * \rho_i$ Partition in bins of equal $\rho_i$
	<b>Parameter estimation (binned fits)</b>		Maximise $\sum_i s_{tot,i} * \epsilon_{s,i} * \rho_i * (\frac{1}{S_{tot,i}} \frac{\partial S_{tot,i}}{\partial \theta})^2$ Partition in bins of equal $\rho_i * (\frac{1}{S_{tot,i}} \frac{\partial S_{tot,i}}{\partial \theta})$

# (Multi-)Classification problem

- Jet tagging/W tagger
- Event classification

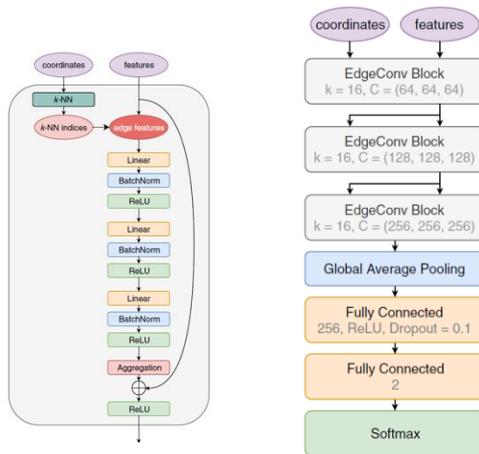
# Algorithms used

## Energy Flow Network(EFN) / Particle Flow Network(PFN)



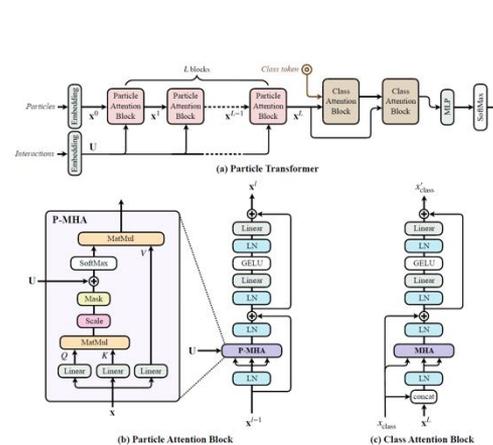
P. T. Komiske, E. M. Metodiev and J. Thaler  
[\[JHEP01\(2019\)121\]](#)

## ParticleNet



H. Qu and L. Gouskos [\[Phys.Rev.D 101 \(2020\) 5, 056019\]](#)

## ParticleTransformers (ParT)

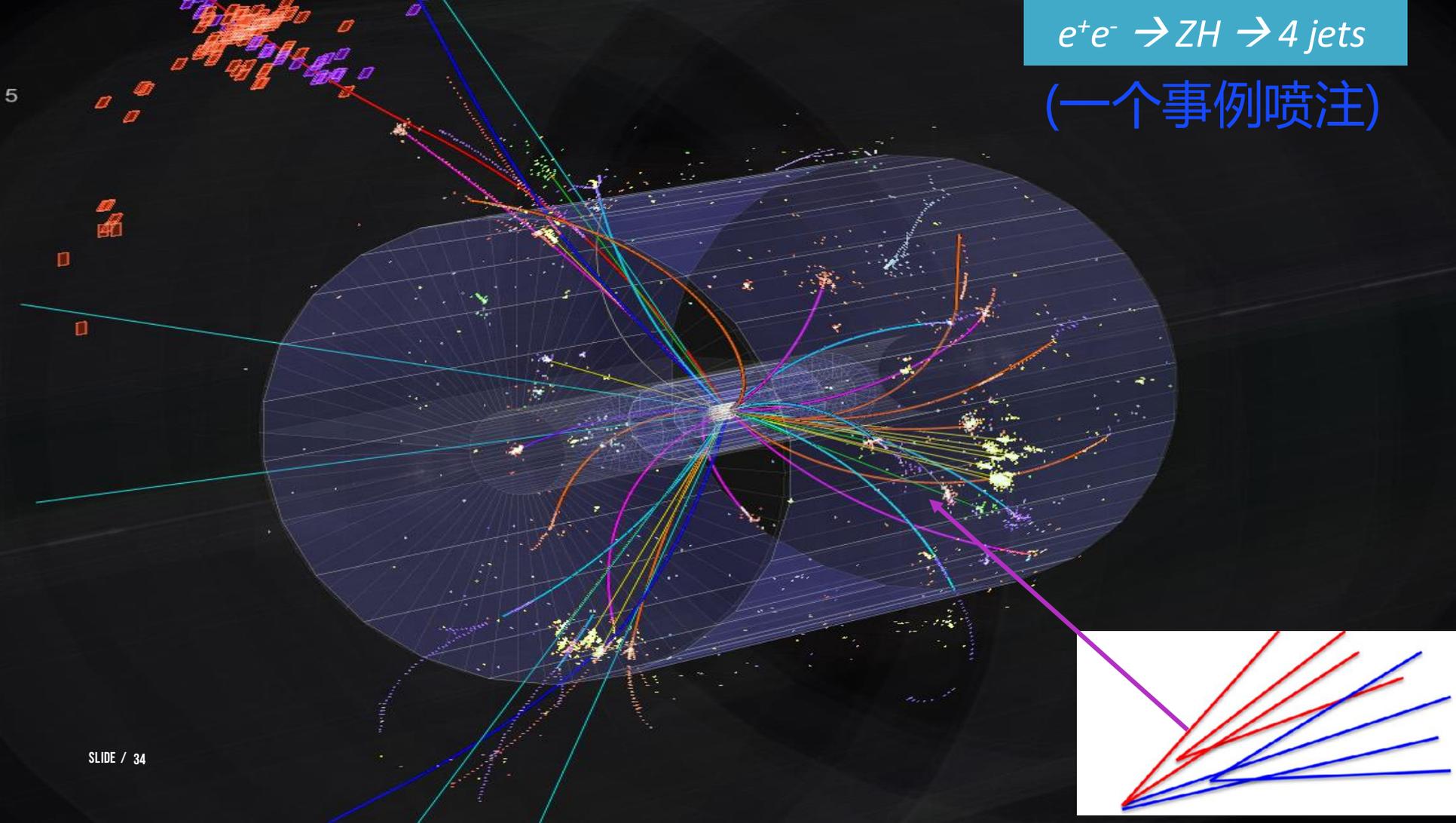


H. Qu, C. Li, S. Qian [\[2202.03772\]](#)



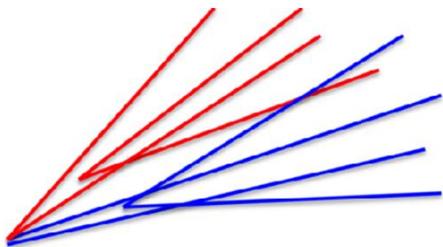
$e^+e^- \rightarrow ZH \rightarrow 4 \text{ jets}$

(一个事例喷注)



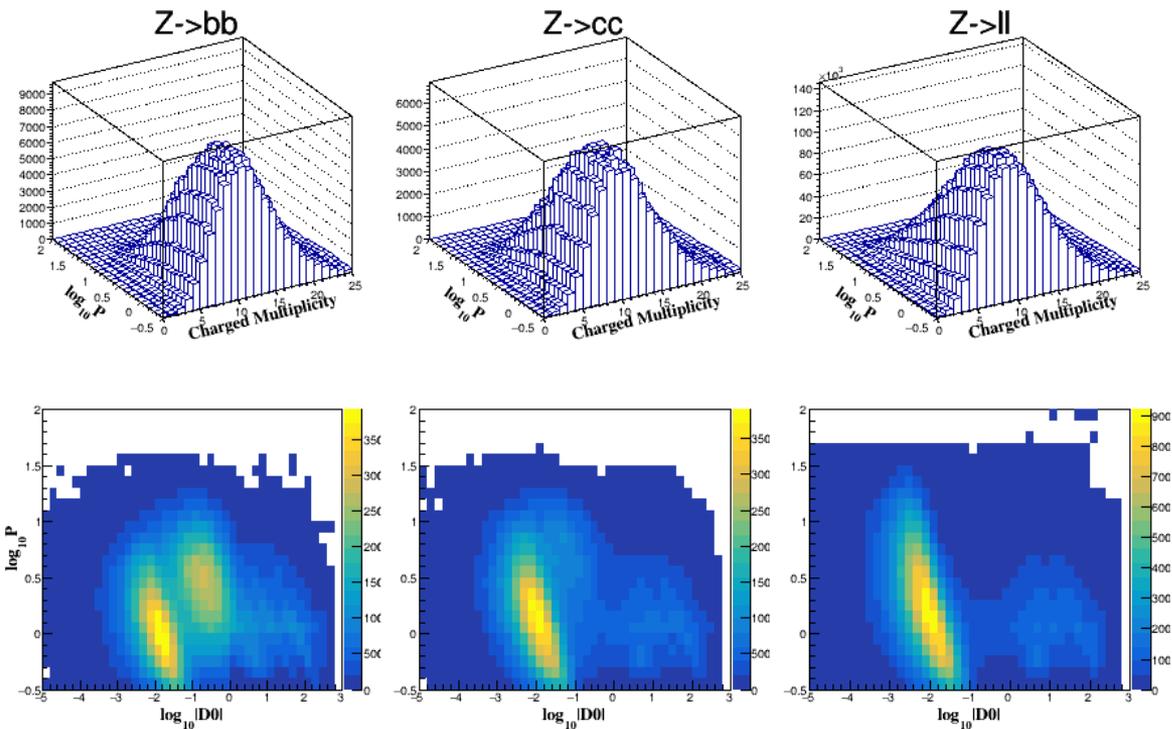
# Jet (flavor) tagging(单个喷注)

- 91 GeV
- $Z \rightarrow bb, cc, ll$  (uu,dd,ss)
- 450k events (900k jets) for each class
  
- Take particle level information as input

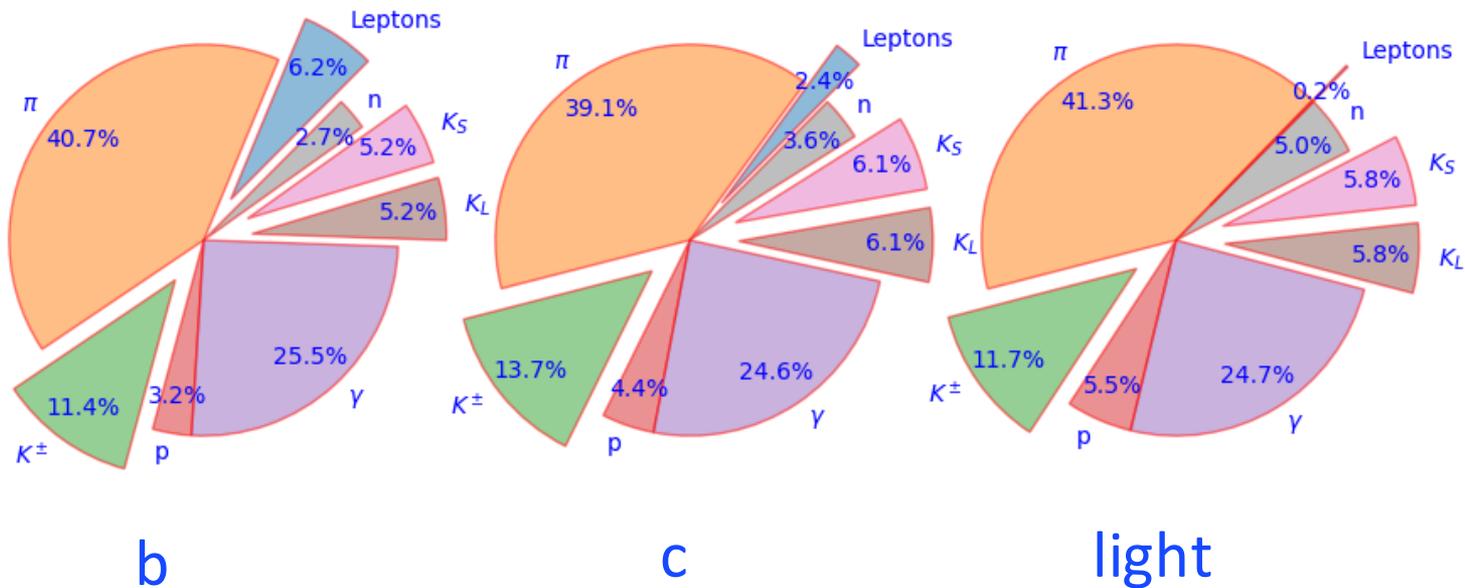


- 4-momenta
- $d_0/z_0$
- PID
- ... ..

# Multiplicity, impact parameters



# PID information



Weighted by momenta

Accuracy  $\longrightarrow$

Algorithm	ParticleNet	PFN	DNN	BDT	GBDT	gforest	XGBoost
Accuracy	0.872	0.850	0.788	0.776	0.794	0.785	0.801

Purity  $\times$  efficiency  $\longrightarrow$

tag	$\epsilon_S(\%)$	$\epsilon \times \rho$			
		LCFIPlus	XGBoost	ParticleNet	PFN
<i>b</i>	60	-	-	0.589	0.596
	70	-	-	0.694	0.689
	80	-	0.747	0.780	0.763
	90	0.72	0.713	0.810	0.752
	95	-	0.609	0.721	0.645
<i>c</i>	60	0.36	-	0.548	0.485
	70	-	-	0.589	0.497
	80	-	0.345	0.584	0.467
	90	-	0.292	0.516	0.402
	95	-	0.251	0.451	0.348

Take c-tagging as example

$$\sqrt{0.584/0.345}=1.3$$

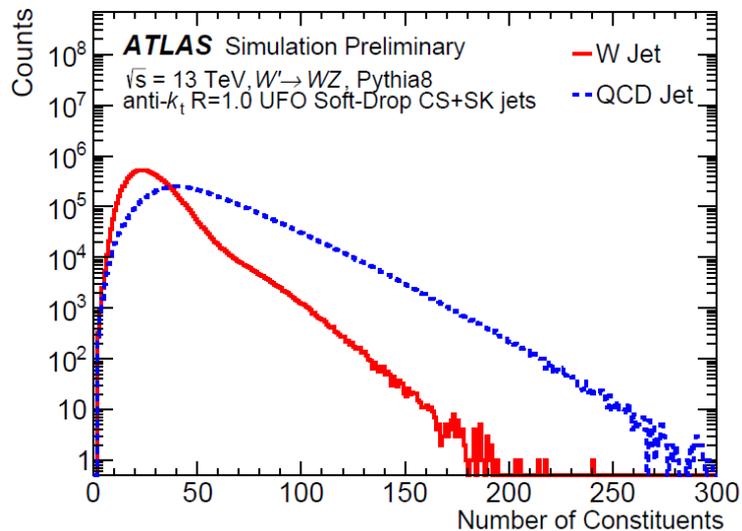
Statistical uncertainty: 30%  $\downarrow$

$$\frac{1}{(\Delta\sigma_s)^2} = \frac{1}{\sigma_s} \mathcal{L}_{\epsilon_s \rho} = \frac{1}{\sigma_s^2} S_{\text{tot}} \epsilon_s \rho$$

# W Jet Taggers (ATLAS, by Shudong Wang)

(一对喷注)

- In this study, a maximum of 200 constituents are considered by all constituent-based taggers. Only a small portion of jets in the dataset have more than 200 constituents (less than 0.04%). As jet constituents are sorted by decreasing  $p_T$ , truncation eliminates the softest constituents of the jet.



Distributions of the number of constituents in a large- $R$  jet.

# W Jet Taggers

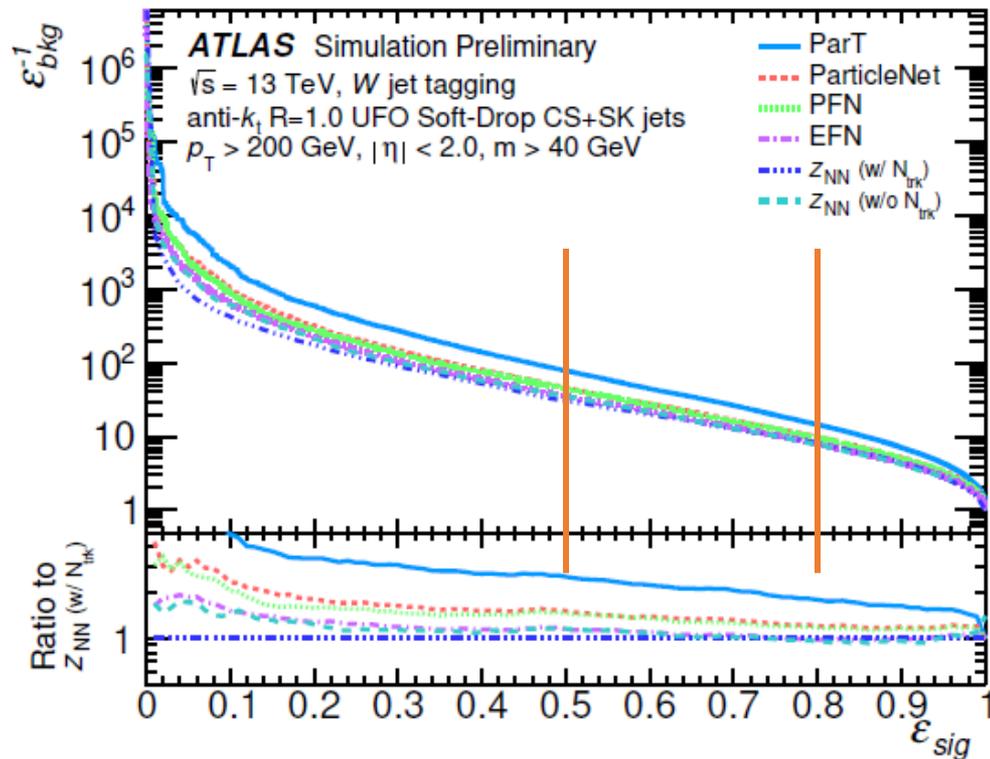
- Particle Flow Network(PFN)/Energy Flow Network(EFN)
  - Based on Deep Sets Theorem
  - [JHEP01\(2019\)121](#)
- ParticleNet
  - Customized graph neural network architecture for jet tagging with the point cloud approach
  - [Phys.Rev.D 101 \(2020\) 5, 056019](#)
- ParticleTransformer
  - Transformer designed for particle physics
  - [arxiv: 2202.03772](#)

Models	Input variables
EFN	$\Delta\eta, \Delta\phi, \ln p_T$
PFN	$\Delta\eta, \Delta\phi, \ln p_T, \ln E, \ln \frac{p_T}{\sum_{jet} p_T}, \ln \frac{E}{\sum_{jet} E}, \Delta R$
ParticleNet	$\Delta\eta, \Delta\phi, \ln p_T, \ln E, \ln \frac{p_T}{\sum_{jet} p_T}, \ln \frac{E}{\sum_{jet} E}, \Delta R$
ParticleTransformer	$\Delta\eta, \Delta\phi, \ln p_T, \ln E, \ln \frac{p_T}{\sum_{jet} p_T}, \ln \frac{E}{\sum_{jet} E}, \Delta R$ $(E, p_x, p_y, p_z)$

# Tagger Performance

Calculated using samples with steeply falling pT spectra, i.e. both sig & bkg are weighted to have falling pT spectra.

For a signal efficiency of 0.5 (0.8) case, the background rejection of ParticleTransformer is about 1.8-2.8 (1.6-2.7) times better than the baseline tagger.



# Tagger Performance

Model	AUC	ACC	$\varepsilon_{bkg}^{-1}$ @ $\varepsilon_{sig} = 0.5$	$\varepsilon_{bkg}^{-1}$ @ $\varepsilon_{sig} = 0.8$	# Params	Inference Time
EFN	0.920	0.835	35.1	7.95	56.73k	0.065 ms
PFN	0.931	0.853	44.7	9.50	57.13k	0.11 ms
ParticleNet	0.933	0.826	46.2	9.76	366.16k	0.36 ms
ParticleTransformer	0.951	0.880	77.9	14.6	2.14M	0.28 ms

Table 3: The performance of each  $W$  jet tagger is measured with several metrics evaluated on the testing set.

Transformers the best

But the # of parameters is almost one order of magnitude larger

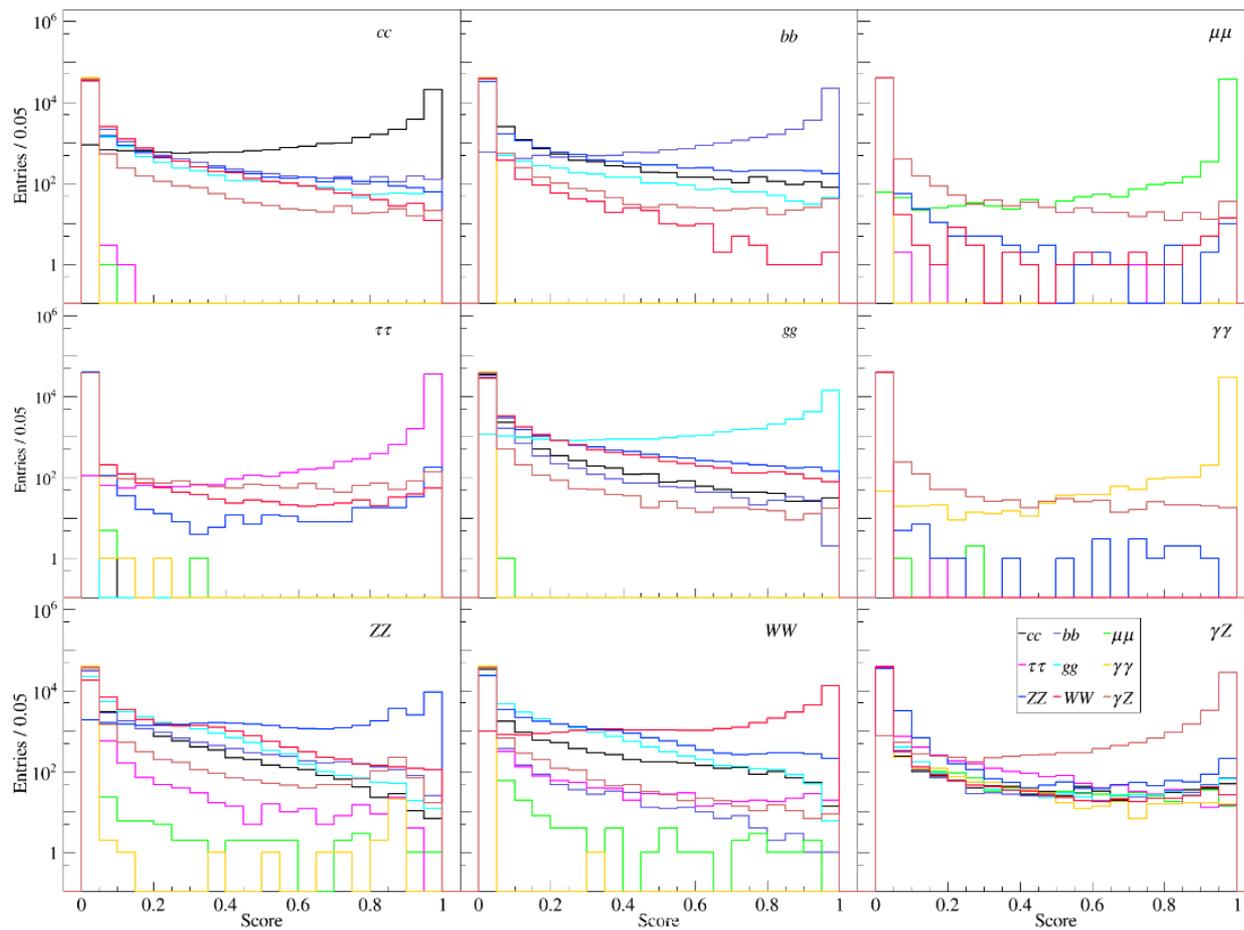
# Many processes are selected simultaneously

Prod/decay	cc	bb	mm	$\tau\tau$	gg	gg	WW	ZZ	aZ	ee, uu, dd, ss
eeH	3	1	5	2	4	1	2	3	5	Not covered yet
mmH	3	1	5	2	4	1	2	3	5	
$\tau\tau$ H	3	1	5	2	4	1	2	3	5	
qqH	4	1	2	1	2	5	5	5	3	
nnH	5	1	3	2	3	5	4	2	4	

Consider:  $\psi(2S) \rightarrow \pi^+ \pi^- J/\psi$ ,  $J/\psi \rightarrow$  various processes

Try eeH first

Probability distributions of each class

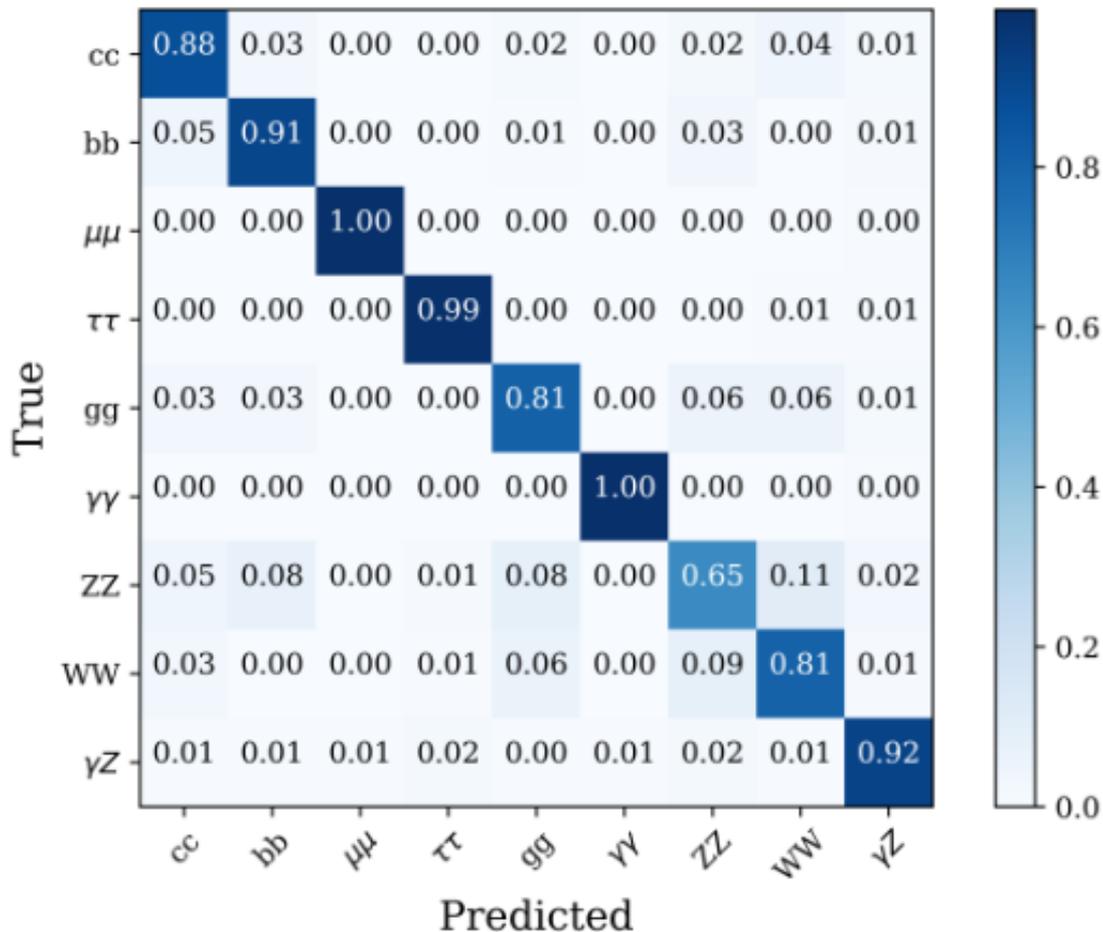


Try eeH first

Sufficiently good performance

Average Accuracy ~ 87%

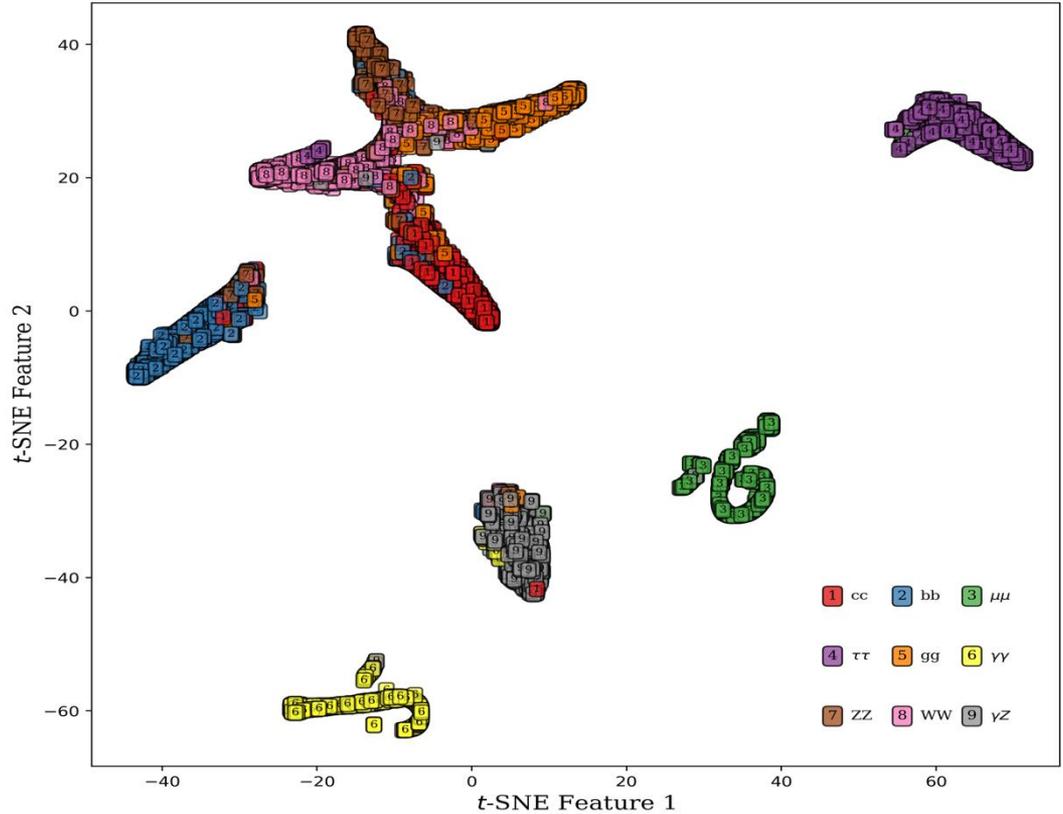
(11% for random guess)



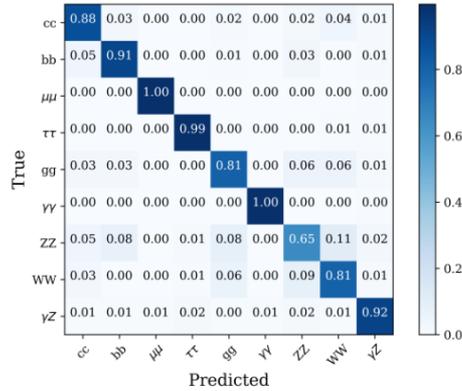
Taking the one has largest probability (ArgMax)

## Dimension reduction

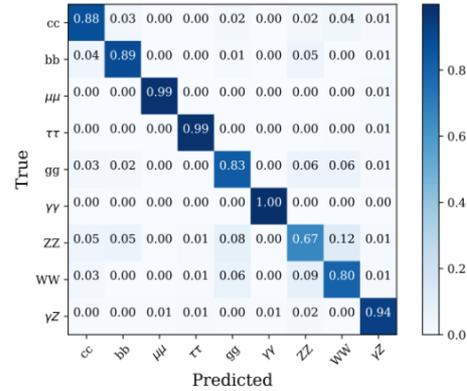
- ✓  $\mu\mu$ ,  $\gamma\gamma$ ,  $\tau\tau$  well classified as expected
- ✓  $bb$  and  $\gamma Z$  also good
- ✓  $cc$ ,  $gg$ ,  $WW$ , and  $ZZ$  fake each other, but under control



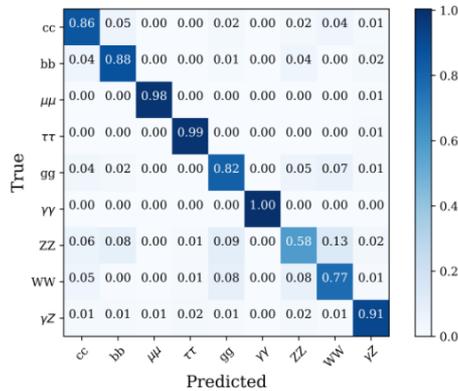
# All 4 production modes



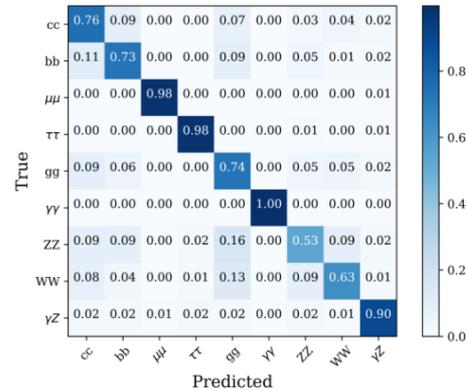
$eeH$



$\mu\mu H$



$\tau\tau H$



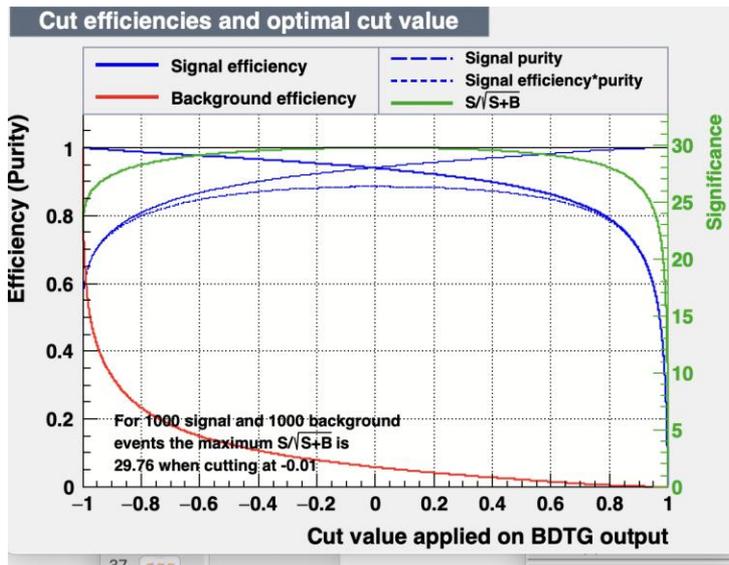
$qqH$





# 操作实例 1 : ROOT TMVA 分类

- 观察输出信息
- 理解数据的预处理
  - 归一化
  - 权重
  - 相关性检查
  - 重要性评价
- 训练过程
- 结果评估
  - 是否过渡训练, 如何防止
- 结果诠释和使用
  - ROC
- ... ..



TMVA Plotting Macros for Classification	
(1a)	Input variables (training sample)
(1b)	Input variables 'Deco'-transformed (training sample)
(1c)	Input variables 'PCA'-transformed (training sample)
(1d)	Input variables 'Gauss_Deco'-transformed (training sample)
(2a)	Input variable correlations (scatter profiles)
(2b)	Input variable correlations 'Deco'-transformed (scatter profiles)
(2c)	Input variable correlations 'PCA'-transformed (scatter profiles)
(2d)	Input variable correlations 'Gauss_Deco'-transformed (scatter profiles)
(3)	Input Variable Linear Correlation Coefficients
(4a)	Classifier Output Distributions (test sample)
(4b)	Classifier Output Distributions (test and training samples superimposed)
(4c)	Classifier Probability Distributions (test sample)
(4d)	Classifier Rarity Distributions (test sample)
(5a)	Classifier Cut Efficiencies
(5b)	Classifier Background Rejection vs Signal Efficiency (ROC curve)
(5b)	Classifier 1/(Backgr. Efficiency) vs Signal Efficiency (ROC curve)
(6)	Parallel Coordinates (requires ROOT-version $\geq 5.17$ )
(7)	PDFs of Classifiers (requires "CreateMVAPdfs" option set)
(8)	Training History
(9)	Likelihood Reference Distributions
(10a)	Network Architecture (MLP)
(10b)	Network Convergence Test (MLP)
(11)	Decision Trees (BDT)
(12)	Decision Tree Control Plots (BDT)
(13)	Plot Foams (PDEFoam)
(14)	General Boost Control Plots
(15)	Quit

# 操作实例2 : 一个用GNN做三分类的python例子

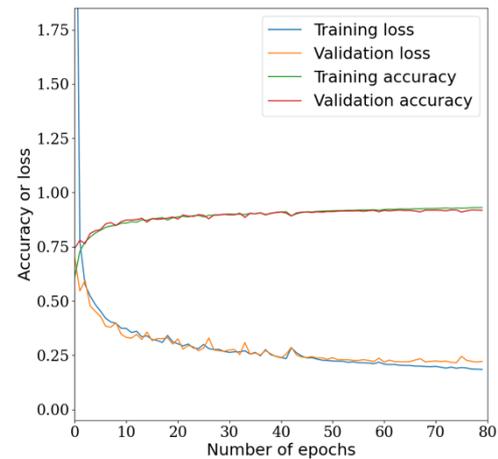
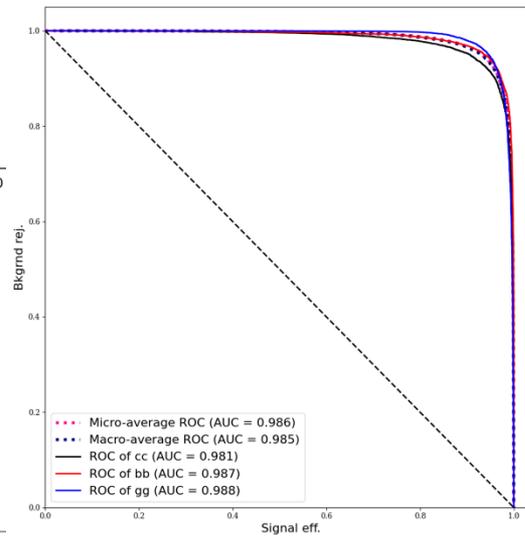
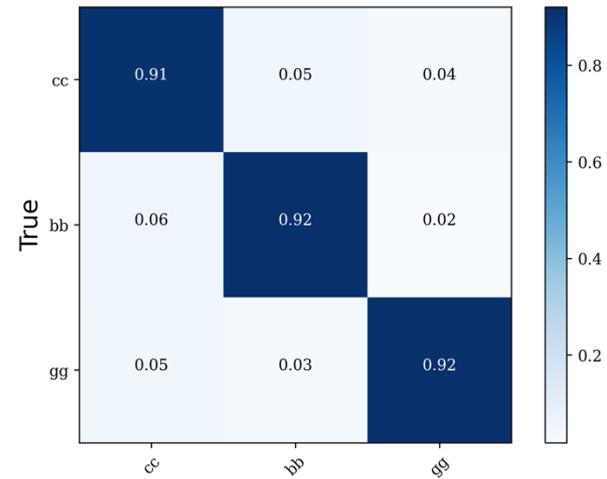
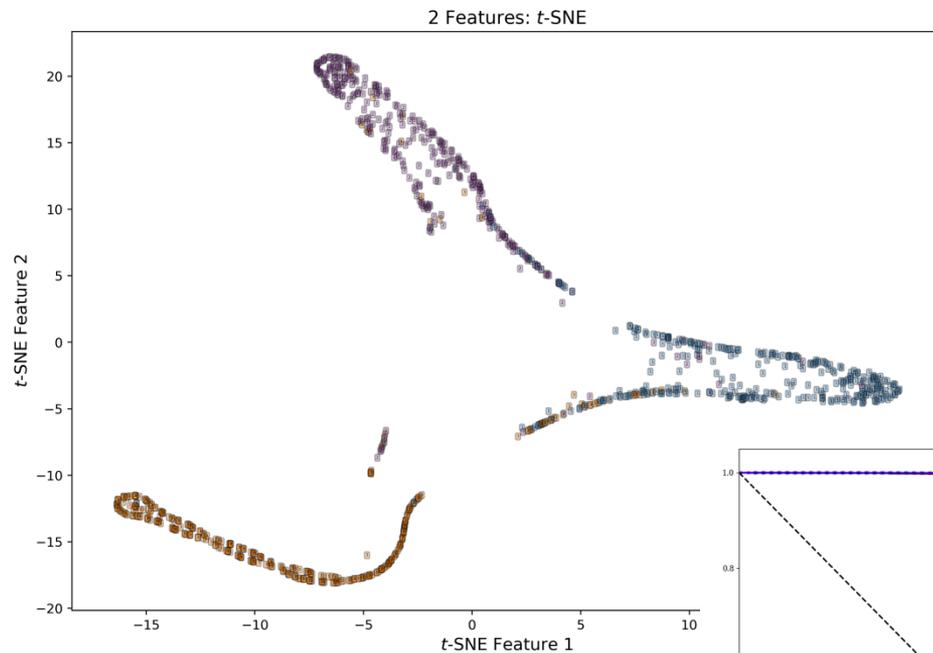
建立环境: miniconda

执行作业: 训练曲线

分析结果: ROC, confusion matrix

展示结果: 降维

基于 DeepSet 的一个简化例子  
JHEP 01 (2019) 121 [1810.05165].



# 小结

- 机器学习本质上是统计学习：NFL一直存在
- 机器学习处理的往往是高维问题，必然存在 CoD 问题，而 NN 被证实有应对 CoD 的能力。
- 我们在实际研究工作中应该选择具有恰当 inductive bias 的算法，这样在性能和解释性方面都非常有利。
- 机器学习在高能物理实验各个环节上会有光明前景
- 除了机器学习本身的知识外，请大家多关注 高维统计、信息论等更为基础的数学。我们不是单纯的用户。
- 大模型更为强大，各中可能使用仍在尝试当中：DrSai, ChatLas, ...