Machine learning in heavy ion collisions

LongGang Pang

UC Berkeley & Lawrence Berkeley National Laboratory

Jan 13, 2019 JETSCAPE winterschool and workshop @ Texas A&M

Outline

- What’s machine learning
- Applications of machine learning in HIC
    - Supervised learning
    - Unsupervised learning
- Challenges for heavy ion jets

What’s machine learning

Artificial intelligence

Machine Learning

PCA
SVM
Bayesian analysis
Decision Tree
Random Forest
Gradient Boosting Tree
Neural networks
...

Deep Learning

Mainly deep neural network

AlphaGo, AlphaGo Zero, Alpha Zero, Google translate, Amazon Echo, Self driving cars ...

Applications of machine learning in HIC

- **Supervised learning**
  - The most popular method in HIC: Bayesian analysis
  - Neural network for impact parameter determination
  - Convolution neural network for nuclear equation of state and nuclear structure.
  - UNET for fast relativistic hydrodynamics
- **Unsupervised learning**
  - Principle component analysis (PCA) for flow harmonics

Bayesian analysis in heavy ion collisions

What's scientific method? by Feymann

In general, we look for a new law by the following process. First, we guess it, no, don't laugh, that's the truth. Then we compute the consequences of the guess, to see what, if this is right, if this law we guess is right, to see what it would imply and then we compare the computation results to nature or we say compare to experiment or experience, compare it directly with observations to see if it works.

Bayesian analysis is the scientific method!

$$P(\theta|\mathrm{data}) = \frac{P(\theta)P(\mathrm{data}|\theta)}{P(\mathrm{data})}$$

where $P(\theta|\mathrm{data})$ is the posterior distribution of parameters θ given the experimental data, $P(\theta)$ is the prior (guess) distribution of θ, $P(\mathrm{data}|\theta)$ is the Gaussian likelihood between experimental data and model output for any given θ, $P(\mathrm{data}) = \int d\theta P(\theta)P(data|\theta)$ is the evidence.

Bayesian analysis in heavy ion collisions

The evidence $P(\text{data}) = \int d\theta P(\theta)P(data|\theta)$ is very expensive to compute. One can use Markov Chain Monte Carlo (MCMC) method to generate θ, whose distribution mimics the unnormalized probability distribution,

$$P(\theta|\text{data}) \propto P(\theta)P(\text{data}|\theta)$$

with Metropolis Hastings algorithm (importance sampling).

- Initialize θ^0.
- For i=0 to N-1
  - Sample $r \sim U[0,1]$.
  - Sample $\theta^* \sim q(\theta^*|\theta^i)$
  - If $r < \min\left(1, \frac{p(\theta^*)q(\theta^i|\theta^*)}{p(\theta^i)q(\theta^*|\theta^i)}\right)$

    $$\theta^{i+1} = \theta^*$$

    else

    $$\theta^{i+1} = \theta^i$$

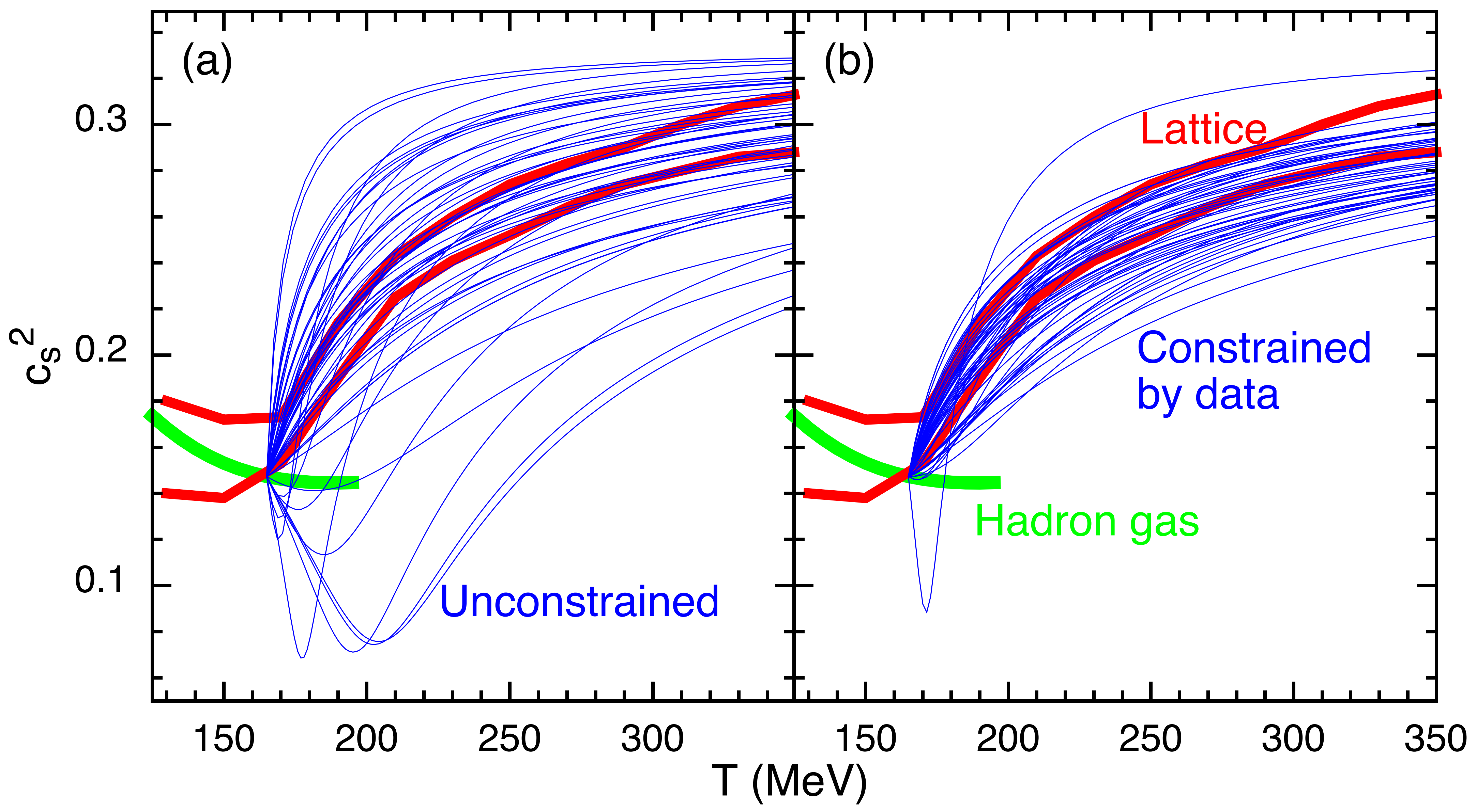
Bayesian analysis for EoS in heavy ion collisions

Parameterized QGP EoS with 2 parameters

PRL**114**, 202301 Scott Pratt, Evan Sangaline, Paul Sorensen, and Hui Wang

$$c_s^2(\epsilon) = c_s^2(\epsilon_h) + \left(\frac{1}{3} - c_s^2(\epsilon_h)\right)\frac{X_0 x + x^2}{X_0 x + x^2 + X'^2},$$

$$X_0 = X' R c_s(\epsilon)\sqrt{12}, \qquad x \equiv \ln \epsilon/\epsilon_h,$$



where ϵ_h is the energy density corresponds

to temperature T=165 MeV;

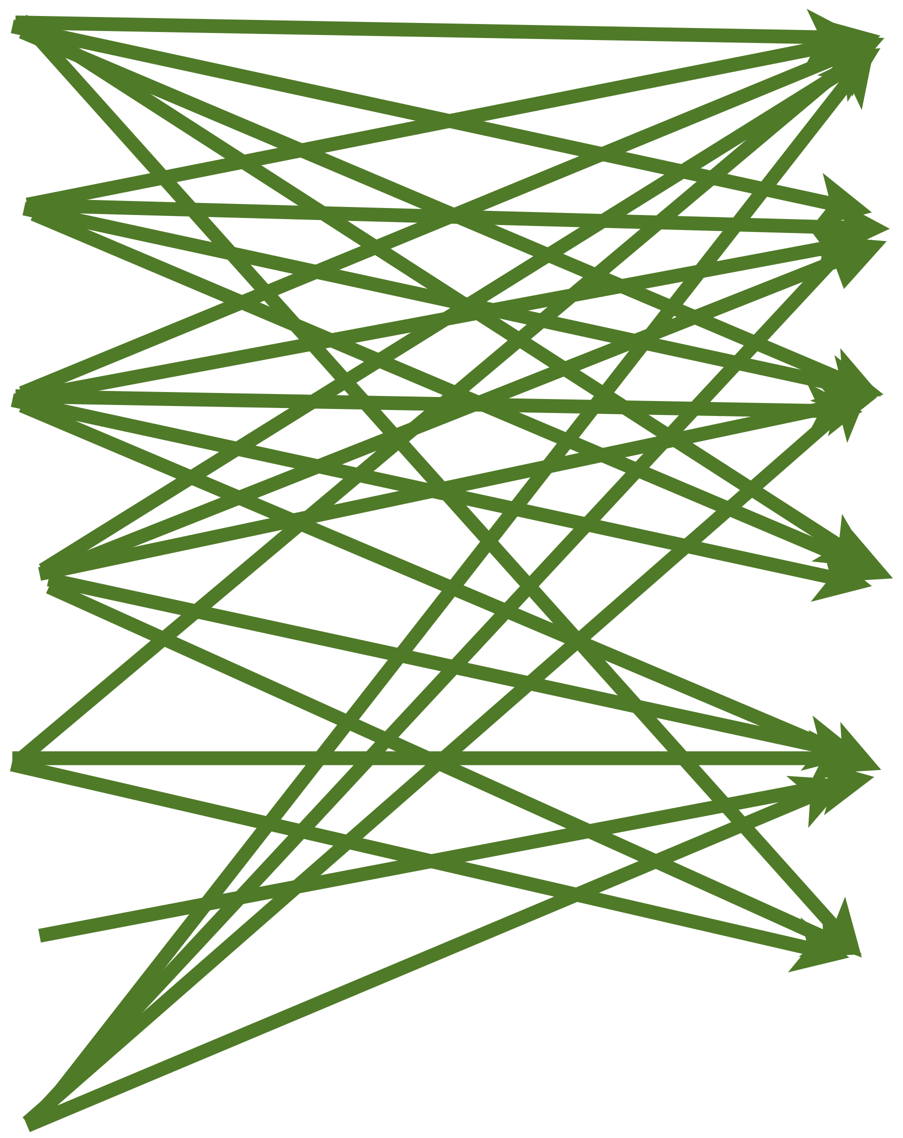
R and X' controls the shape of the speed of sound as a function of energy density.

Multiple parameters entangle with multiple observables

From S.Bass QM2017 talk.

Model Parameter:

eqn. of state
shear viscosity
initial state
pre-equilibrium dynamics
thermalization time
quark/hadron chemistry
particlization/freeze-out



experimental data:

π/K/P spectra
yields vs. centrality & beam
elliptic flow
HBT
charge correlations & BFs
density correlations

Bayesian analysis for global fitting in heavy ion collisions

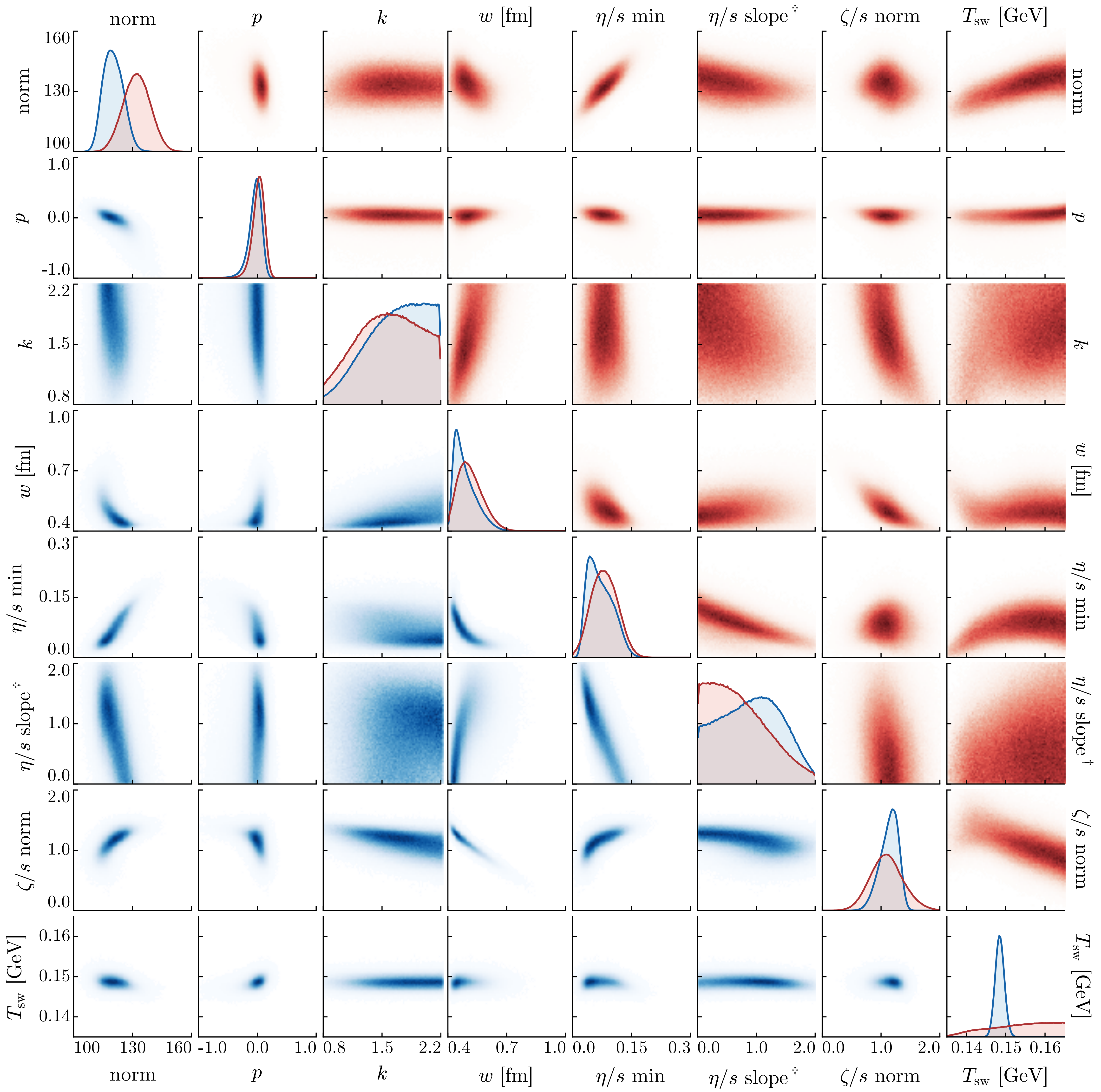


FIG. 7. Posterior distributions for the model parameters from calibrating to identified particles yields (blue, lower triangle) and charged particles yields (red, upper triangle). The diagonal has marginal distributions for each parameter, while the off-diagonal contains joint distributions showing correlations among pairs of parameters. †The units for η/s slope are [GeV^{-1}].

PRC 94.024907

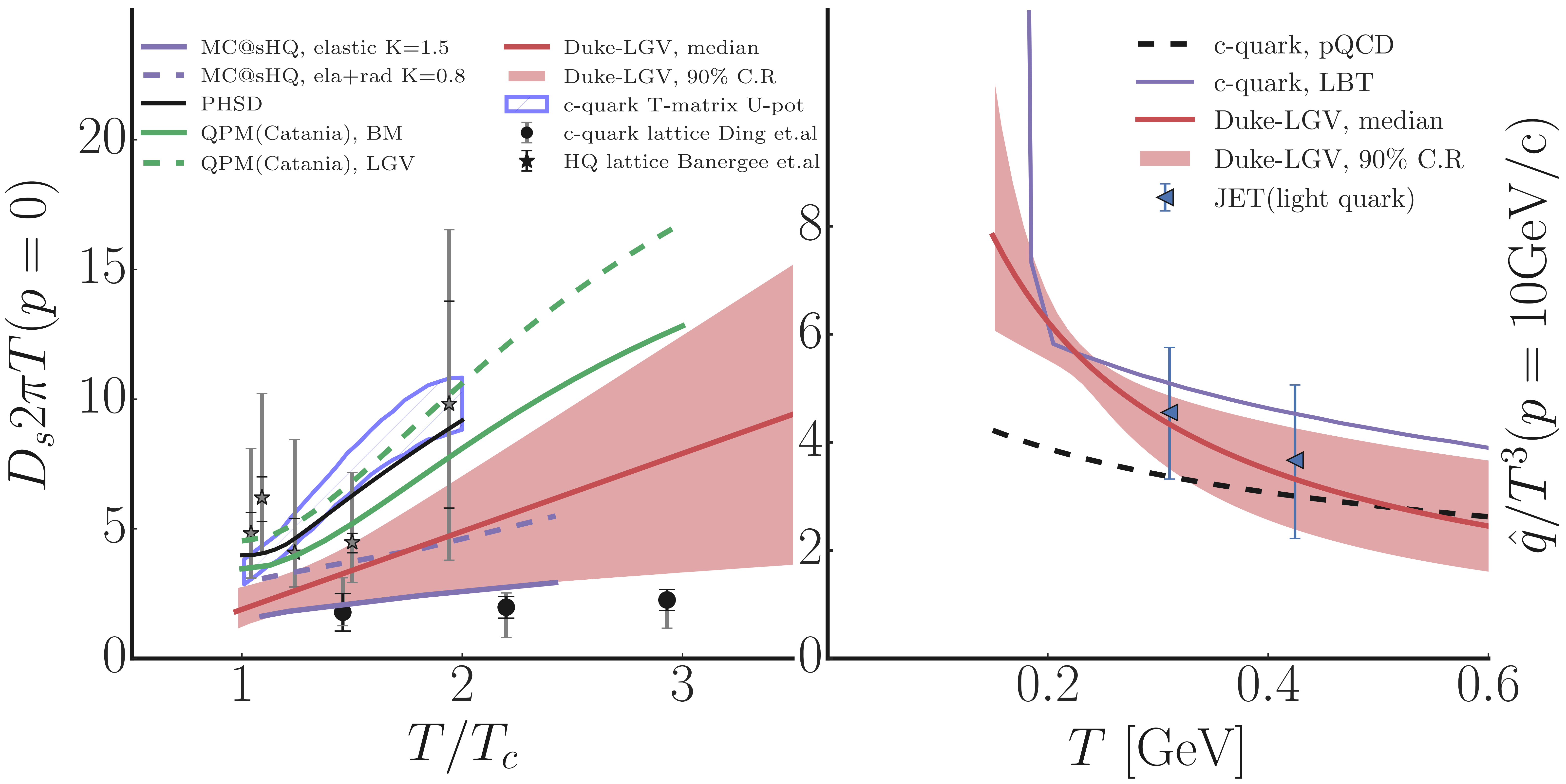
Jonah E. Bernhard, J. Scott Moreland, and Steffen A. Bass
Department of Physics, Duke University, Durham, NC 27708-0305

Jia Liu and Ulrich Heinz
Department of Physics, The Ohio State University, Columbus, OH 43210-1117

TABLE I. Input parameter ranges for the initial condition and hydrodynamic models.

Parameter	Description	Range
Norm	Overall normalization	100–250
p	Entropy deposition parameter	-1 to $+1$
k	Multiplicity fluct. shape	0.8–2.2
w	Gaussian nucleon width	0.4–1.0 fm
η/s hrg	Const. shear viscosity, $T < T_c$	0.3–1.0
η/s min	Shear viscosity at T_c	0–0.3
η/s slope	Slope above T_c	0–2 GeV^{-1}
ζ/s norm	Prefactor for $(\zeta/s)(T)$	0–2
T_{switch}	Particlization temperature	135–165 MeV

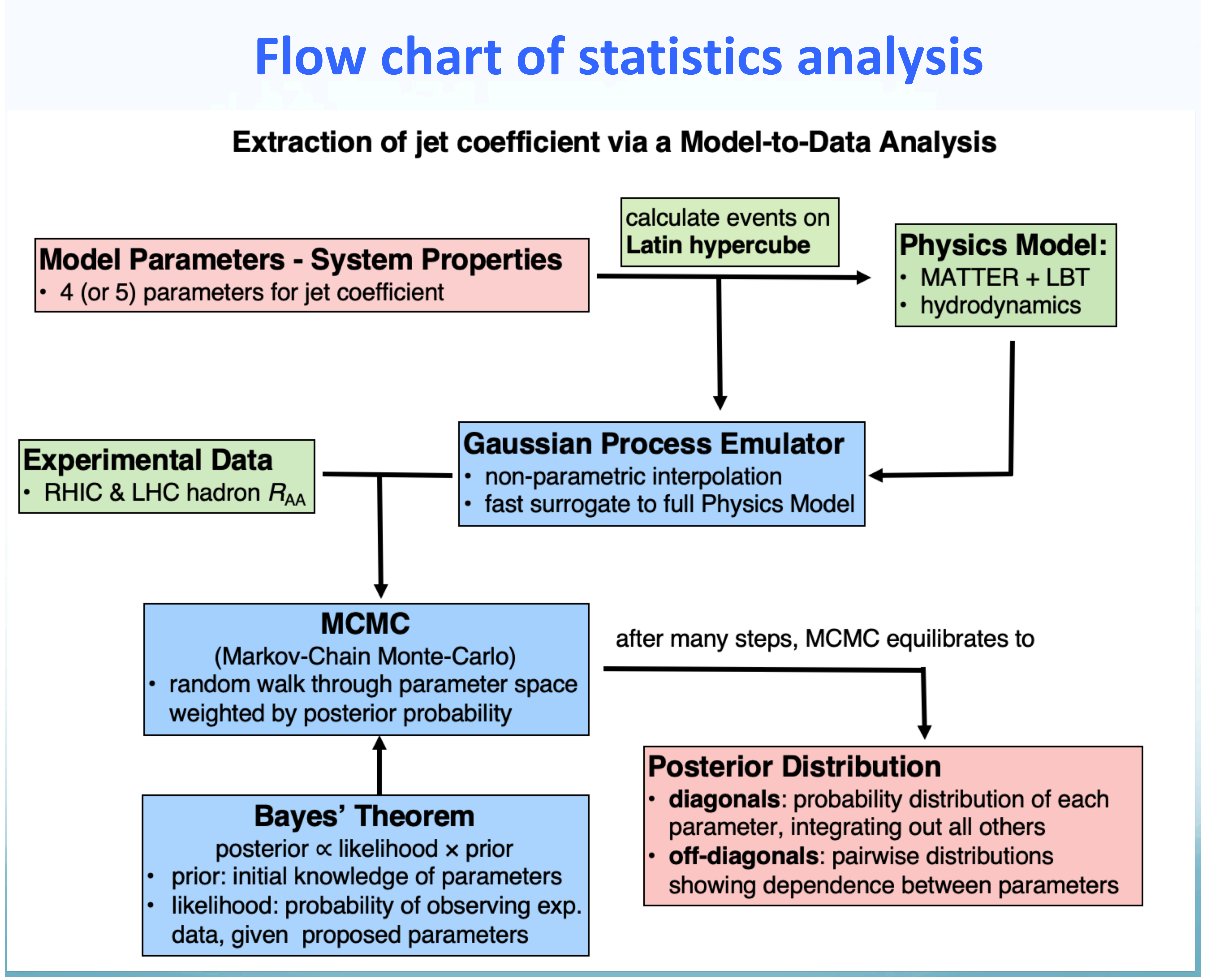
Bayesian analysis for heavy-quark diffusion coefficient in heavy ion collisions

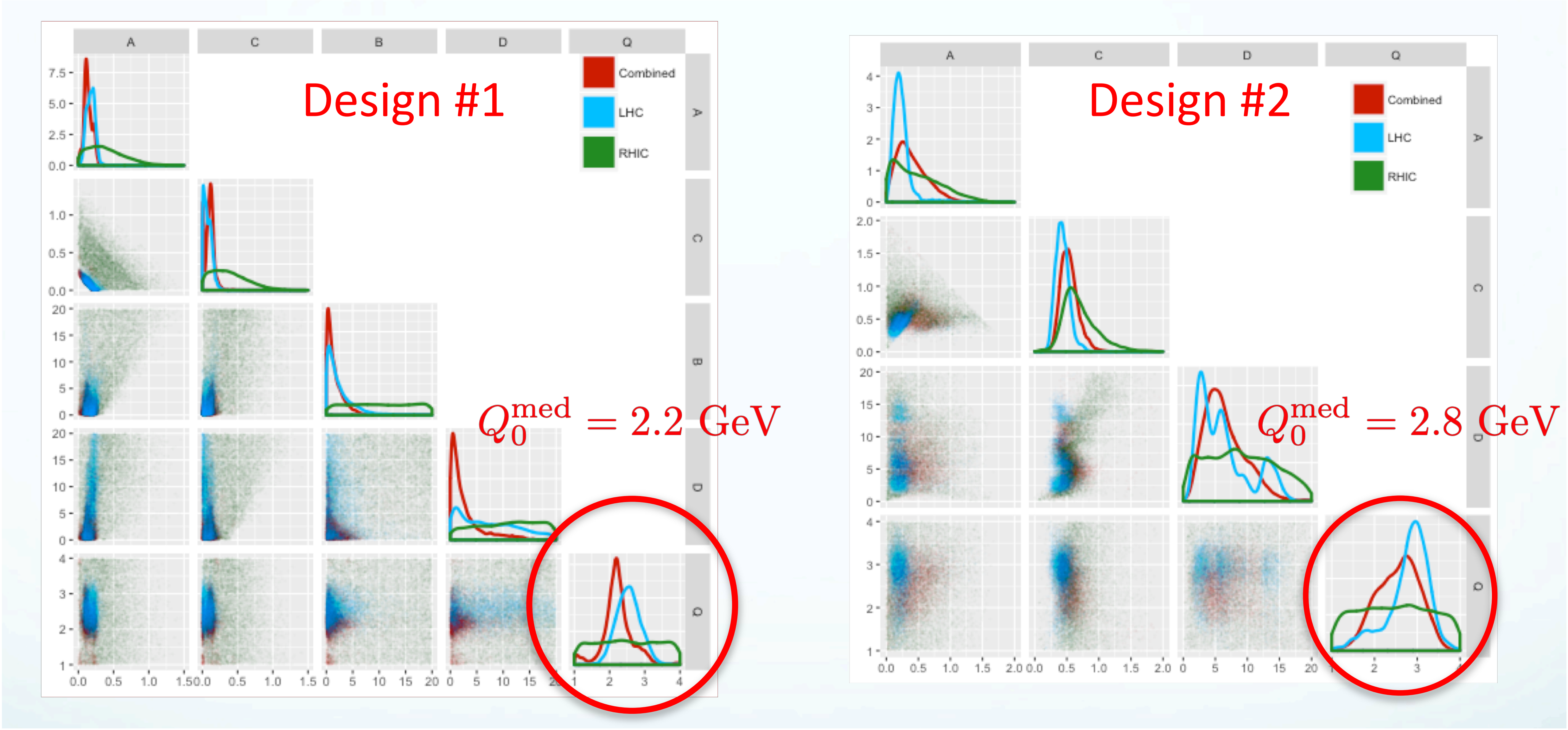


(Color online) Comparison of the heavy quark diffusion coefficients across multiple approaches available in the literature. (**left**) spatial diffusion coefficient at zero momentum $D_s 2\pi T(p=0)$. (**right**) momentum diffusion coefficient $\hat{q}/T^3$ at $p = 10$ GeV.

PRC. **97 (2018)**, 014907, Yingru Xu, et.el

Bayesian extraction of jet transport coefficient using Jetscape





- First quantitative constraint on Q0.
- Better constrained using more data

From ShanShan Cao's talk, in present of JETSCAPE

Bayesian extraction of jet energy loss distributions in heavy-ion collisions

$$R_{AA}(p_T) \approx \frac{\int d\Delta p_T d\sigma_{pp}^{\text{jet}}(p_T + \Delta p_T) W_{AA}(p_T + \Delta p_T \to p_T, R)}{d\sigma_{pp}^{\text{jet}}(p_T)}.$$

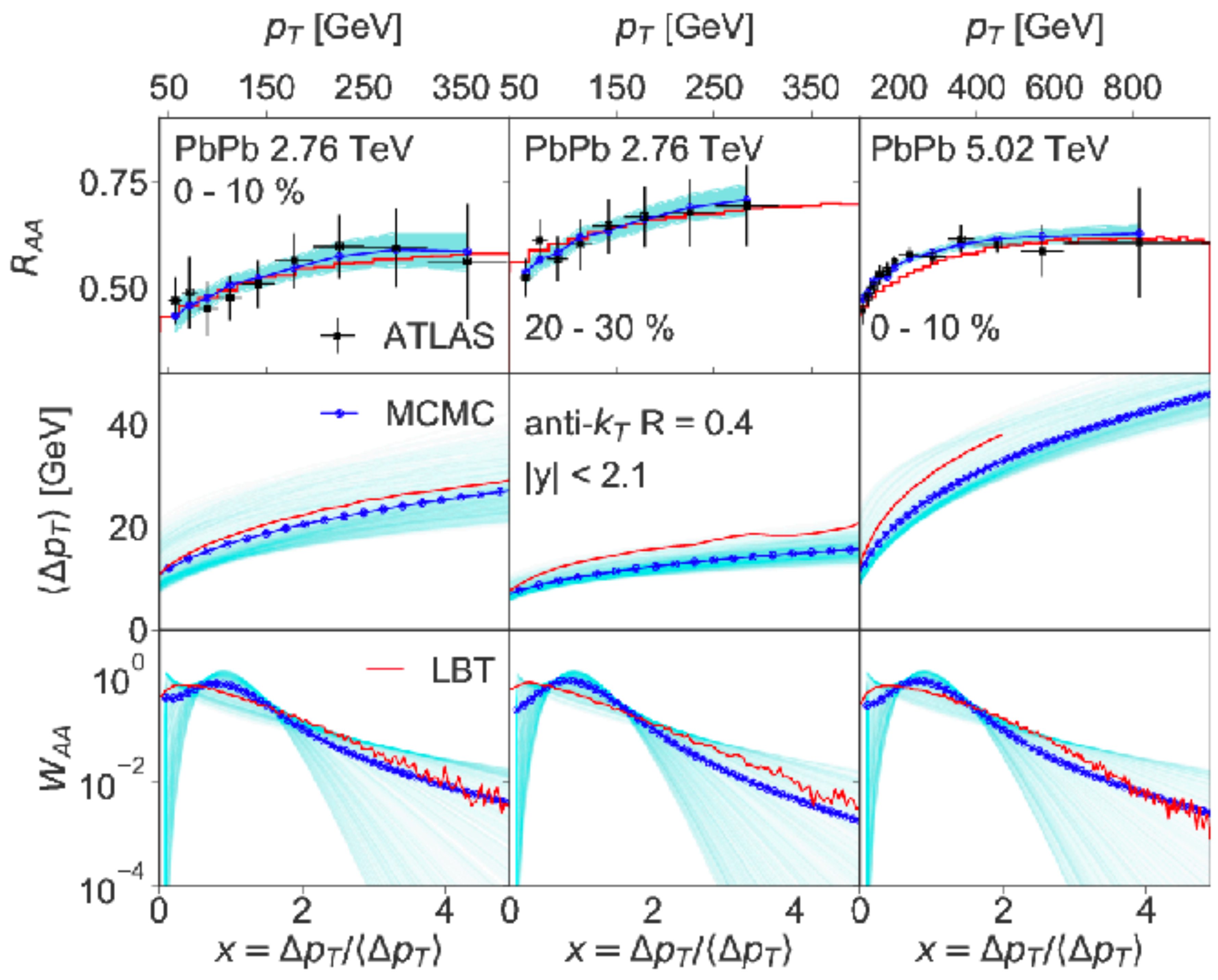
from where one can define the mean p_T loss,

$$\langle \Delta p_T \rangle (p_T) = \int \mathrm{d}\Delta p_T \Delta p_T W_{AA}(p_T \to p_T - \Delta p_T, R)$$

LBT inspired statistical model:

$$W_{AA}(x) = \frac{\alpha^\alpha x^{\alpha-1} e^{-\alpha x}}{\Gamma(\alpha)} \quad \text{where } x = \frac{\Delta p_T}{\langle \Delta p_T \rangle}$$

$$\langle \Delta p_T \rangle (p_T) = \beta p_T^\gamma \log(p_T)$$



arXiv:1808.05310, with YaYun He and Xin-Nian Wang

What is artificial neural network

Forward pass

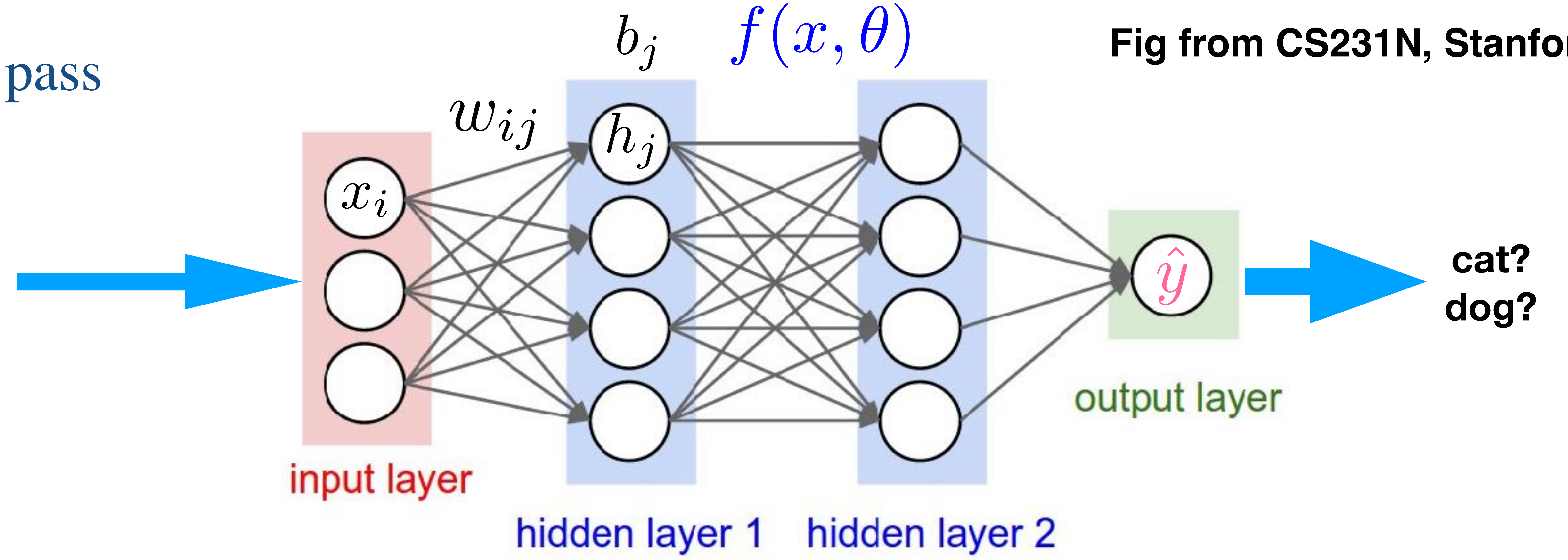


Fig from CS231N, Stanford

Linear operation

$$z_j = \sum_{i=1}^{N} x_i w_{ij} + b_j$$

scaling, rotating, boosting,

changing dimensions

Non-linear activation function $h_j = \sigma(z_j)$

(a) Sigmoid

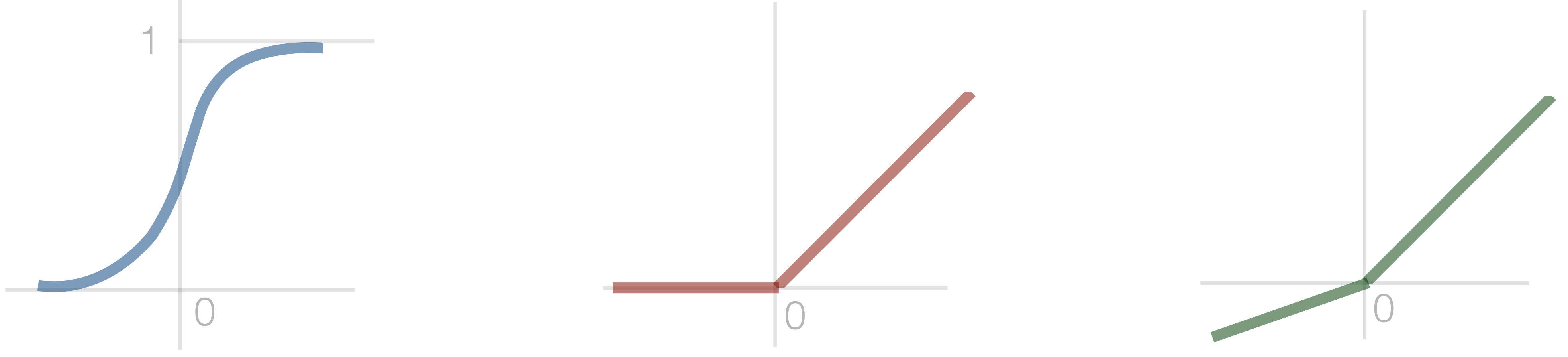
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

(b) ReLU

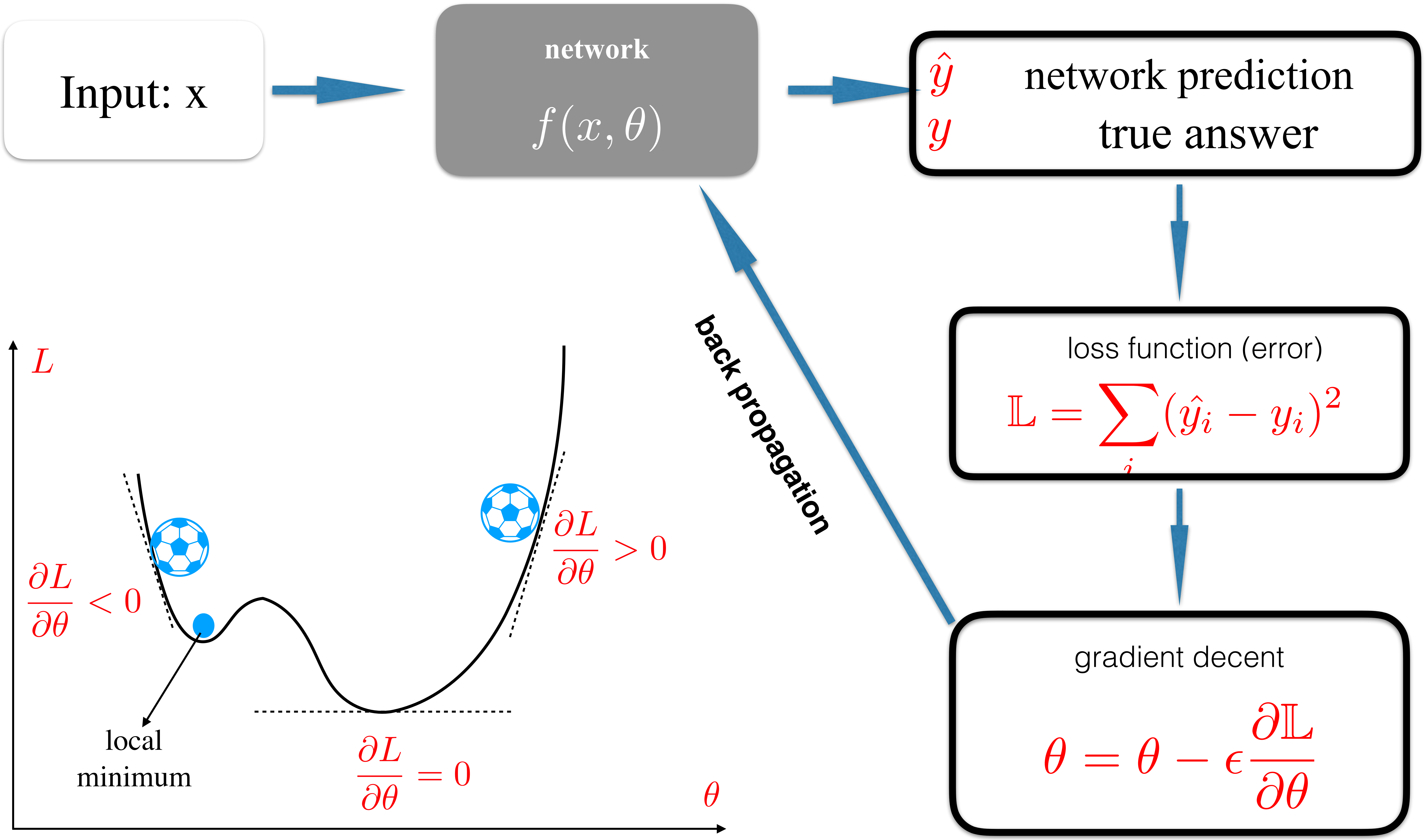
$$\sigma(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases}$$

(c) PReLU

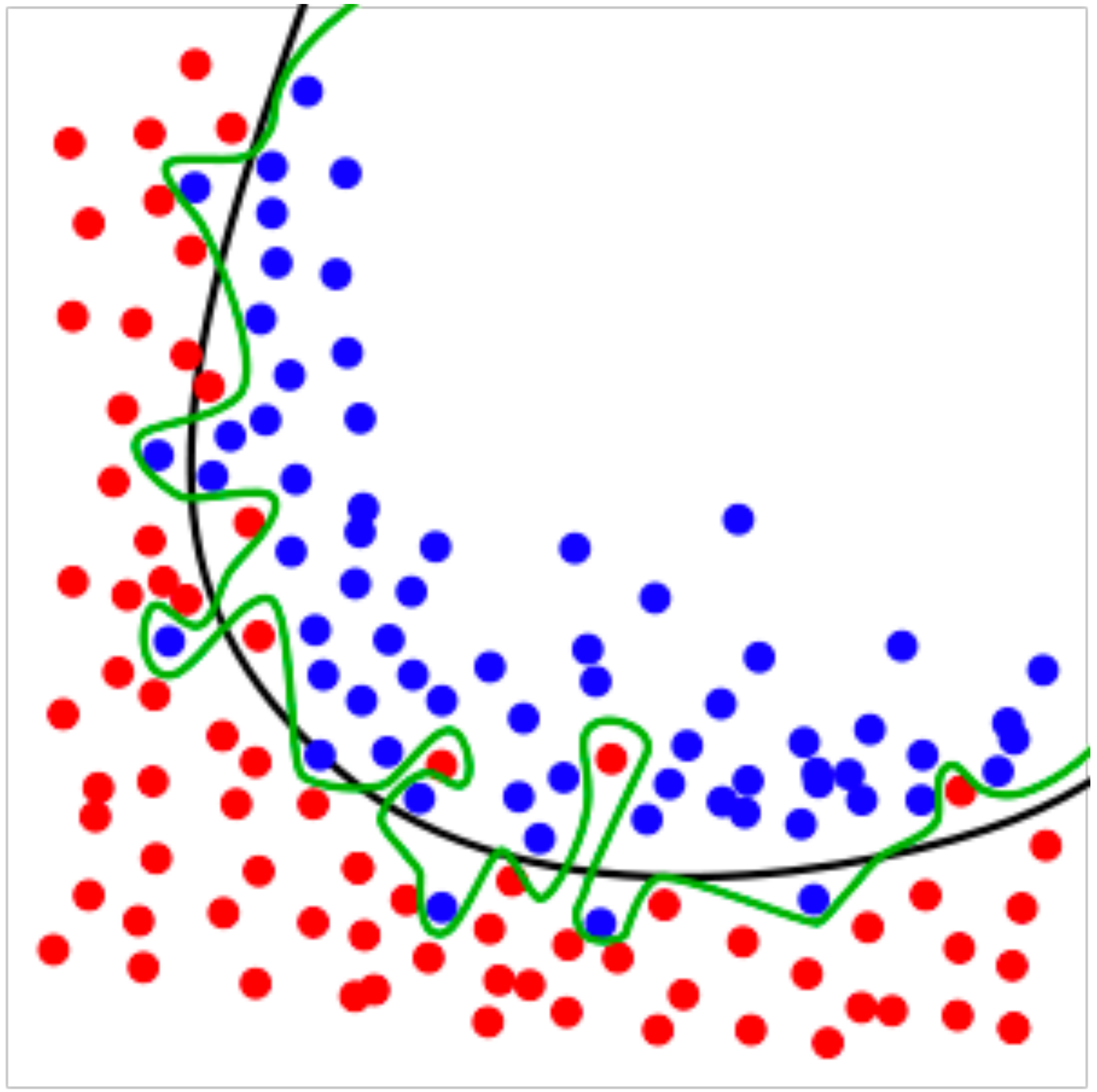
$$\sigma(z) = \begin{cases} z, & z > 0 \\ az, & z \leq 0 \end{cases}$$

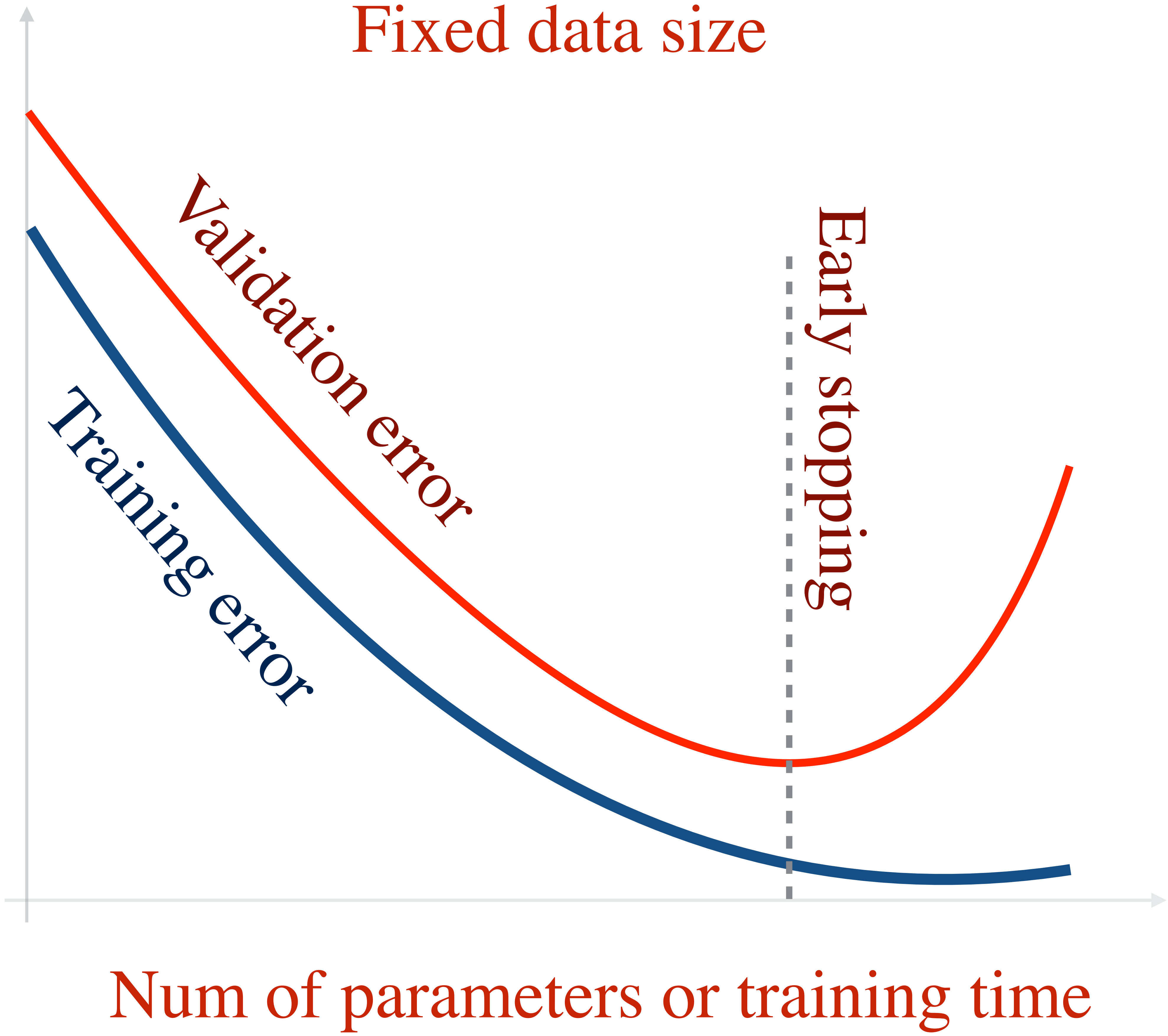


How does neural network learn



Overfitting problem in fully connected network





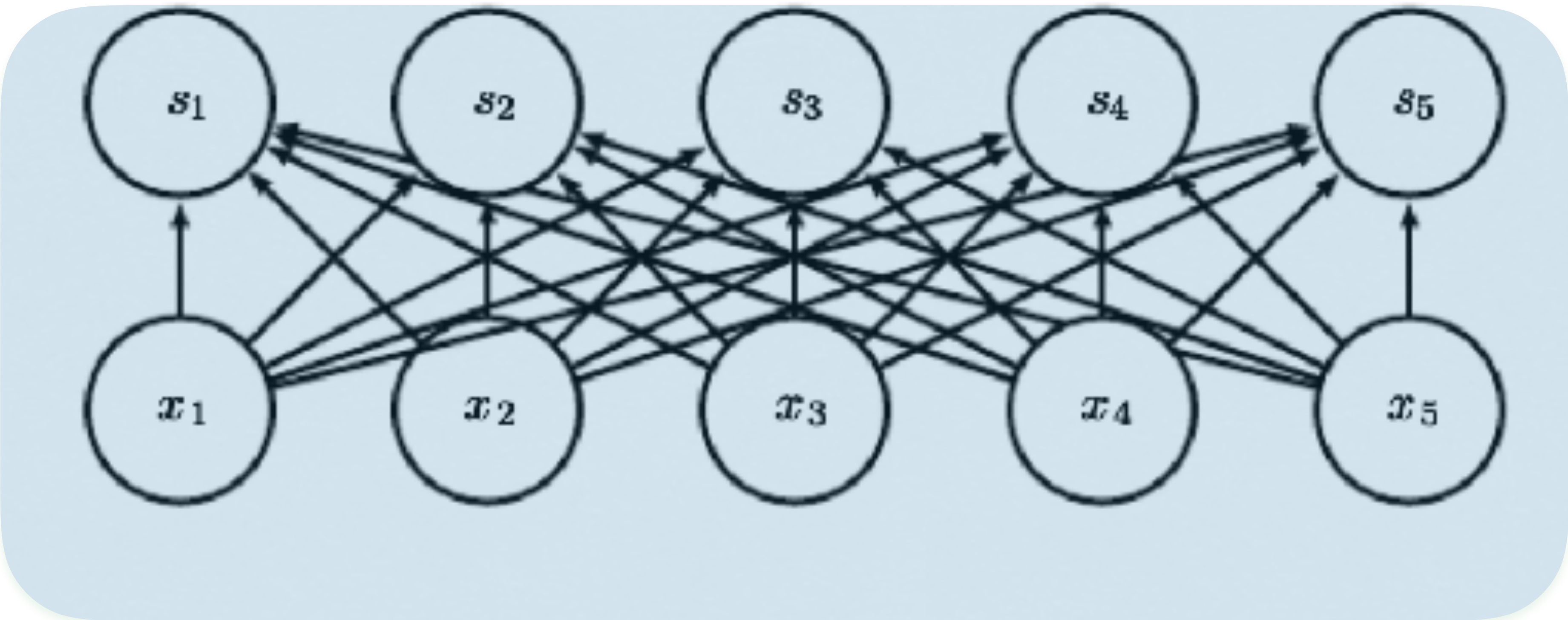
Goal: network trained with some data (training data) should generalize well on new data (validation data).

Ways to reduce overfitting

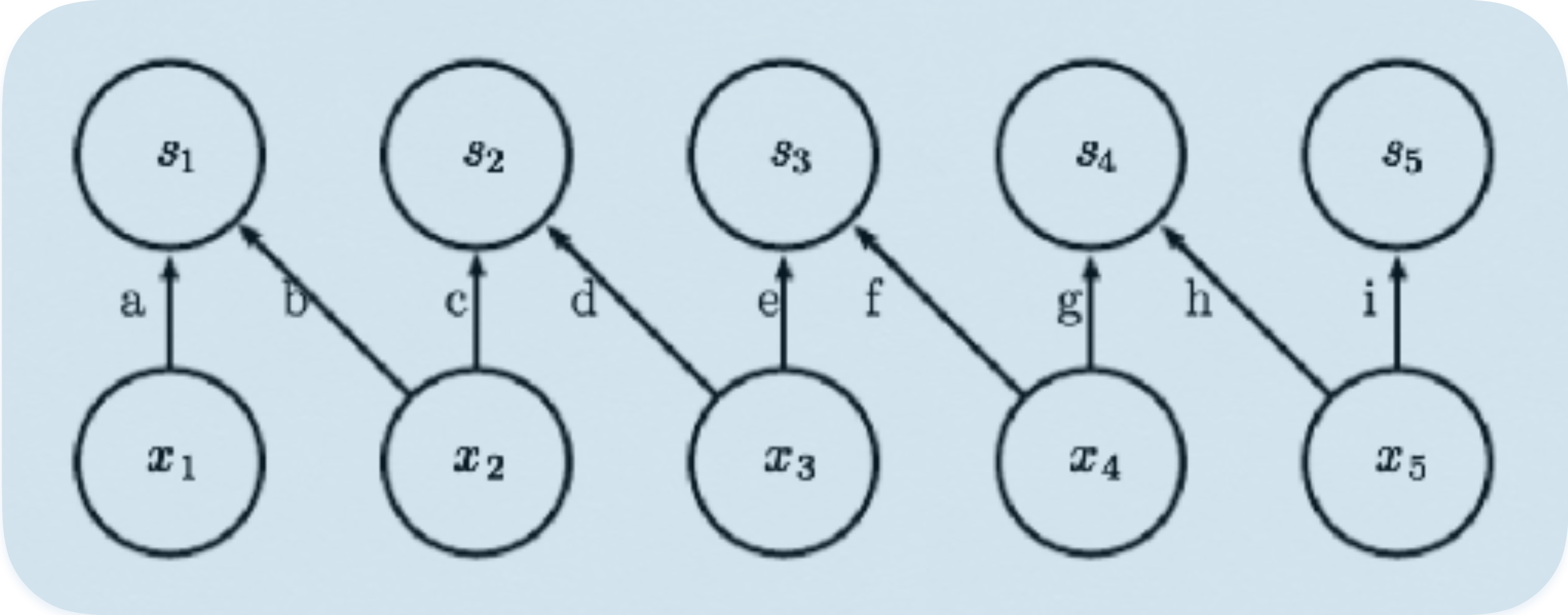
1. Early stopping
2. Increase training dataset by
   a. preparing big amount of data.
   b. data augmentation (crop, scale, rotate, flip …).
3. Reduce number of parameters
   a. Dropout: randomly discard neurons.
   b. Drop connection: randomly discard connections.
   c. CNN: locally connected to a small chunk of neurons in the previous layer.
   d. Go deep. S.Liang & R.Srikant, arXiv:1610.04161,
4. Regularization, weight decay …

Convolution neural network — 1D

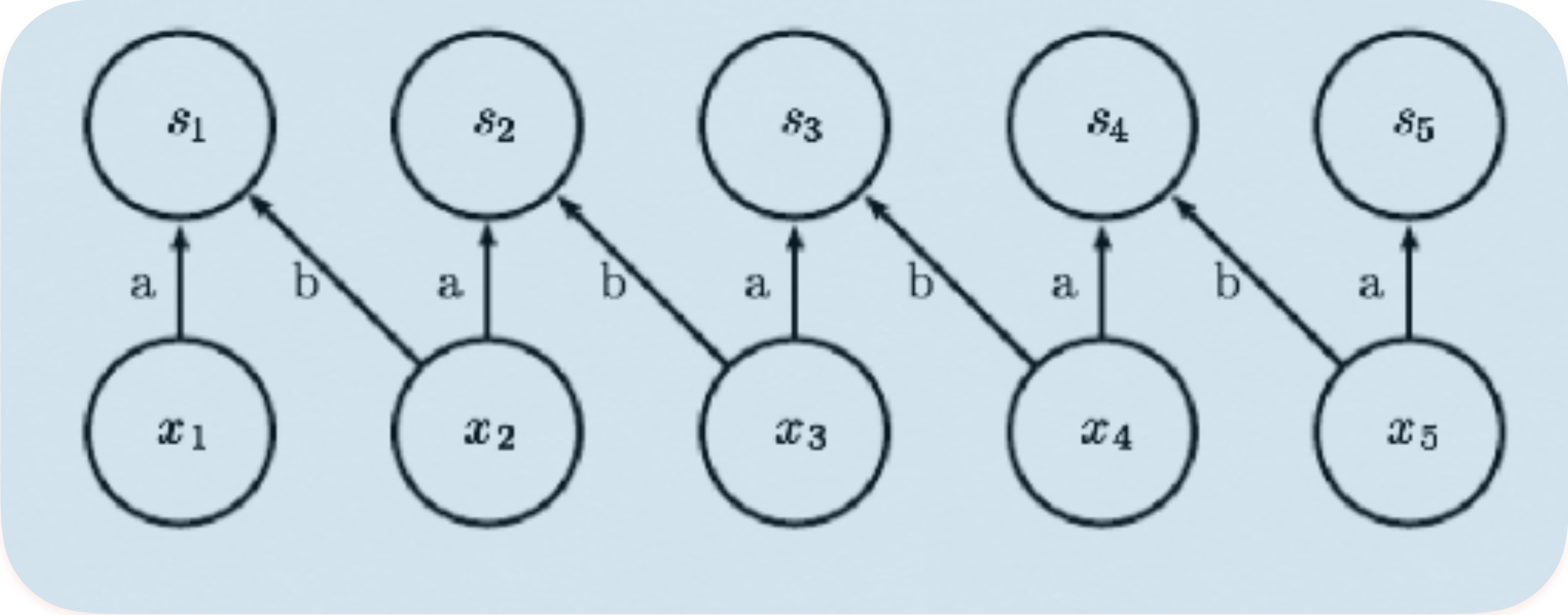
Fully Connected



Locally Connected



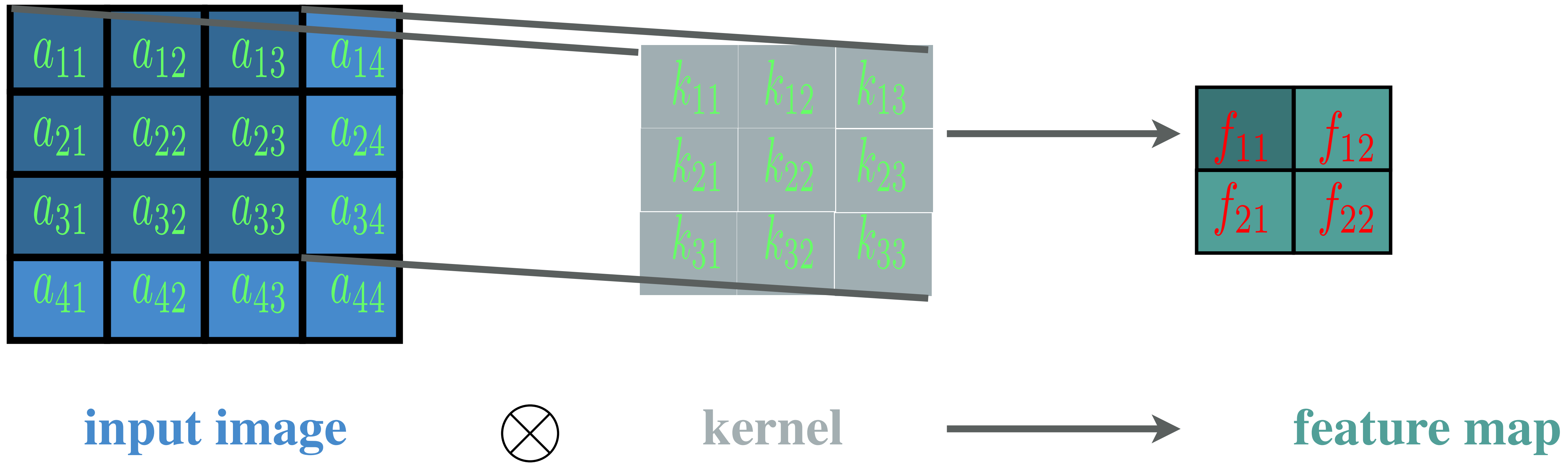
Locally Connected
+
Share Weights



Convolution

From "Deep Learning" Book.

Convolution Neural Network — 2D

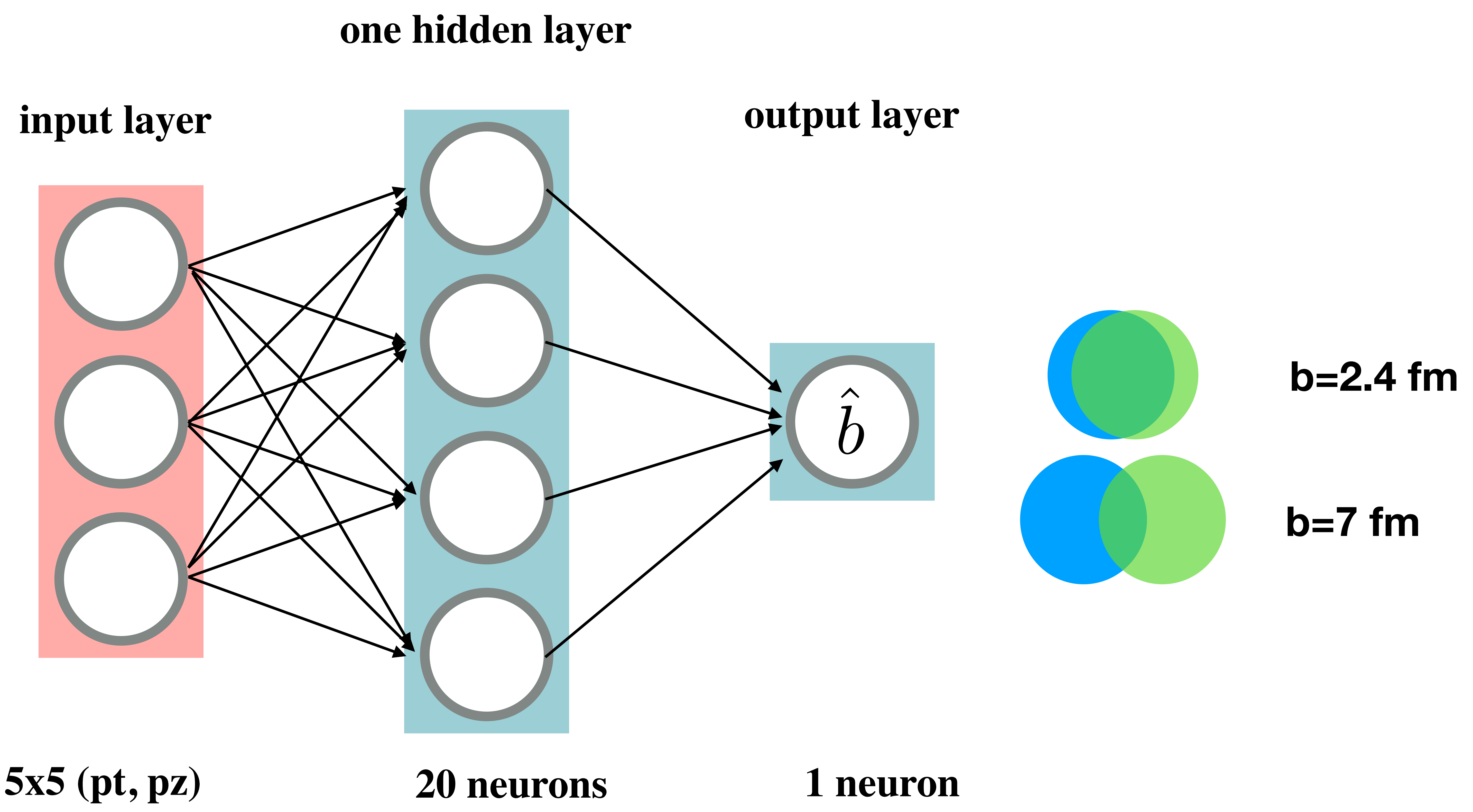


$$
\begin{aligned}
f_{11} = {} & a_{11}k_{11} + a_{12}k_{12} + a_{13}k_{13} \\
& a_{21}k_{21} + a_{22}k_{22} + a_{23}k_{23} \\
& a_{31}k_{31} + a_{32}k_{32} + a_{33}k_{33}
\end{aligned}
$$

Neural Network for Impact Parameter Determination

PRC53:2358-2363,1996; S.A.Bass et.al

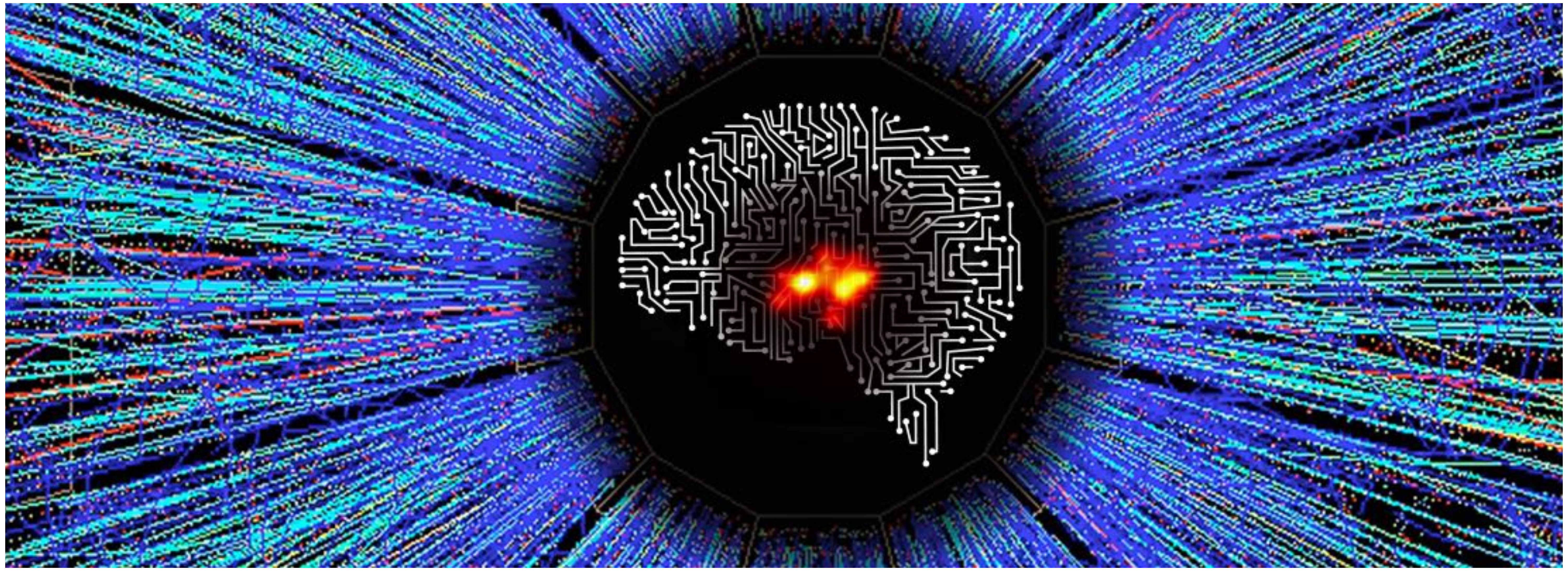
An improvement in performance of a factor of two as compared to classical techniques.

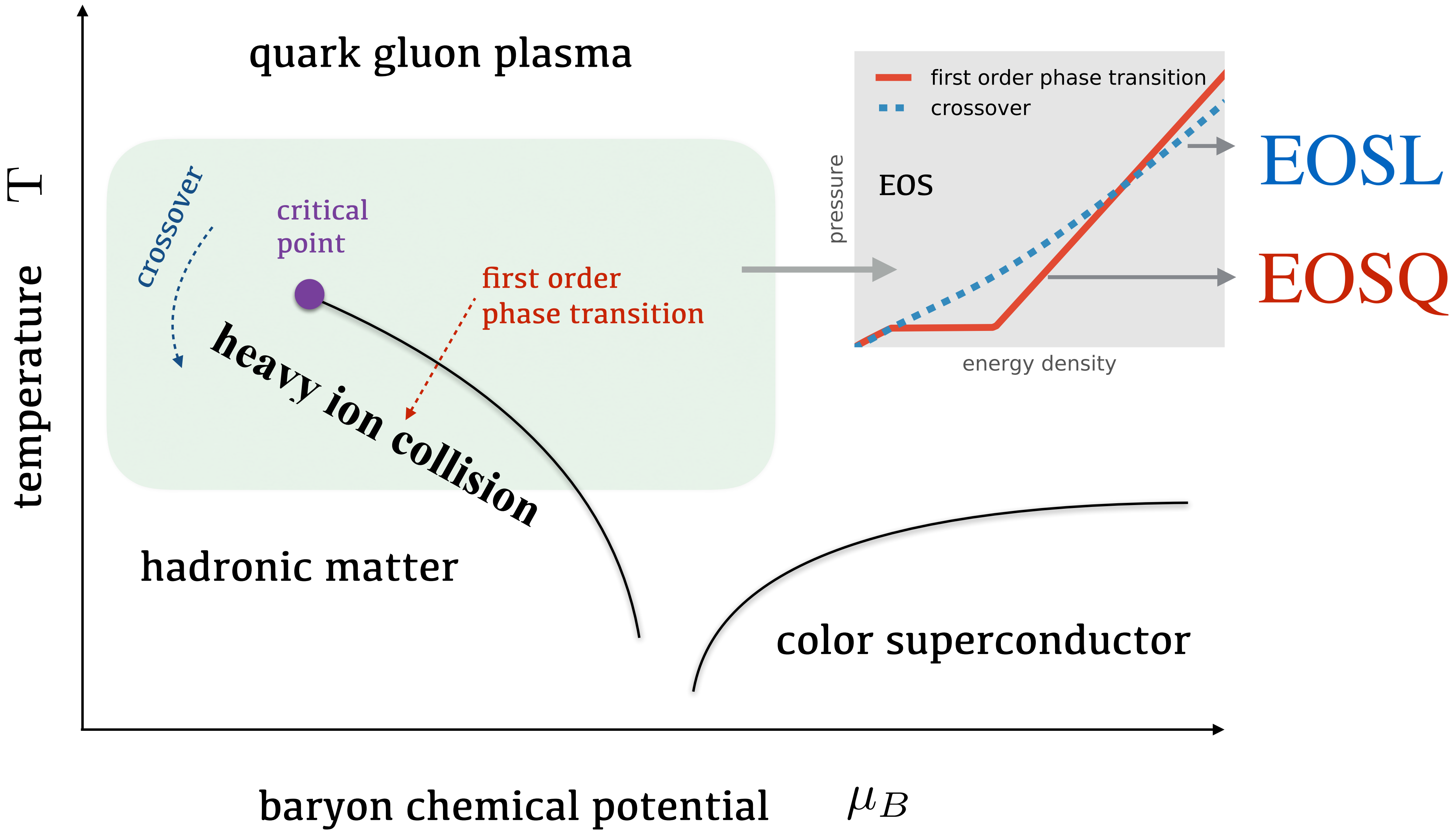


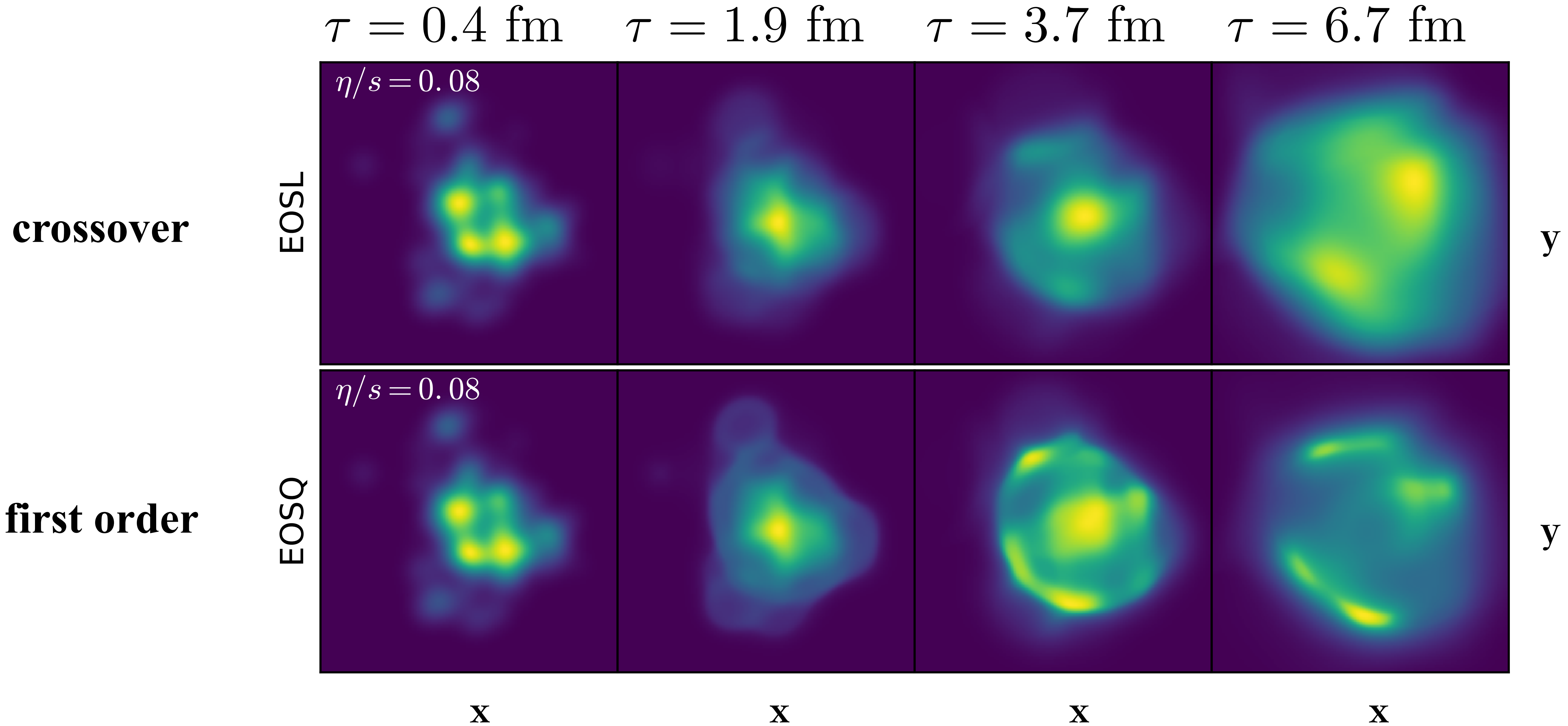
Regression problem that predicts the transverse distance between 2 colliding nucleus from final state particle distribution in 5x5 (pt, pz) bins

Deep learning for nuclear EoS

LG. Pang, K.Zhou, N.Su, H.Petersen, H. Stoecker, XN. Wang. Nature Communications 2018.

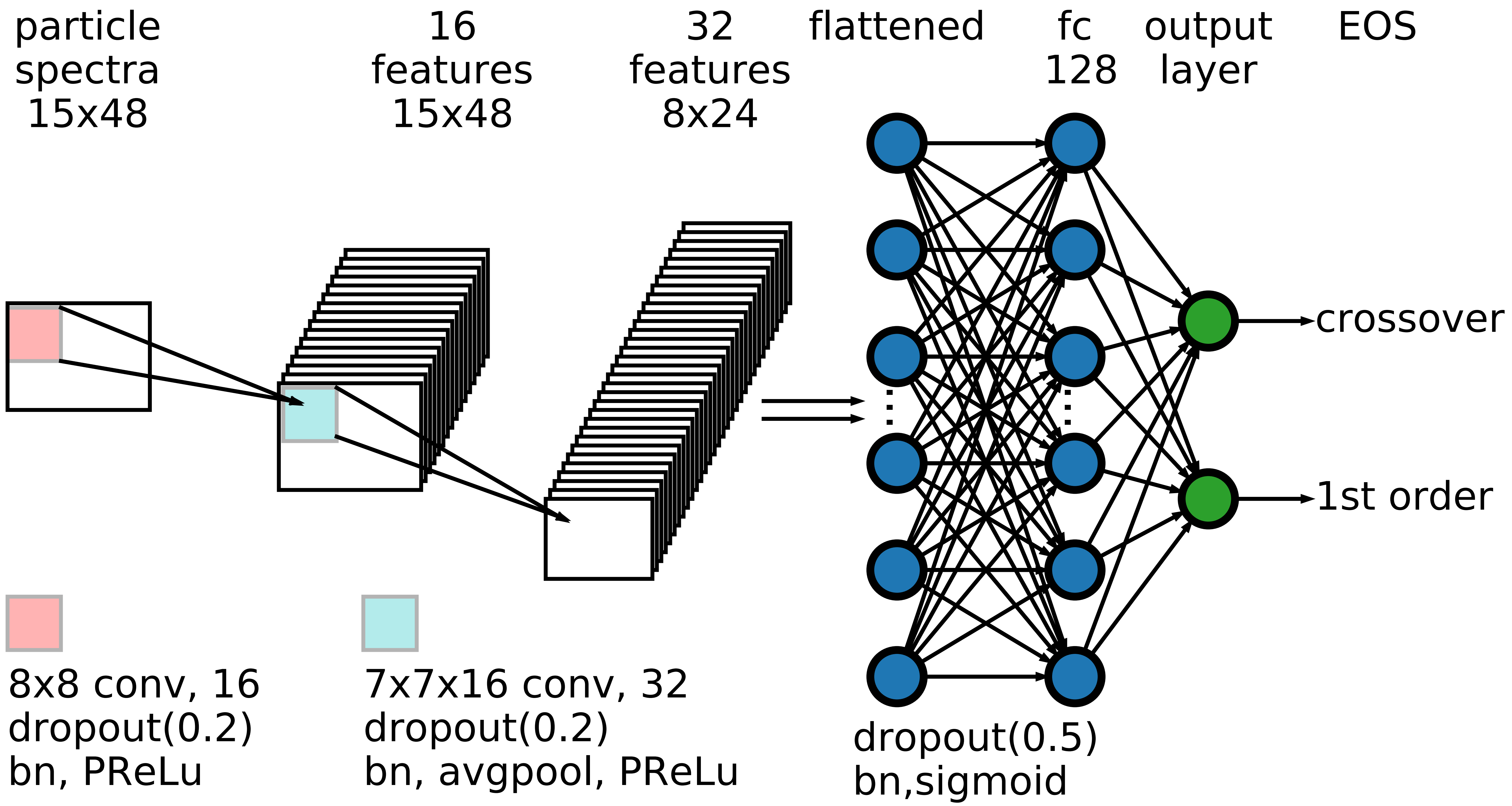






- Does the QCD phase transition signal survive the dynamical evolution of heavy ion collisions and exist in the final state output?
- Can deep neural network decode the phase transition type from complex output of heavy ion collisions

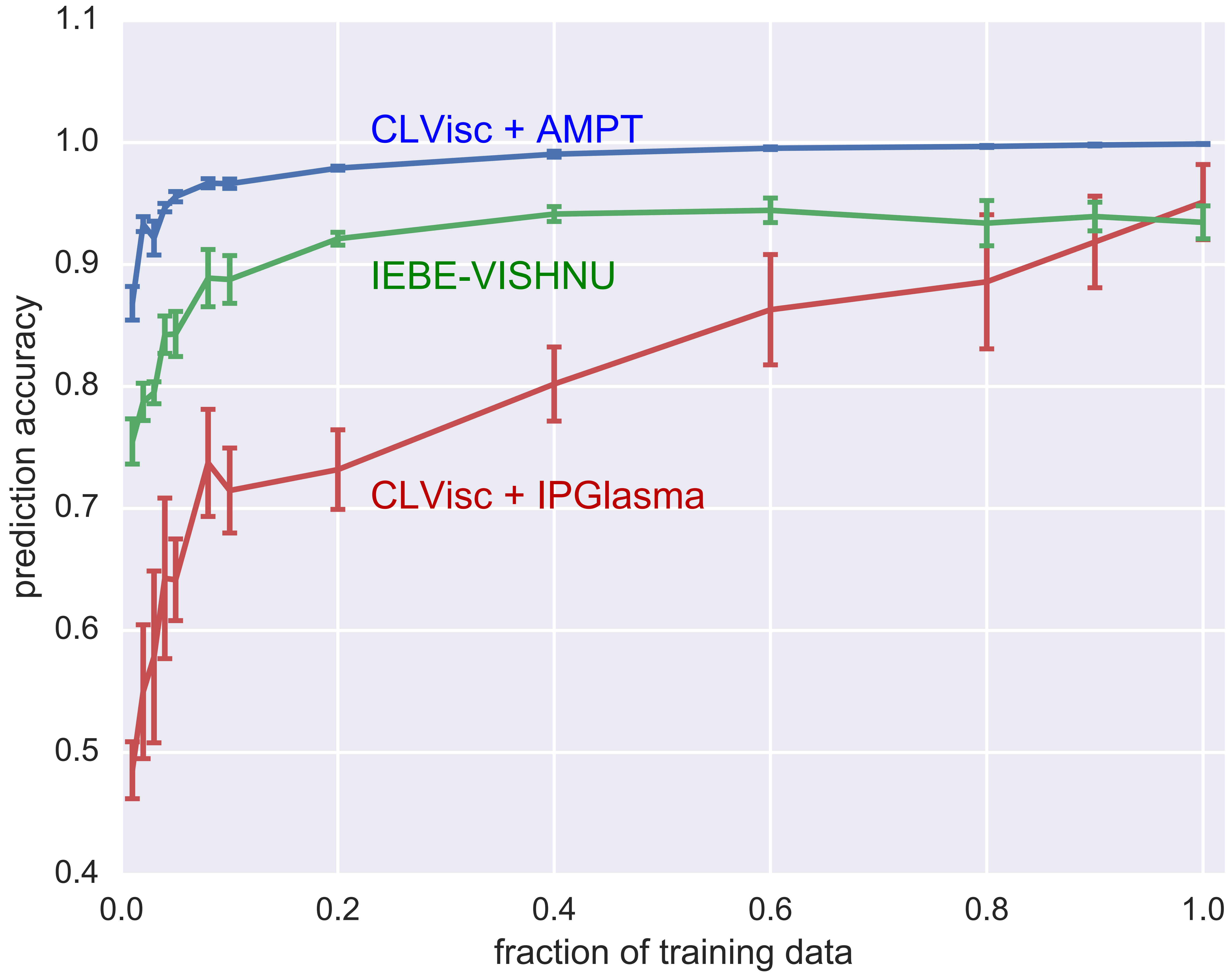
CNN architecture for EoS-meter



$$l(\theta) = \boxed{-\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\right]} + \boxed{\lambda||\theta||_2^2}$$

loss function — cross entropy — L2 regularization

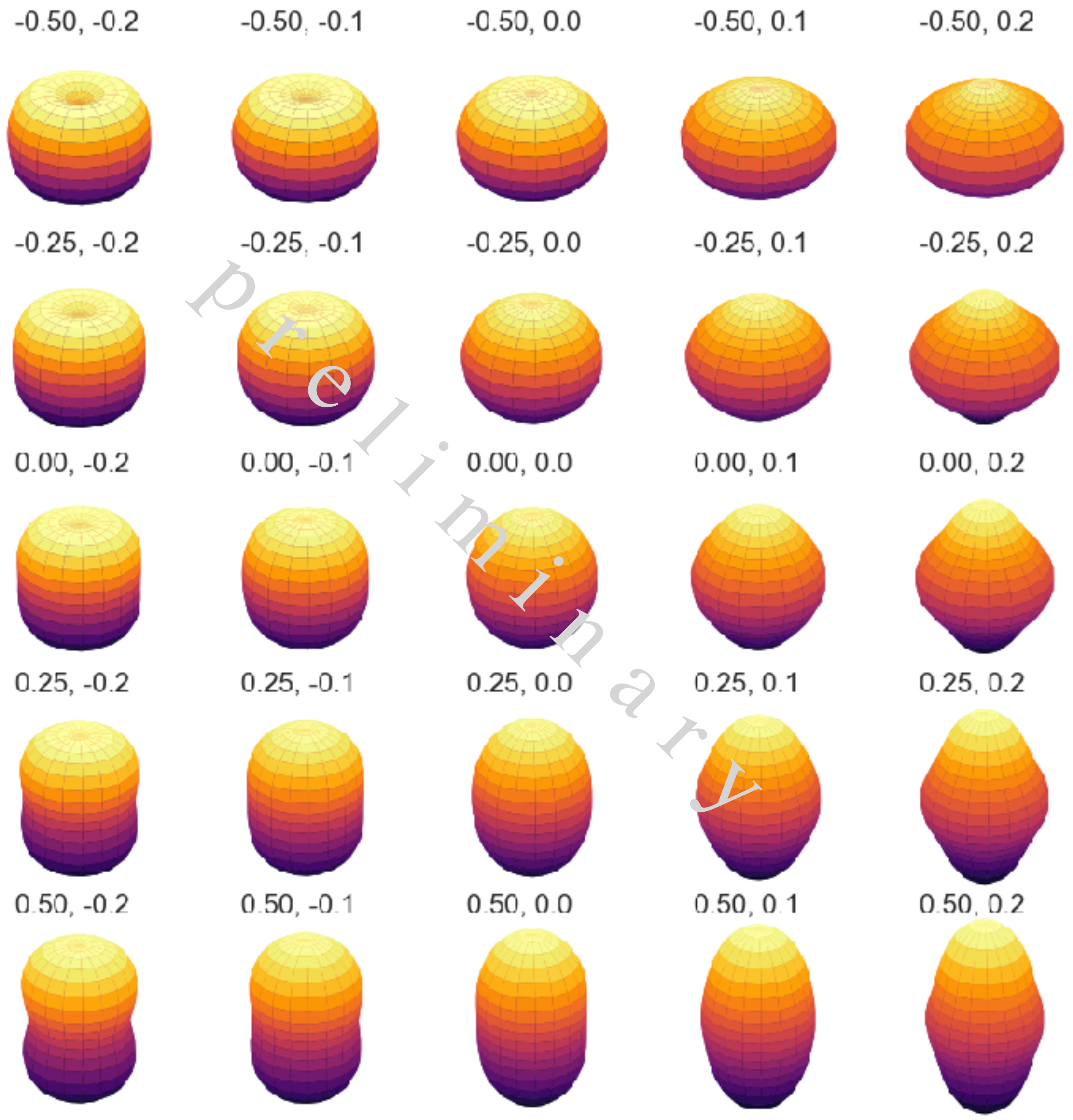
Results: classification accuracies



- 40000 events from CLVisc+AMPT model have been used for training
- Another 4000 events from CLVisc+AMPT have been used for testing
- 18000 events from another hydrodynamic model IEBE-VISHNU and CLVisc+IPGlasma model have been used for further testing

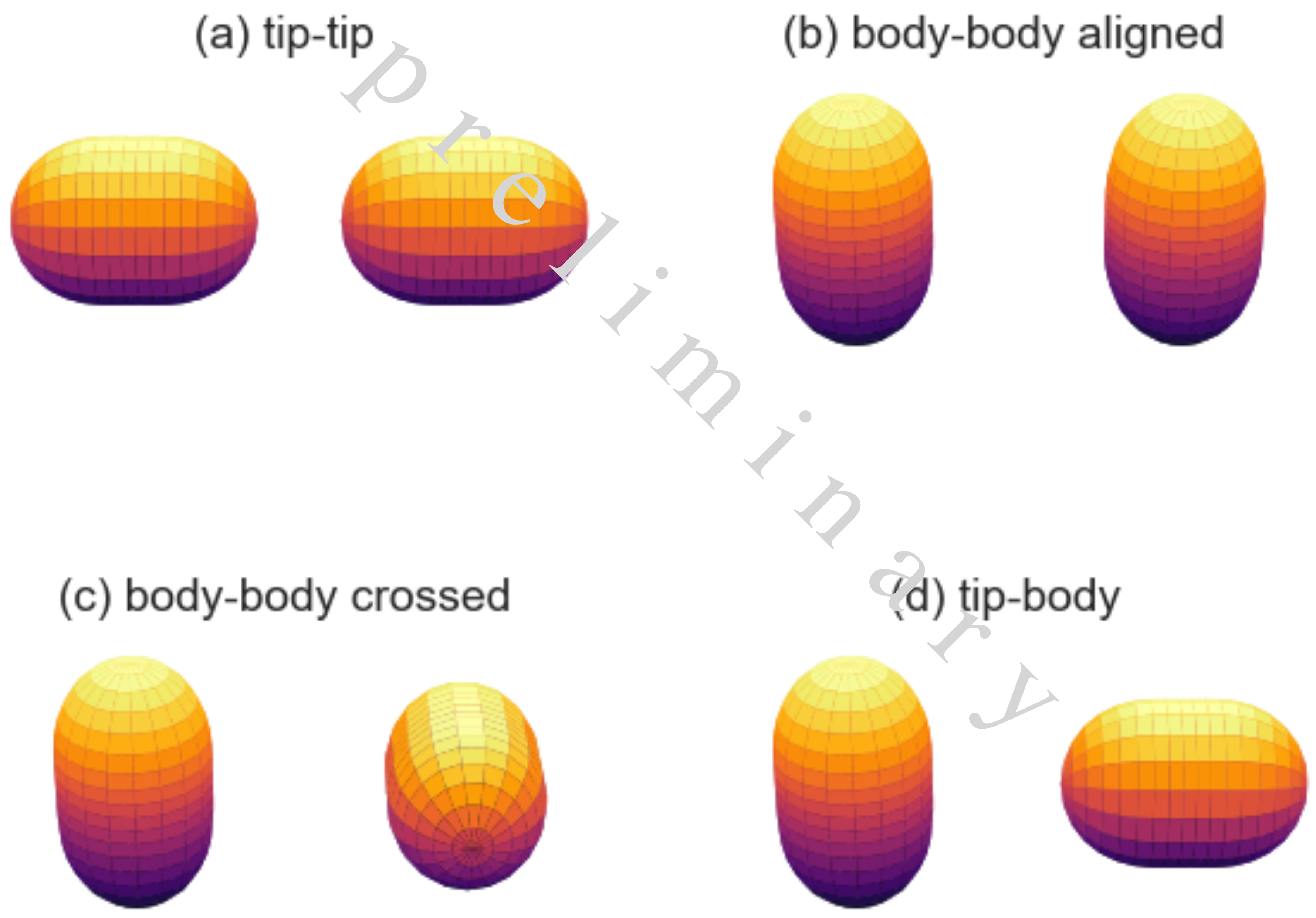
Machine learning nuclear deformation (preliminary)

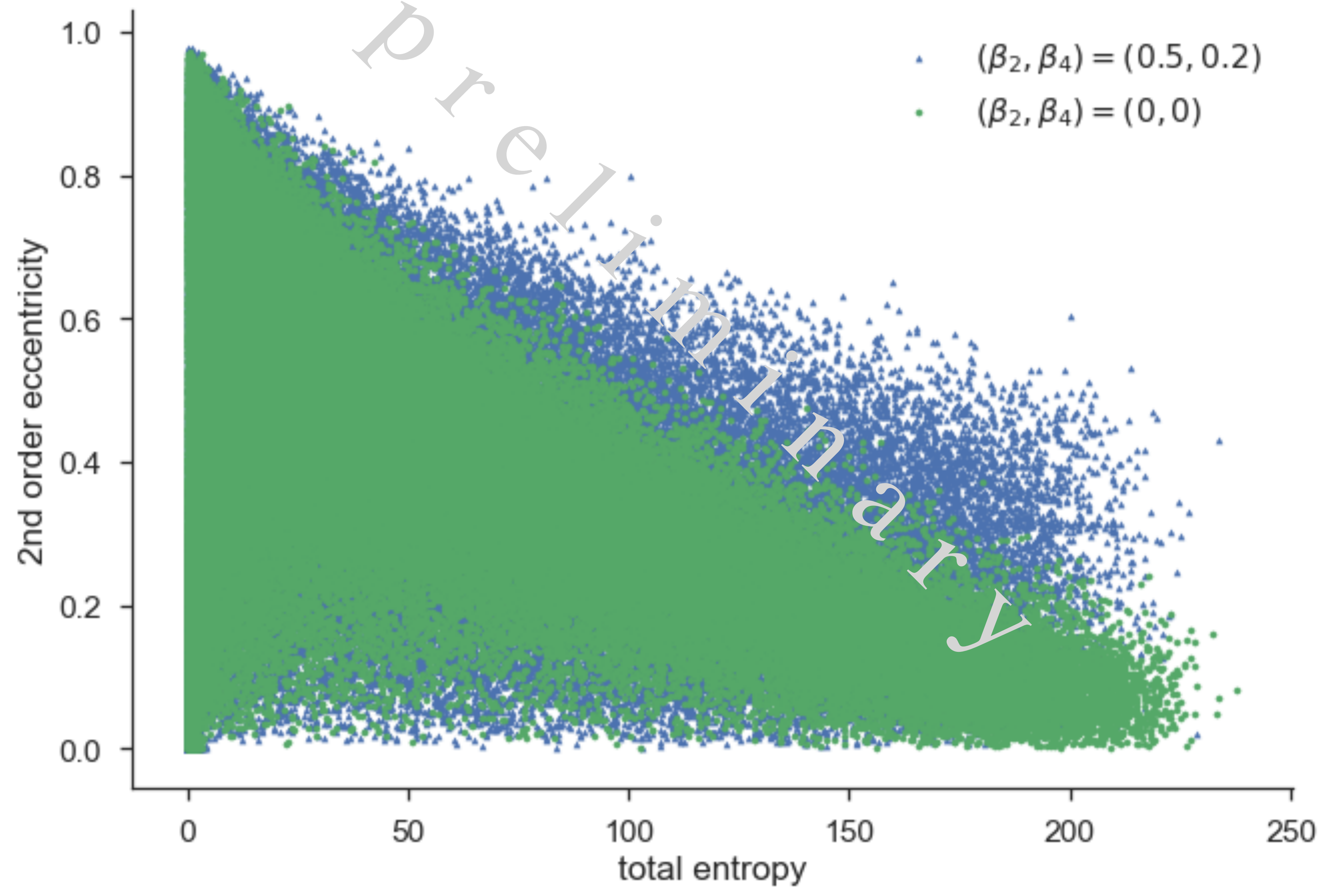
$$\rho(r,\theta,\phi) = \frac{\rho_0}{1 + e^{(r - R_0(1+\beta_2 Y_{20}(\theta) + \beta_4 Y_{40}(\theta)))/a}}$$



- β_2, β_4 controls the nuclear shape in the deformed Woods-saxon distribution
- Can we determine these 2 parameters from final state of heavy ion collisions?
- We test the idea with initial state total entropy and geometric eccentricity.

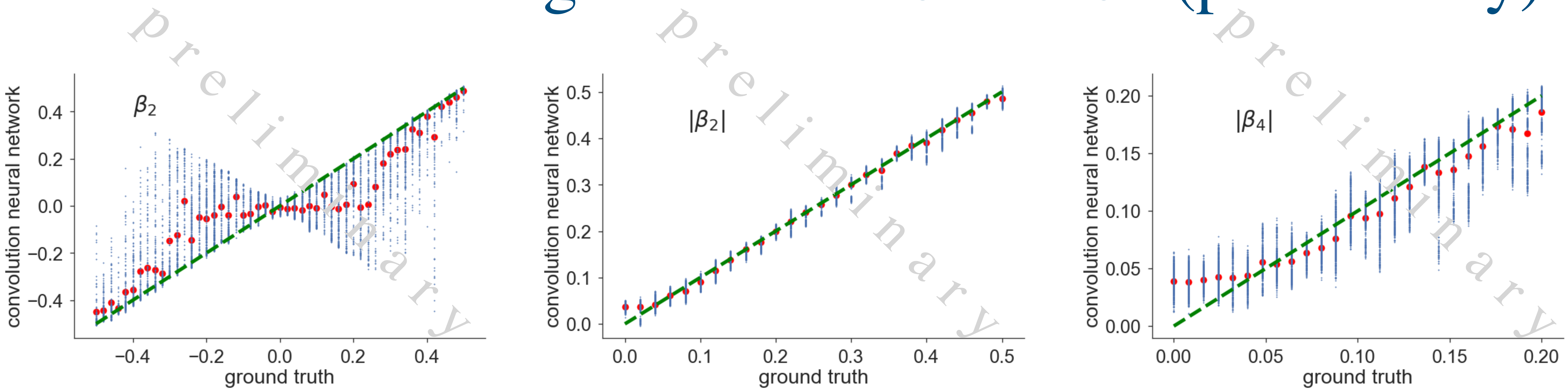
Machine learning nuclear deformation (preliminary)





- The event-by-event distributions of 2nd order eccentricity vs total entropy looks different for collisions of spherical and deformed nuclei.

Machine learning nuclear deformation (preliminary)



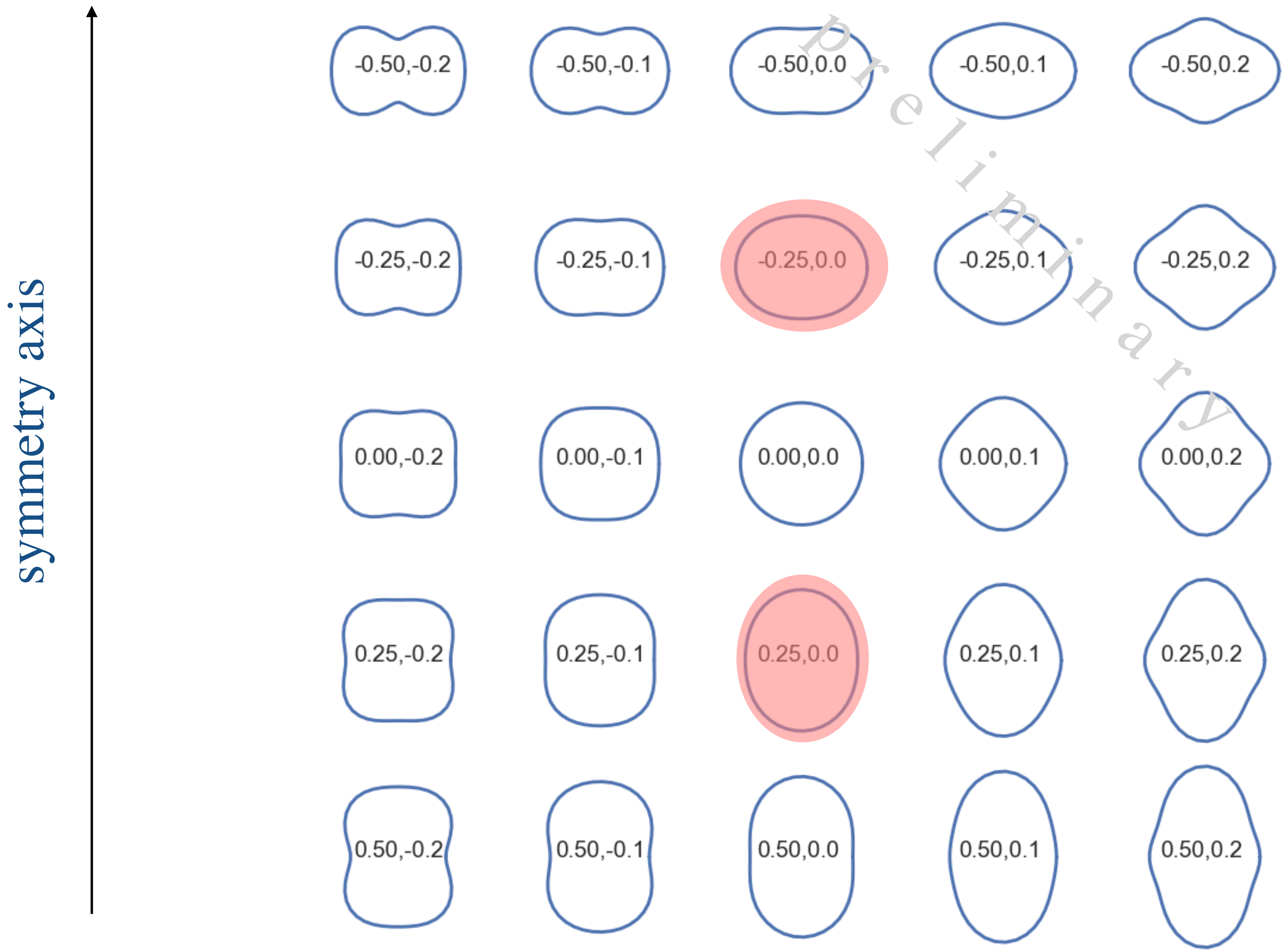
- A prior, it should be easy to learn the deformation parameters from heavy ion collisions.

- In practice, the brute force attempt using machine learning implies that it might be difficult to distinguish $|\beta_2|$ and $-|\beta_2|$

- We are inspired to change the regression target to $|\beta_2|$ and $|\beta_4|$

Machine learning nuclear deformation (preliminary)

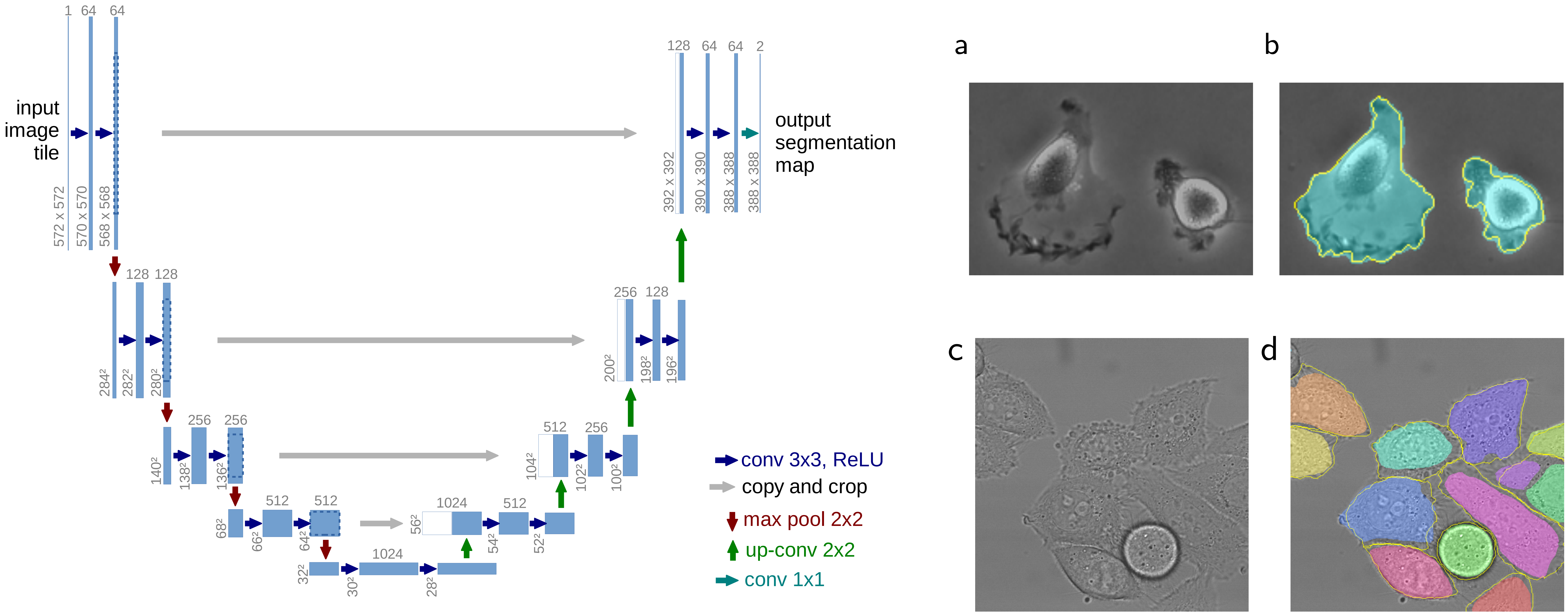
Reason for degeneracy in high energy heavy ion collisions



Stacked UNET for fast relativistic hydrodynamics

U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox



Stacked UNET for fast relativistic hydrodynamics

Applications of deep learning to relativistic hydrodynamics

H.Huang, B.Xiao, H.Xiong, Z.Wu, Y. Mu and H.Song arXiv: 1801.03334; NPA2018

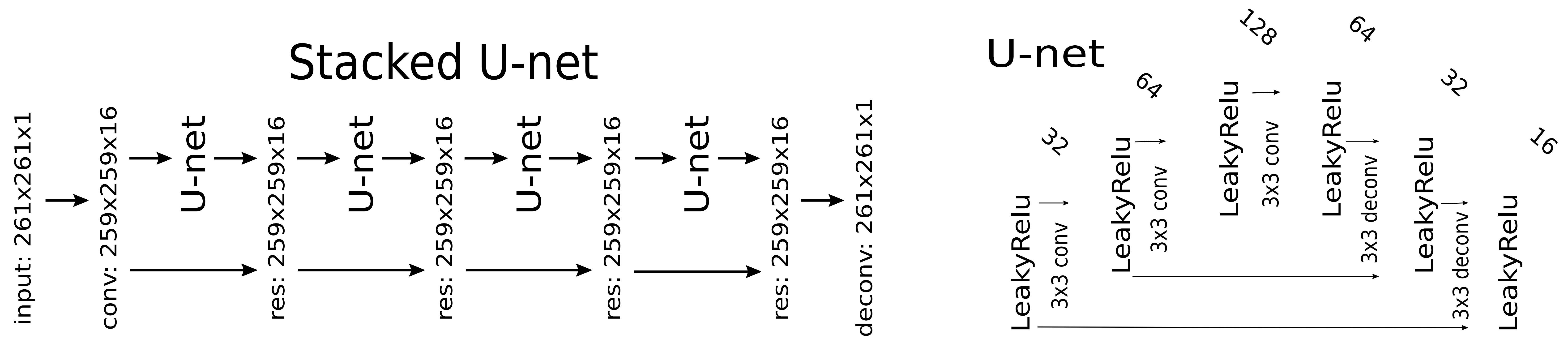
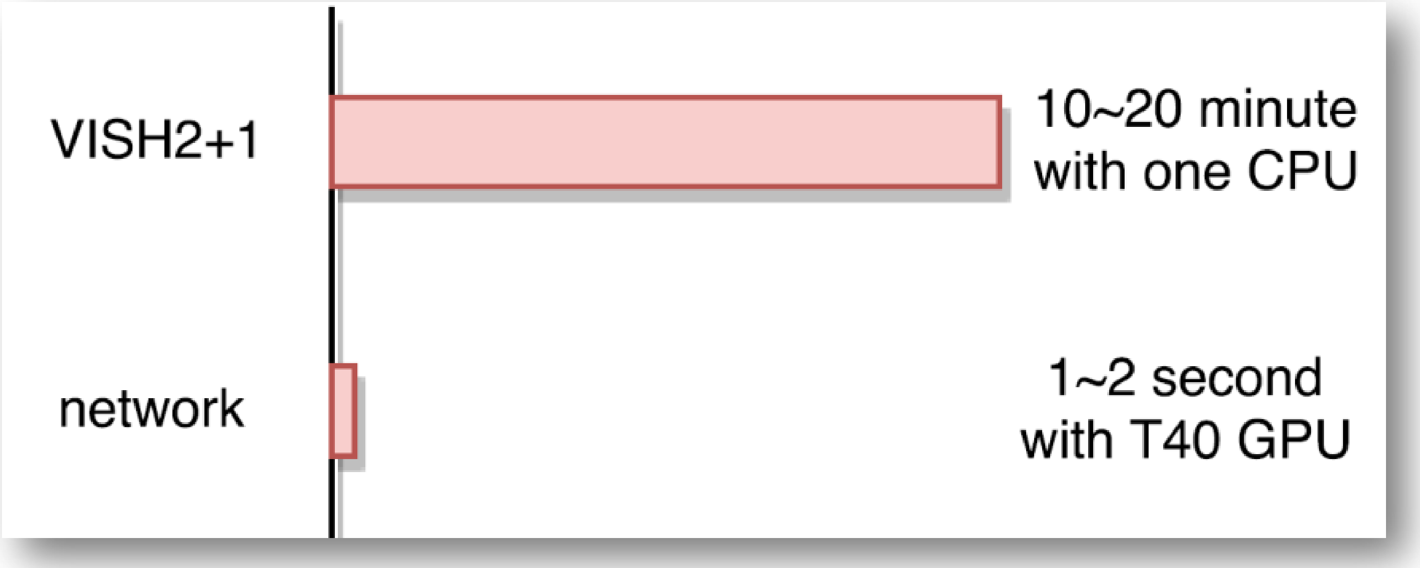


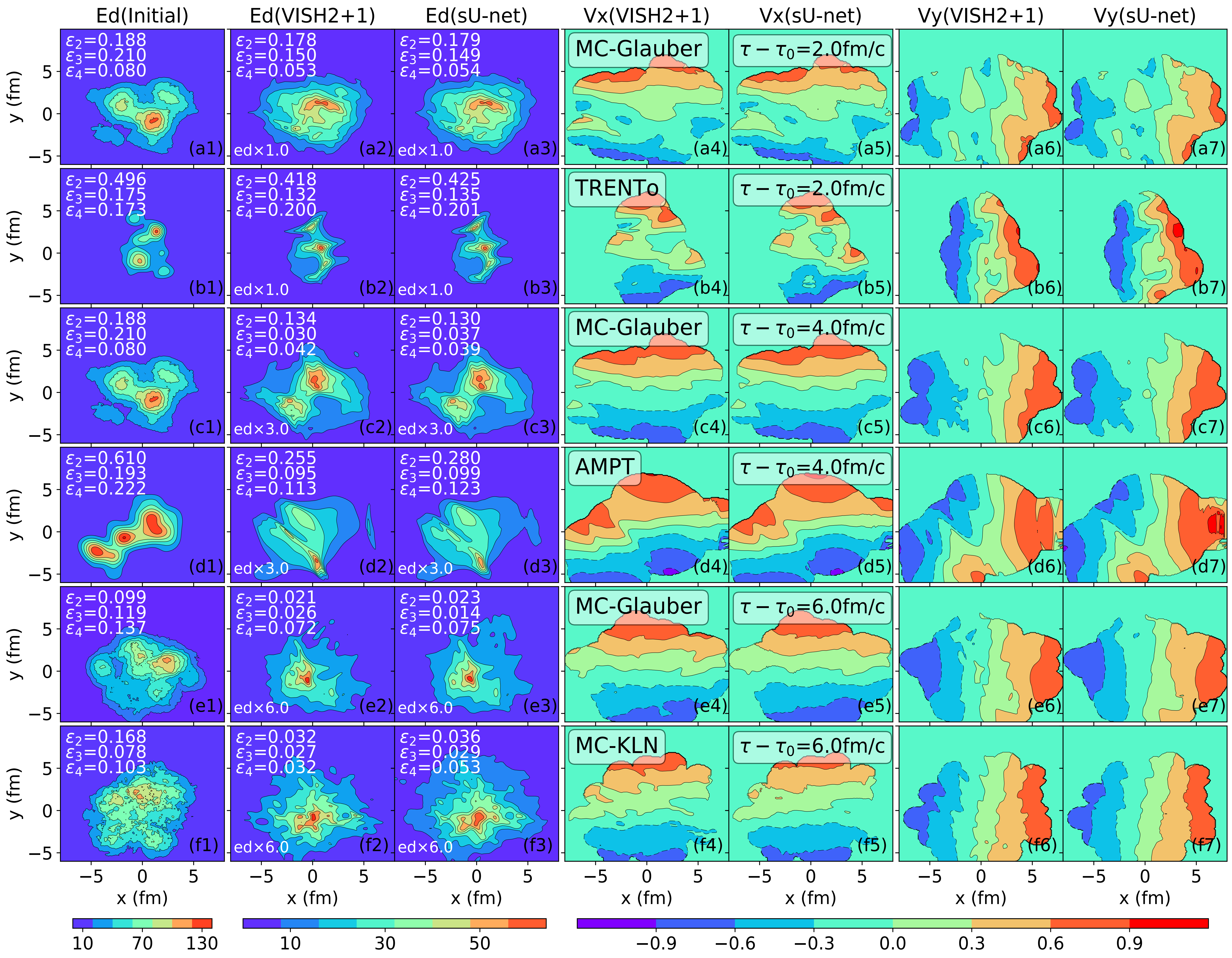
FIG. 1: An illustration of the encode-decode network, `stacked U-net`, which consists of the input and out layers and four residual U-net blocks. The right figure shows the U-net structure, and the depth of the hidden layer is written on the top of them.

The expansion of quark gluon plasma is learned in the image translation task using stacked UNET.

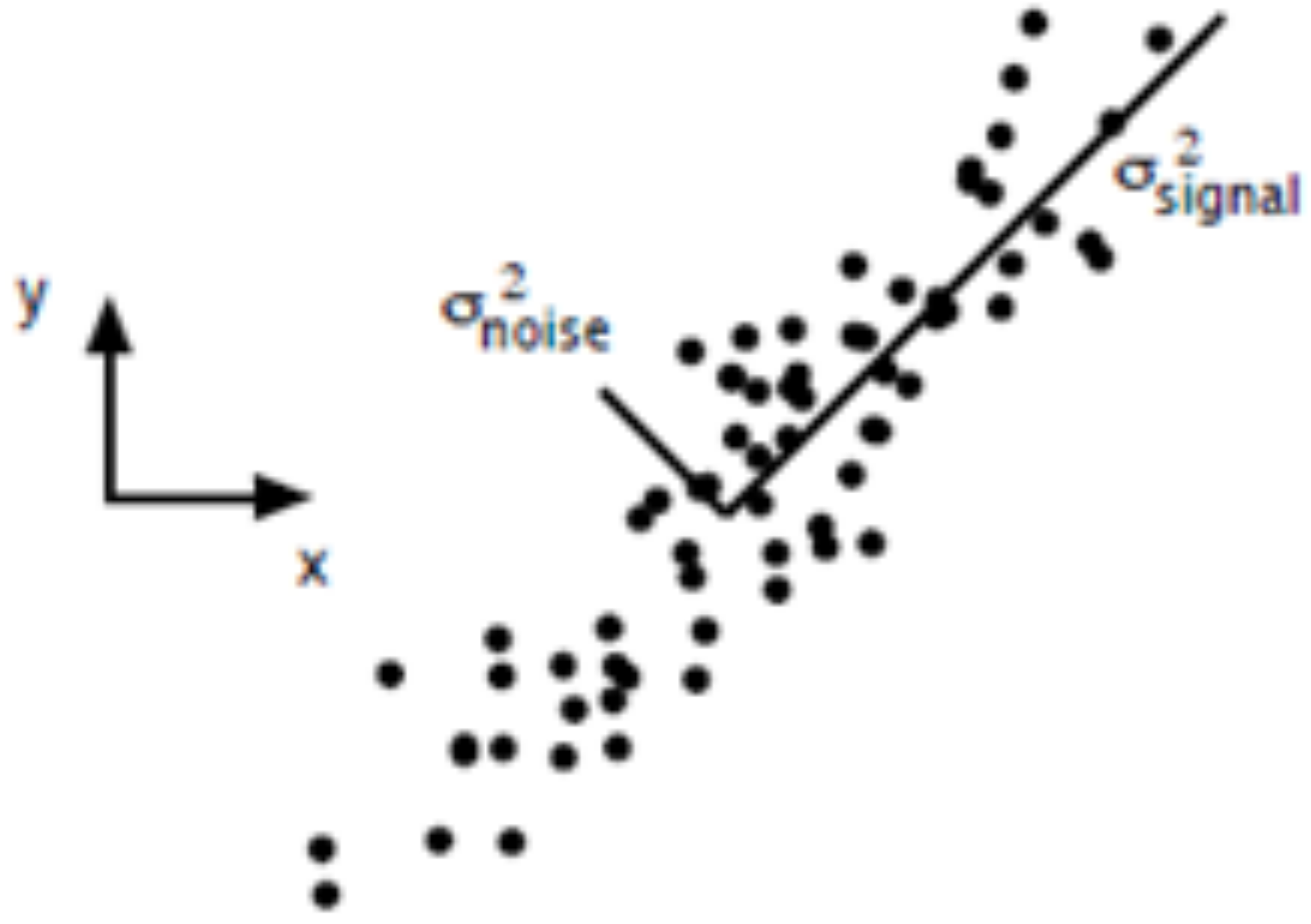
$$\nabla_\mu T^{\mu\nu} = 0$$

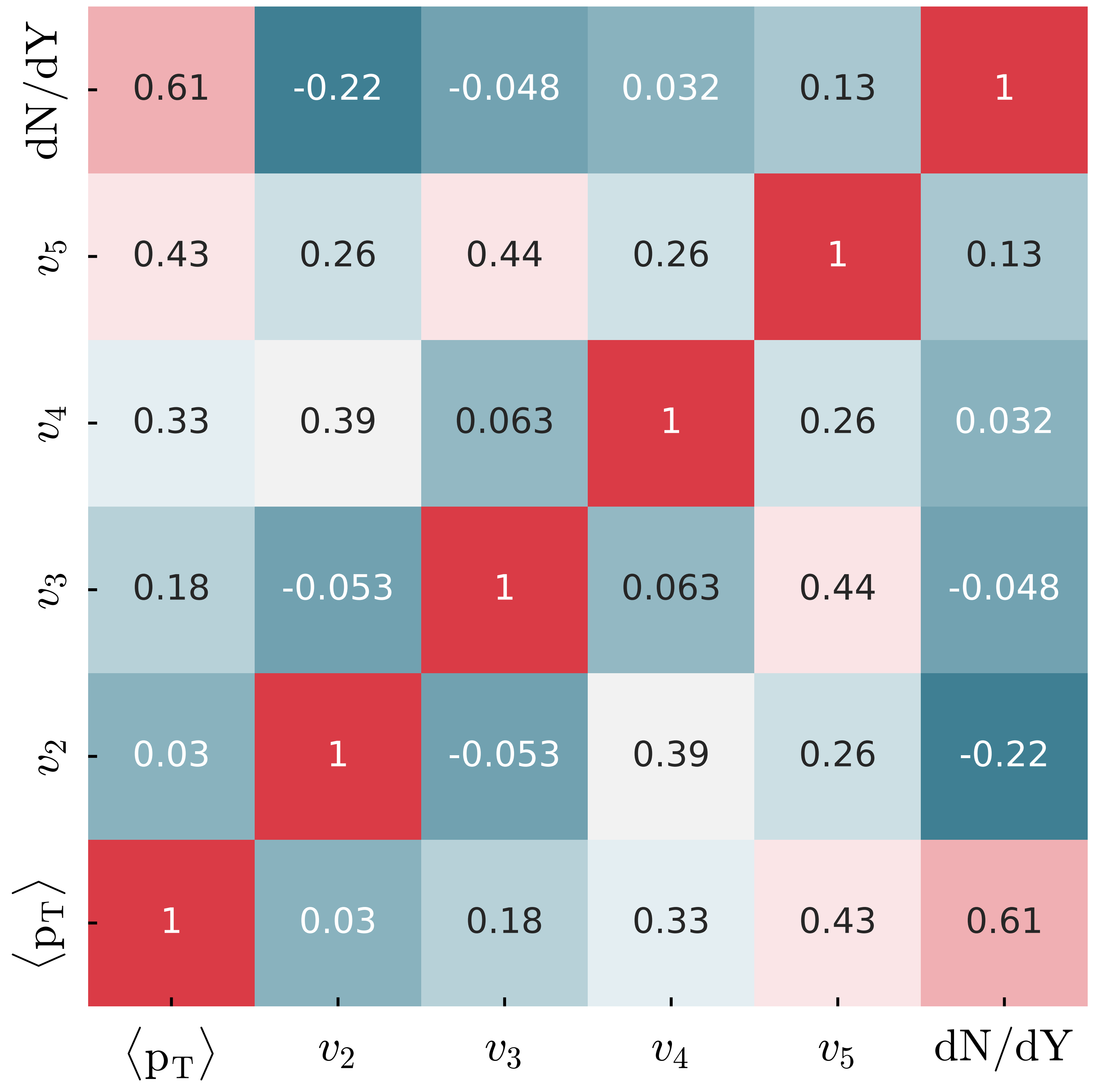


Stacked UNET for fast relativistic hydrodynamics



Unsupervised learning: Principle component analysis (PCA)





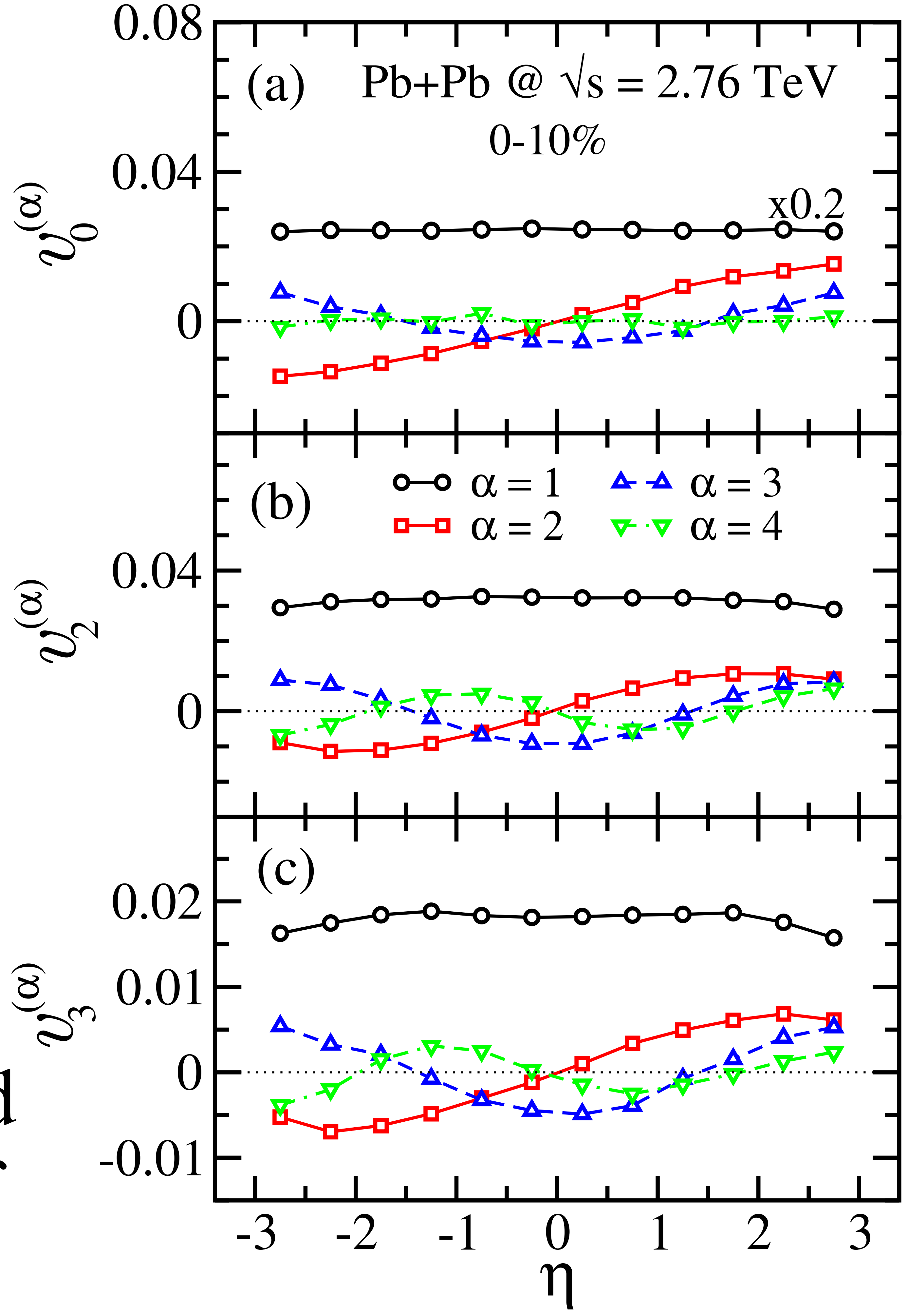
1. PCA looks for most important features (principle components) of data
2. Principle components means largest eigenvalues of the covariance matrix

Unsupervised learning: Principle component analysis (PCA)

Principal component analysis of event-by-event fluctuations

PRL 114, 152301 (2015), Rajeev S. Bhalerao, Jean-Yves Ollitrault, Subrata Pal, Derek Teaney

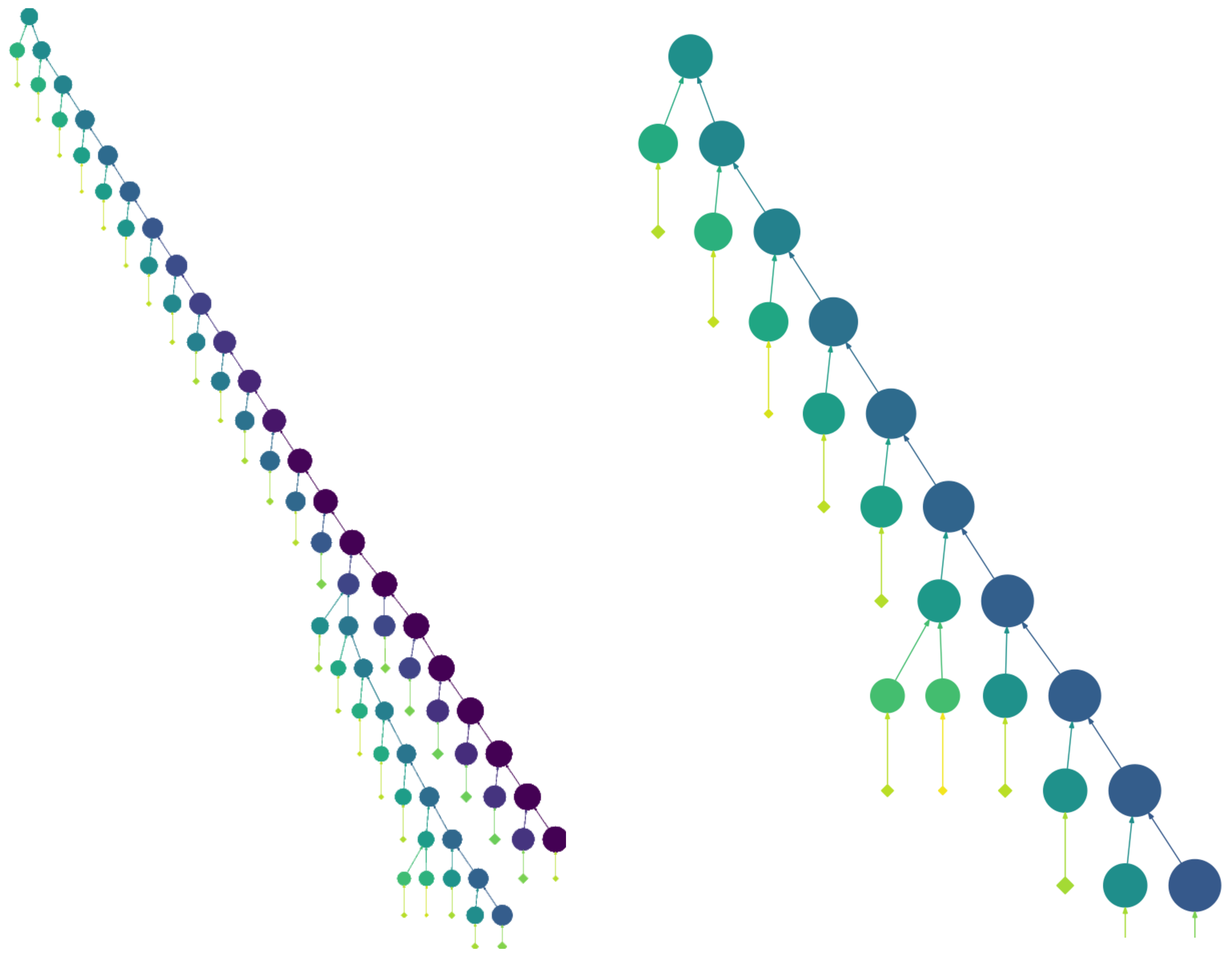
- 12 pseudo-rapidity bins, covariance matrix 12x12
- Multiplicity v0:
  - global fluctuation: $v_0^1 \approx 12\%$
  - forward-backward asymmetry: ~ 1/60 global fluctuation
  - next mode (even parity): ~ 1/300 global fluctuations
- The leading modes in (b) and (c) corresponds to the usual elliptic and triangle flow while subheading modes are attributed to small twist of event plane angles and flow amplitude fluctuations.



Other unsupervised learning

- The unsupervised learning algorithms are extremely useful in experimental data analysis, because of lacking labeled training data.
- k-means clustering (anomaly detection, new physics finding)
- (Denoising/Variational) Autoencoders (detector efficiency correction)
- Generative Adversarial Network (GAN) (for super resolution, image translation, event generator)

Future challenges — deep learning for jet tagging



gluon jet   quark jet

Image from TaoLi Cheng

Deep learning in color: towards automated quark/gluon jet discrimination

Patrick T. Komiske,[a] Eric M. Metodiev,[a] and Matthew D. Schwartz[b]

[a] *Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[b] *Department of Physics, Harvard University, Cambridge, MA 02138, USA*
E-mail: pkomiske@mit.edu, metodiev@mit.edu, schwartz@physics.harvard.edu

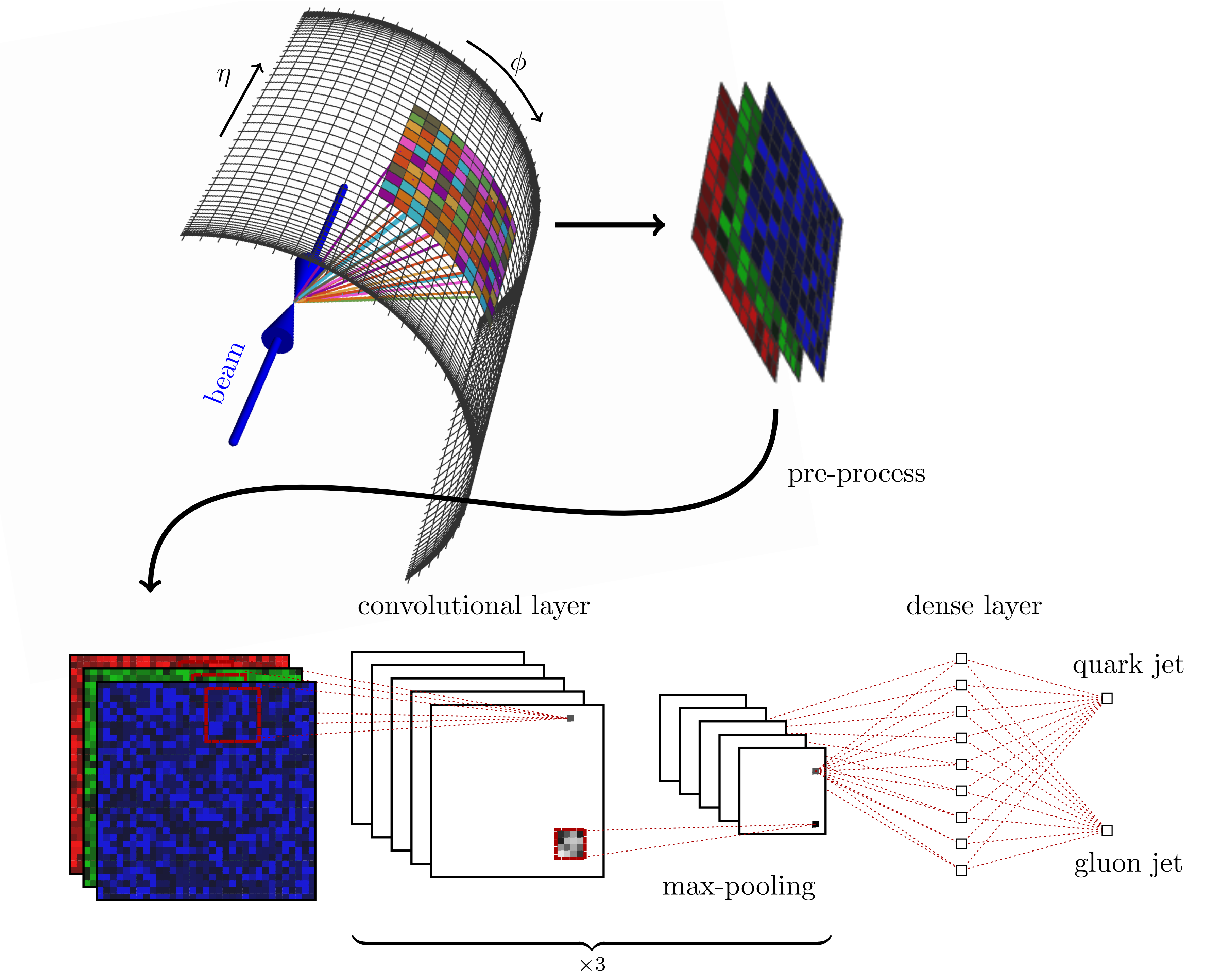
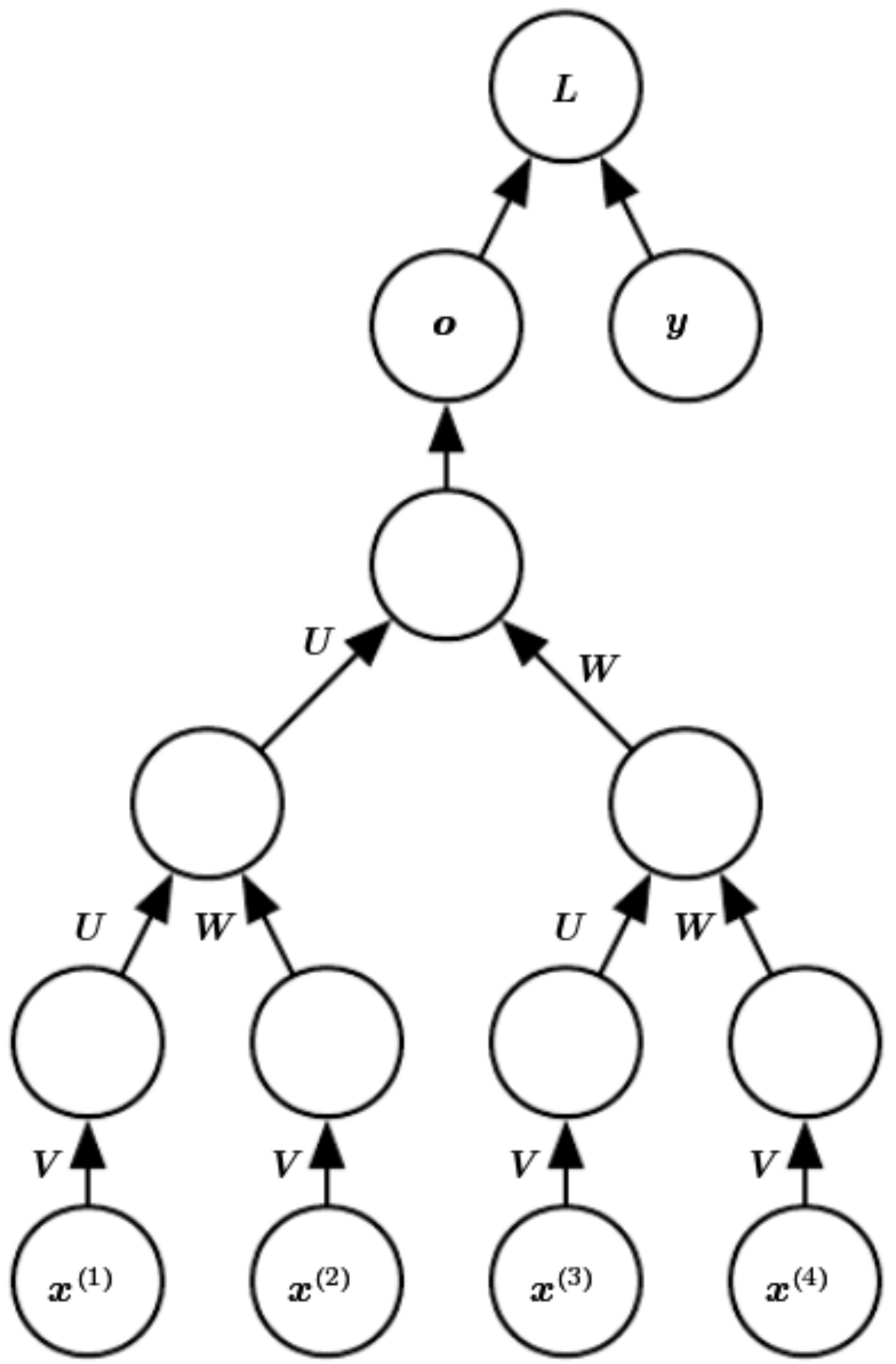


Figure 2: An illustration of the deep convolutional neural network architecture. The first layer is the input jet image, followed by three convolutional layers, a dense layer and an output layer.

Future challenges — deep learning for jet tagging



Recursive neural network

For the leaves, $h_k = u_k = \sigma(xW + b)$

For other nodes,

$$\mathbf{h}_k = \sigma\left(W_h \begin{bmatrix} \mathbf{h}^{\text{jet}}_{k_L} \\ \mathbf{h}^{\text{jet}}_{k_R} \\ \mathbf{u}_k \end{bmatrix} + b_h \right)$$

where $h^{\text{jet}}_{k_L}$ and $h^{\text{jet}}_{k_R}$ are the hidden information of the left and right children of node k.

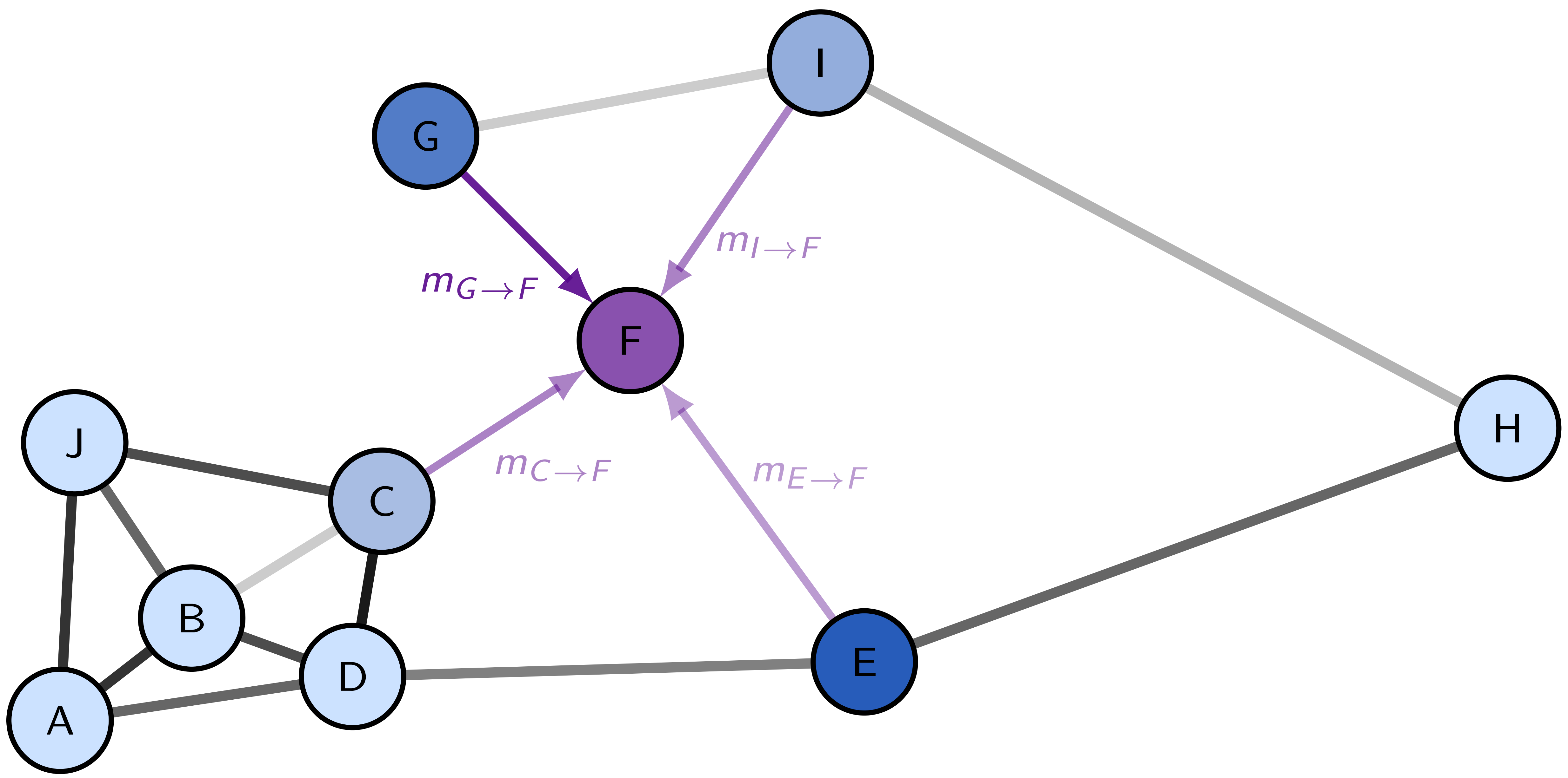
Cross entropy loss function is used for network prediction o and true answer y.

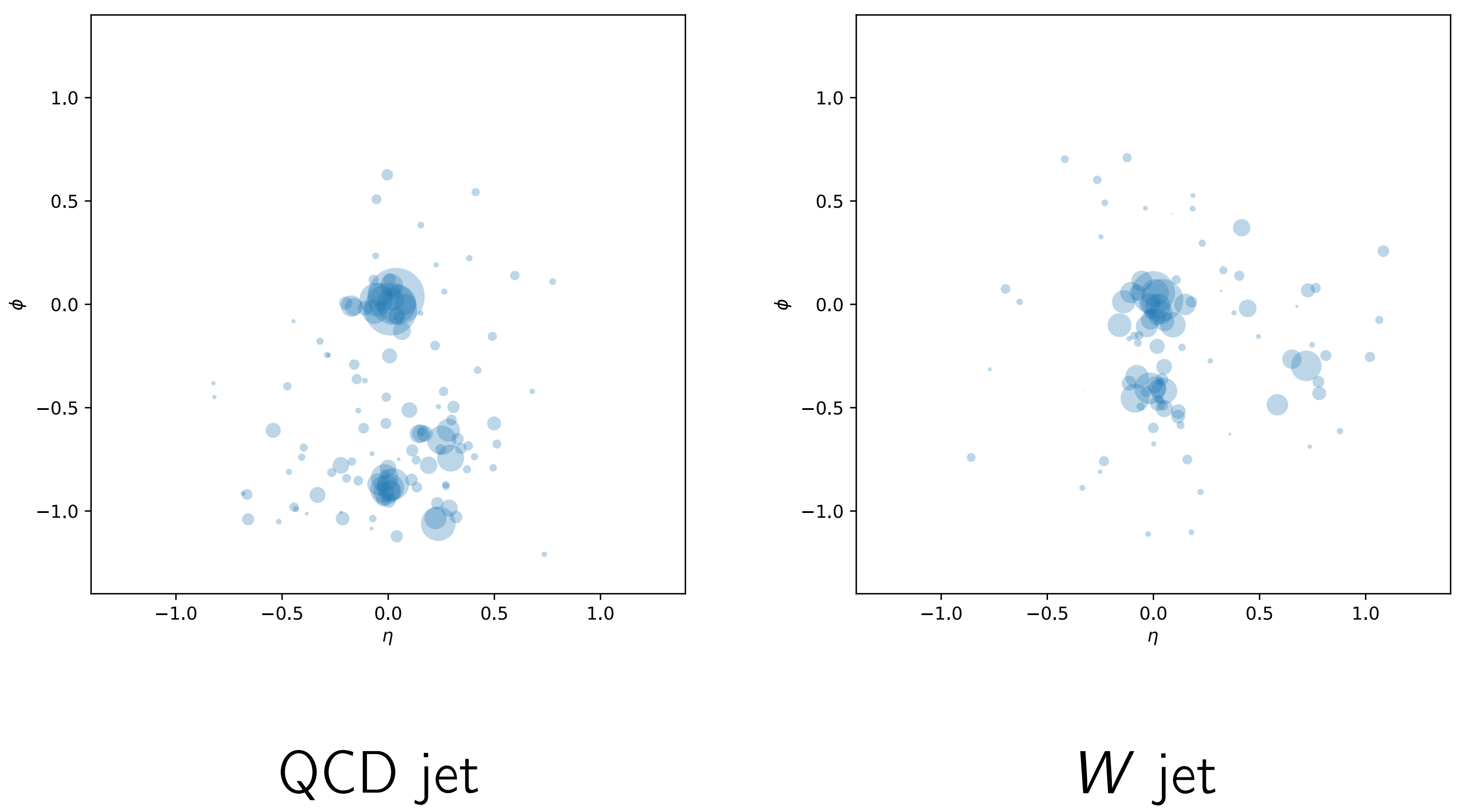
G. Louppe, K. Cho, C. Becot, and K. Cranmer, *QCD-Aware Recursive Neural Networks for Jet Physics*, arXiv:1702.00748.

Recursive Neural Networks in Quark/Gluon Tagging
- Cheng, Taoli Comput.Softw.Big Sci. 2 (2018) no.1, 3 arXiv:1711.02633

Future challenges — deep learning for W jet tagging

Jets as graphs: W tagging with neural message passing,
Isaac Henrion, Johann Brehmer, Joan Bruna, Kyunghyun Cho,
Kyle Cranmer, Gilles Louppe, Gaspar Rochette





$$\tilde{m}_j^t = f(h_j^{t-1})$$
$$m_{j\to i}^t = \sigma(A_{ij}\tilde{m}_j^t)$$
$$h_i^t = \mathrm{GRU}(h_i^{t-1}, \Sigma_j m_{j\to i}^t)$$

State of the art classification result:

Model	Iterations	$R_{\epsilon=50\%}$
Rec-NN (no gating)	1	70.4 ± 3.6
Rec-NN (gating)	1	$\mathbf{83.3 \pm 3.1}$
MPNN (directed)	1	89.4 ± 3.5
MPNN (directed)	2	$\mathbf{98.3 \pm 4.3}$
MPNN (directed)	3	85.9 ± 8.5
MPNN (identity)	3	74.5 ± 5.2
Relation Network	1	67.7 ± 6.8

Future challenges — deep learning for heavy ion jet

b-jet tagging in p+Pb collisions

Machine and deep learning techniques in heavy-ion collisions with ALICE

Rüdiger Haake* **for the ALICE collaboration**
CERN
E-mail: ruediger.haake@cern.ch

Probing heavy ion collisions using quark and gluon jet substructure with machine learning

Yang-Ting Chien
Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning

Andrew J. Larkoski*
Physics Department, Reed College, Portland, OR 97202, USA

Ian Moult†
Berkeley Center for Theoretical Physics, University of California, Berkeley, CA 94720, USA and Theoretical Physics Group, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Benjamin Nachman‡
Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
(Dated: September 15, 2017)

Automated Discovery of Jet Substructure Analyses

Yue Shi Lai
Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Machine Learning for Heavy Flavor Jet Tagging at RHIC
Speaker: George Halal

Countless applications in P+P jets, few in heavy ion jets.

Future challenges — deep learning for heavy ion jets

A fast and reliable Monte Carlo event generator!

Jetscape

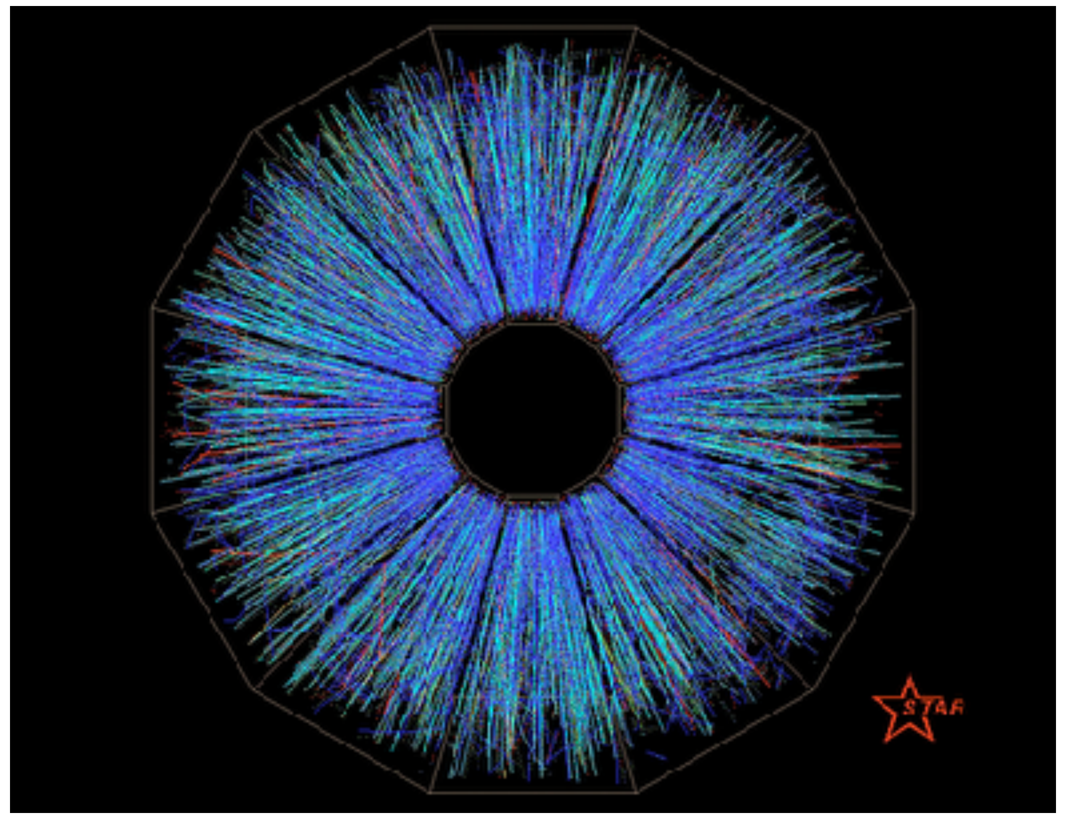
Summary

- Machine learning has long been employed in heavy ion collisions.
- More studies on heavy ion jets will come using Bayesian analysis and deep neural network.
- Monte Carlo simulation (JETSCAPE) is very important to accumulating big amount of labeled training data for heavy ion collisions.

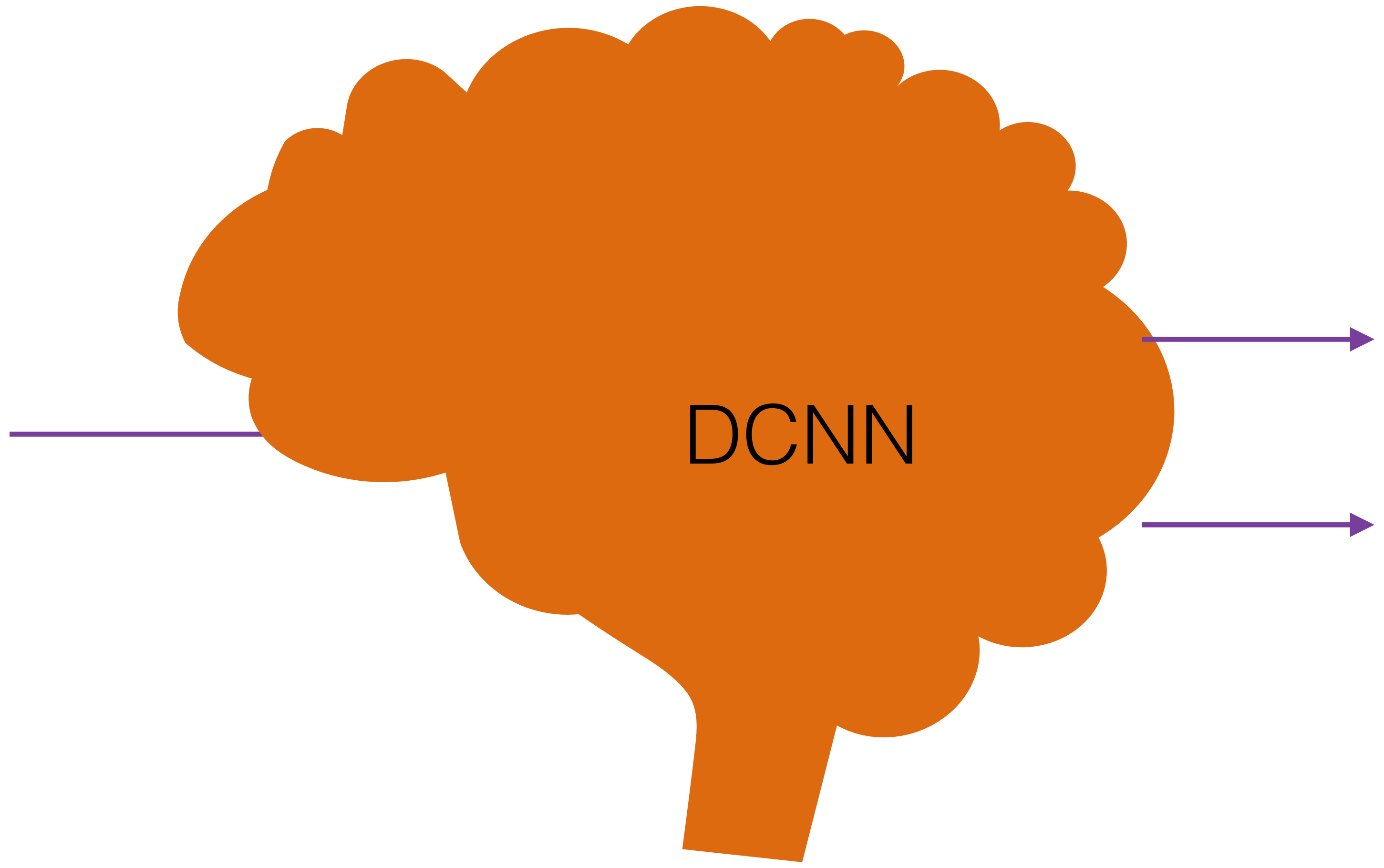
Backups

Sorry for those whose work I did not have time to mention in the short time!

Classifying two phase transition regions



$\rho(p_T, \Phi)$



crossover

1st order phase transition

Key idea for this proof-of-principle study

Supervised learning using deep convolution neural network with big amount of labeled training data (spectra, EoS type) from event-by-event relativistic hydrodynamics.

Open Source Libraries

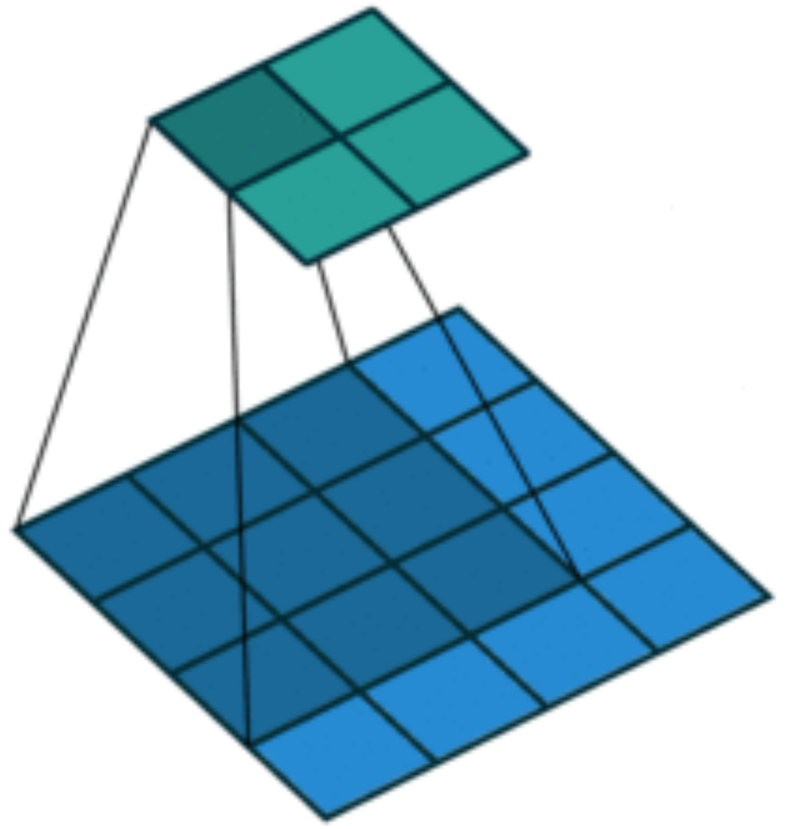
Keras + TensorFlow in the present study

Keras is a high level neural network library, written in Python and capable of running on top of either TensorFlow or Theano.

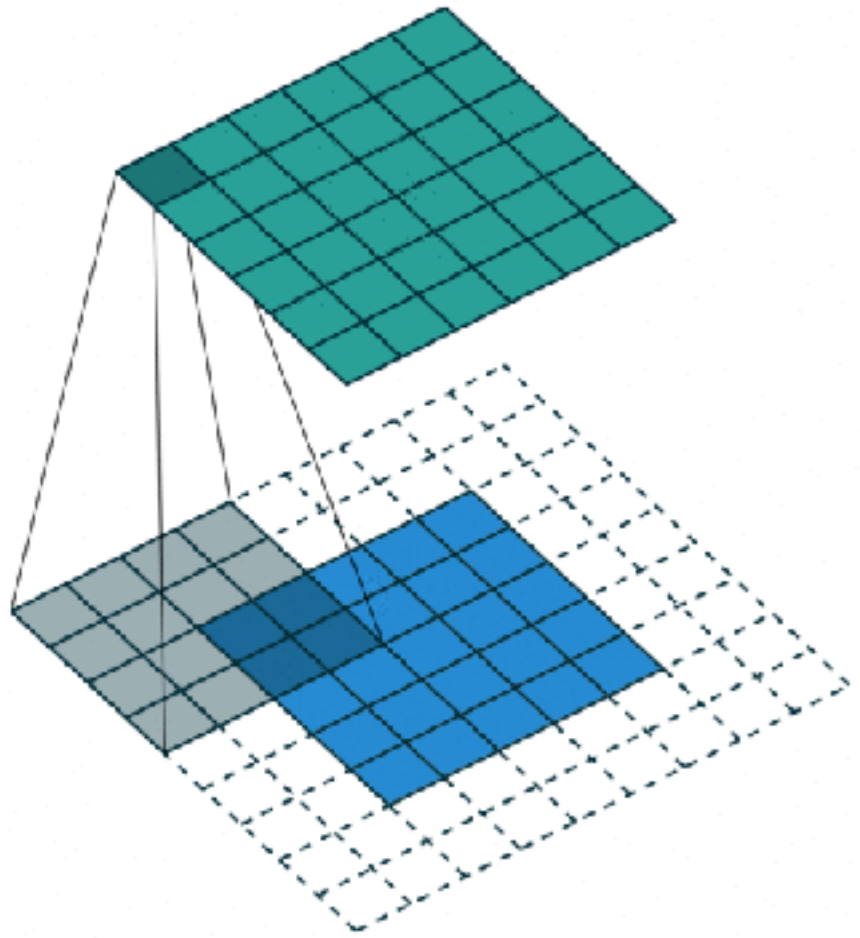
```
# Build one fully connected neural network (784->10->10 neurons) in Keras, for MNIST

from keras.models import Sequential
from keras.layers import Dense, Activation

model = Sequential()
model.add(Dense(output_dim=10, input_dim=784))
model.add(Activation("relu"))
model.add(Dense(output_dim=10))
model.add(Activation("softmax"))
model.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['accuracy'])
```

2017/01/15: Keras becomes a part of Tensorflow.

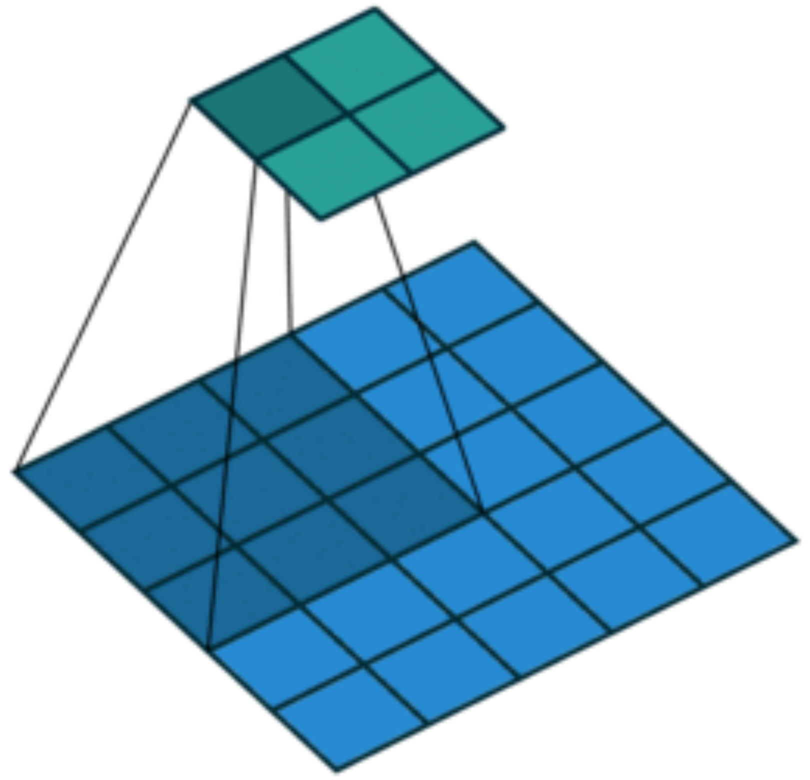
Convolution Neural Network — 2D



no padding, no stride



arbitrary padding, no stride



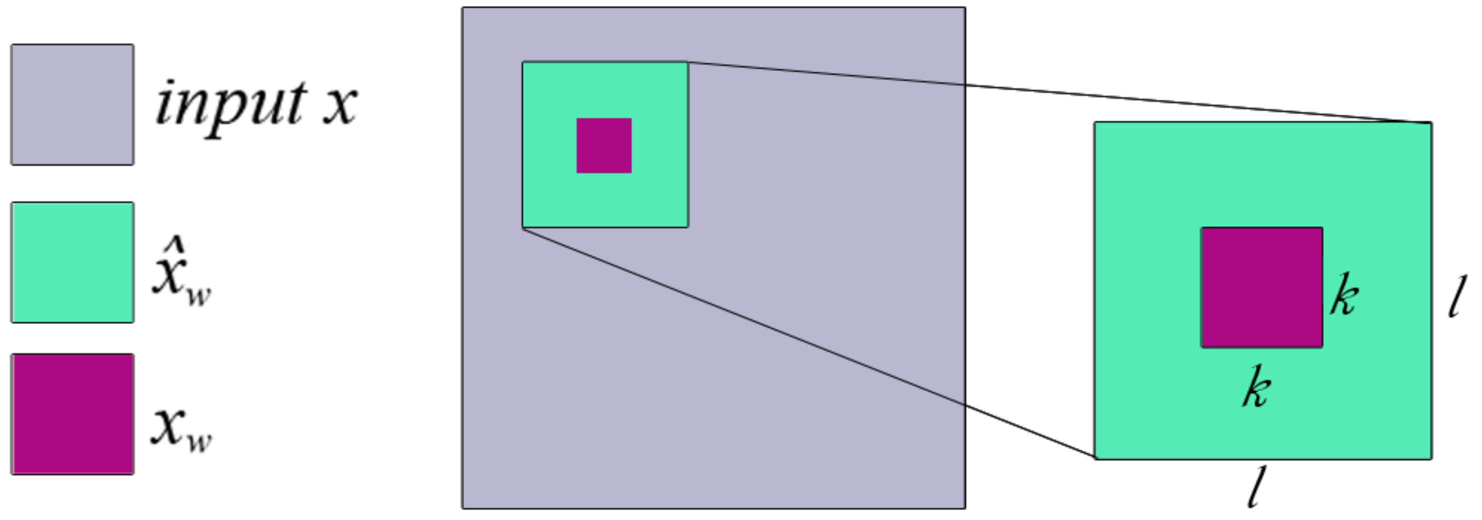
no padding, stride

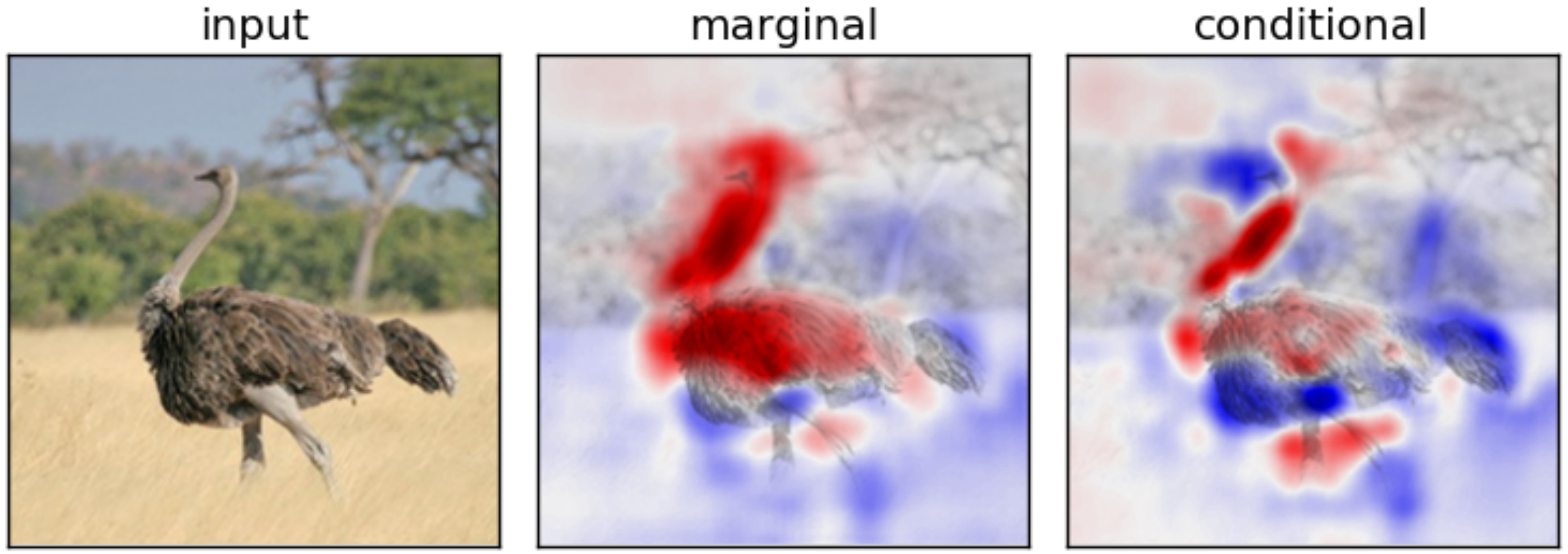
animations from: https://github.com/vdumoulin/conv_arithmetic

Prediction Difference Analysis

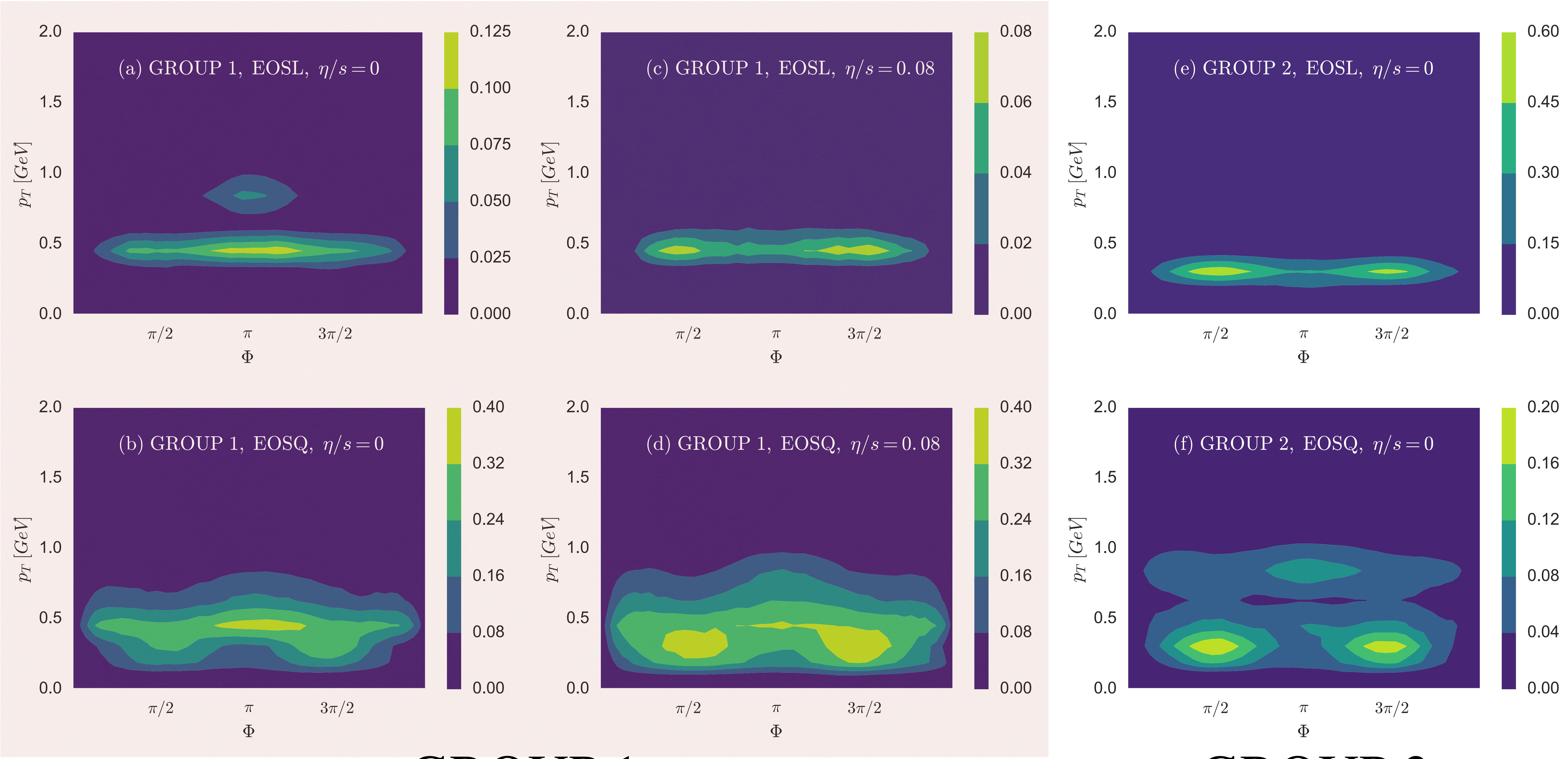
VISUALIZING DEEP NEURAL NETWORK DECISIONS: PREDICTION DIFFERENCE ANALYSIS

Luisa M Zintgraf[1,3], **Taco S Cohen**[1], **Tameem Adel**[1], **Max Welling**[1,2]
[1]University of Amsterdam, [2]Canadian Institute of Advanced Research, [3]Vrije Universiteit Brussel
{lmzintgraf,tameem.hesham}@gmail.com, {t.s.cohen, m.welling}@uva.nl





Importance map for testing dataset



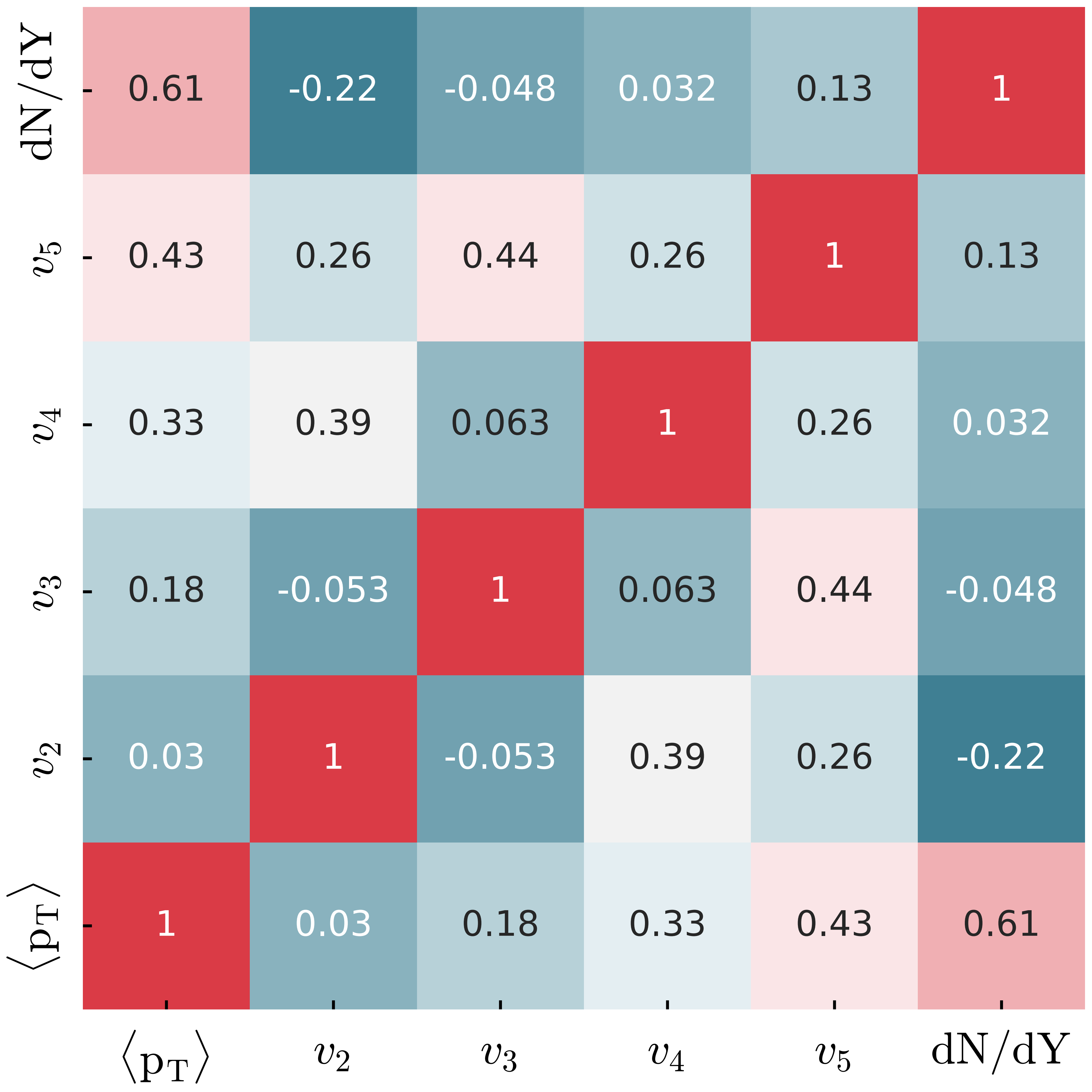
GROUP 1 GROUP 2

- Experimentalists may look for new observables/ correlation functions that are sensitive to EoS, inspired by the importance map given by machine learning. E.g.

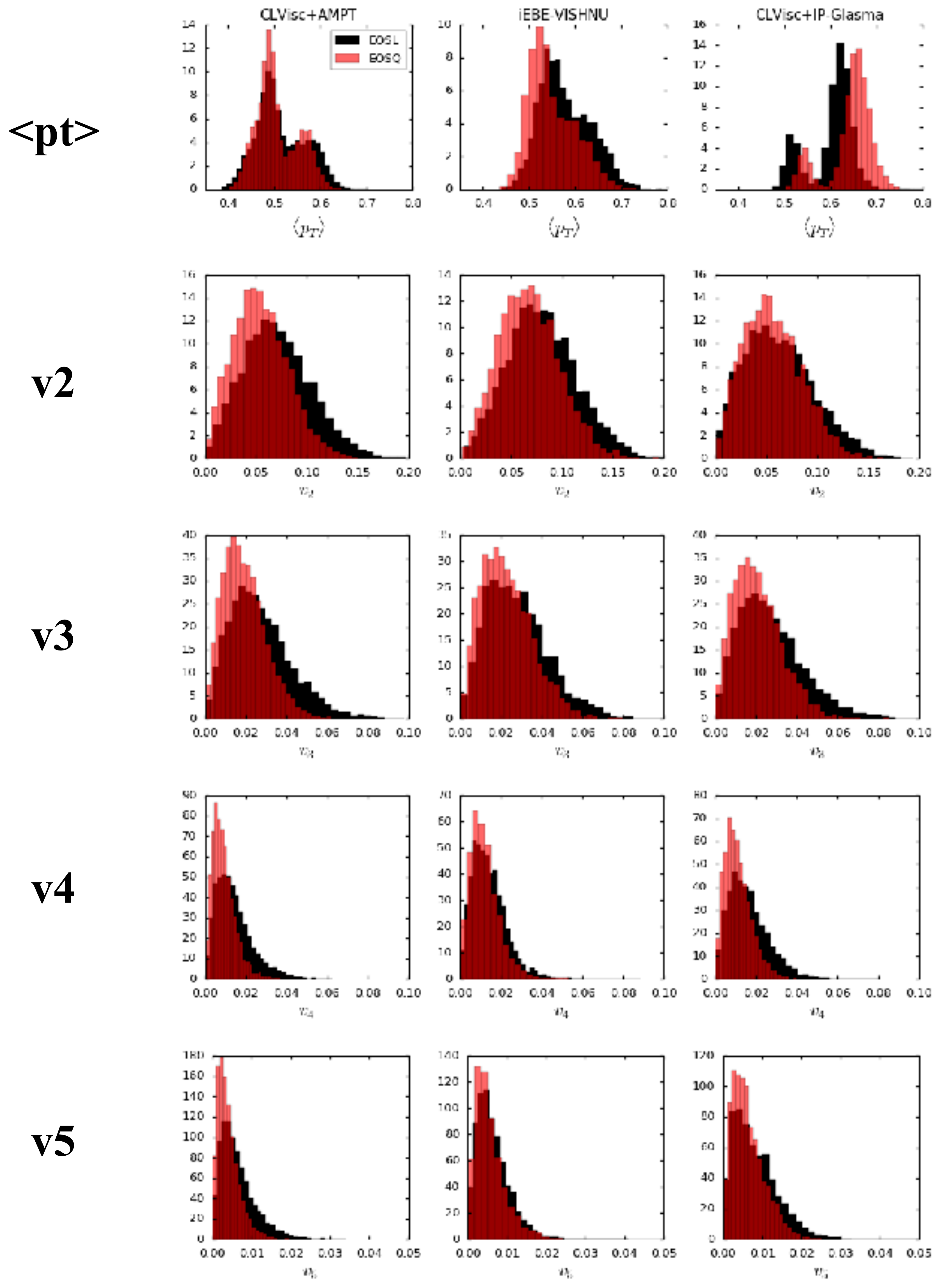
$$C_{12} = < N_A N_B > - < N_A >< N_B >$$
$$N_A = N(p_T = 0.3, \phi = \pm\pi/2)$$
$$N_B = N(p_T = 0.8, \phi = \pi)$$

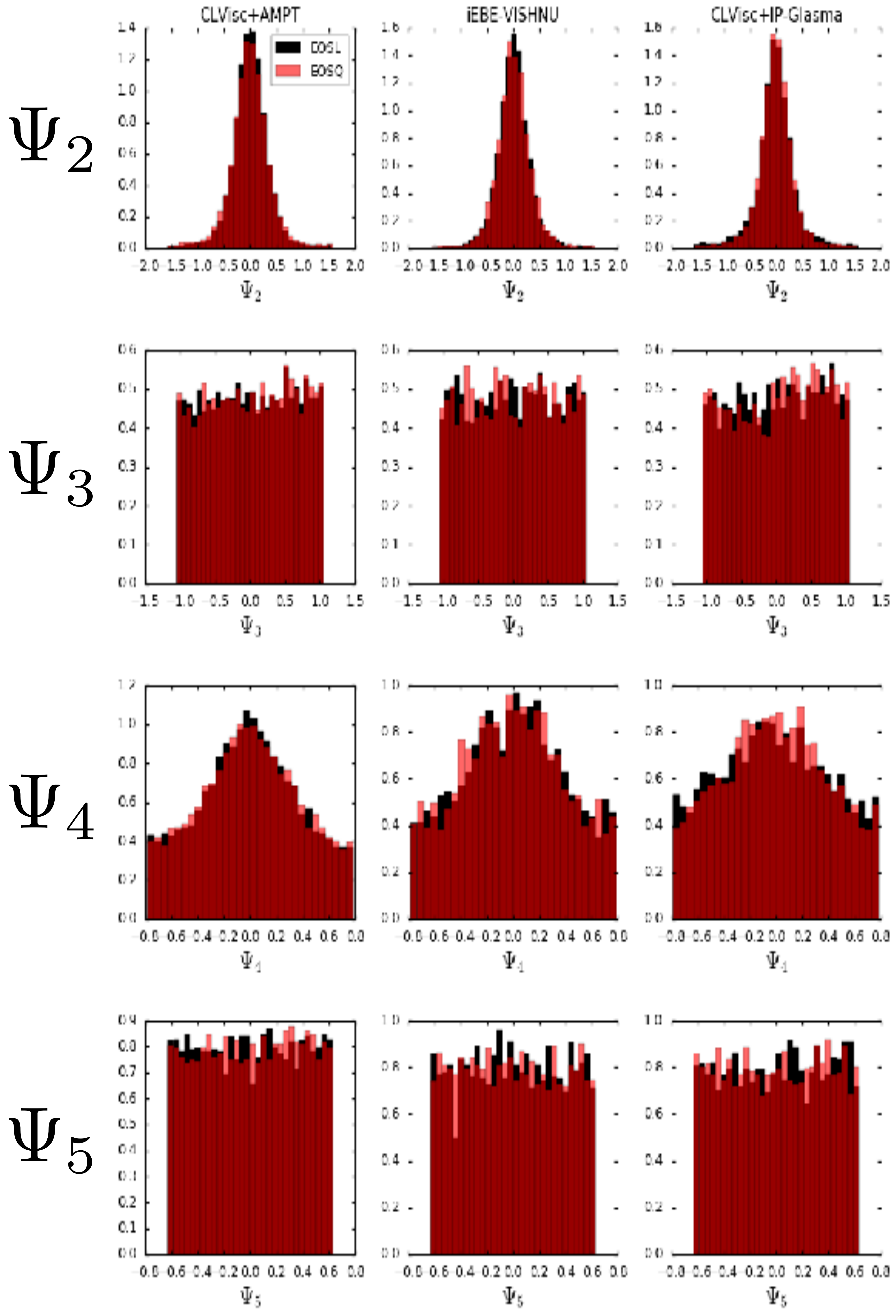
The correlation matrix from the simulated data

- Confirms various correlations, e.g. (v2, v4), (v2, v5), (v3, v5), (<pt>, dN/dY)…
- Reveals strong correlation between <pt> and v5! (never been found before).
- But those traditional observables and correlations can not classify the 2 different EoS.

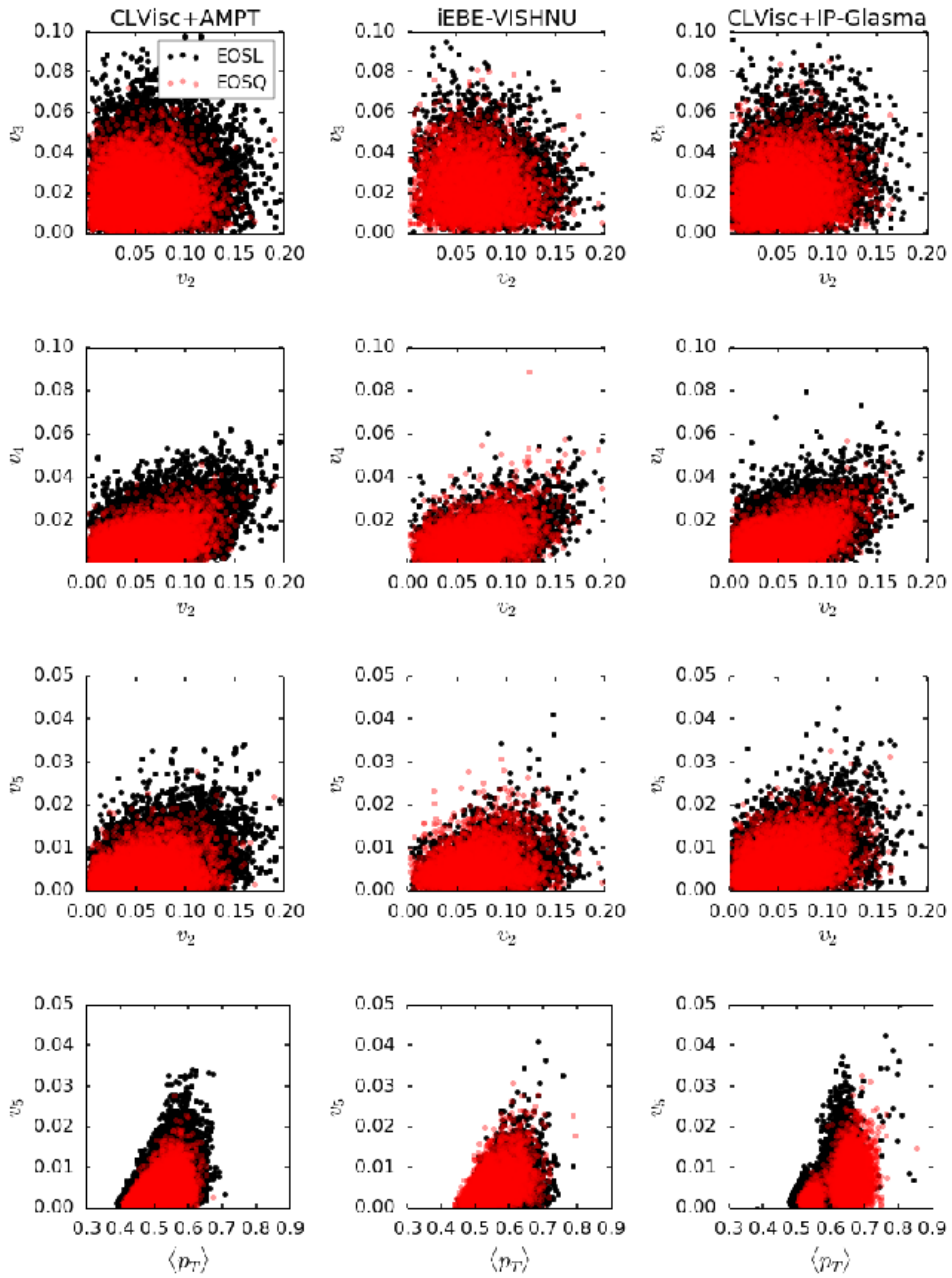


EBE distribution of pre-defined observables (black-EOSL, red-EOSQ)





Correlations between several observables (black-EOSL, red-EOSQ)



- The event-by-event distributions of the traditional observables fail to distinguish two different EoS.

- The correlation between (v2, v3), (v2, v4), (v2, v5) and (<pt>, v5) fail to distinguish two different EoS.

Traditional Machine Learning vs. deep neural network

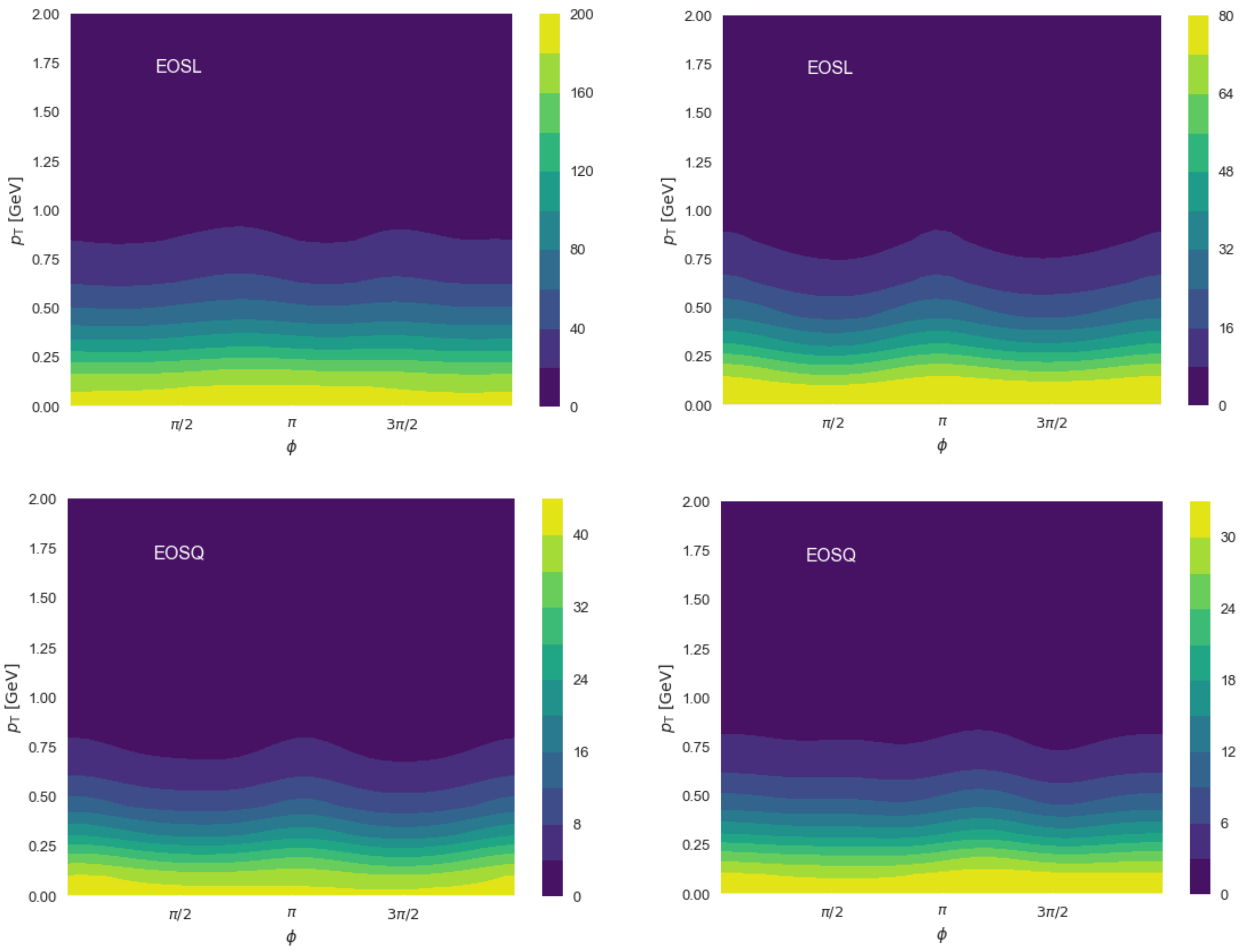
- **Training and testing data**: 15x48 components raw spectra or 85 pre-defined observables or principle components in raw spectra from PCA method
- **Machine learning Tools**:
    - Gaussian Naive Bayes Classifier
    - Support Vector Machine Classifier
    - Decision Tree Classifier
    - Random Forest and Gradient Boosting Trees

Traditional Machine Learning vs. deep neural network

Prediction Accuracy	GROUP1	GROUP2
obs + Gaussian Naive Bayes	46.2%	47.6%
obs + Decision Tree	57.5%	64.9%
obs + Random Forest	62.5%	69.8%
obs + Gradient Boosting Trees	66.9%	81.9%
obs + linear SVC	75.8%	84.6%
obs + SVC rbf kernel	60.9%	56.7%
raw + linear SVC	65.2%	84.3%
pca + linear SVC	46.4%	47.7%

our approach (DCNN) **~95%**

SOME RANDOMLY SELECTED PARTICLE SPECTRA



Gaussian Naive Bayes Classifier

Bayes Classifier:

$$P(c|\mathbf{x}) = \frac{P(c)P(\mathbf{x}|c)}{P(\mathbf{x})}$$

Naive Bayes Classifier:

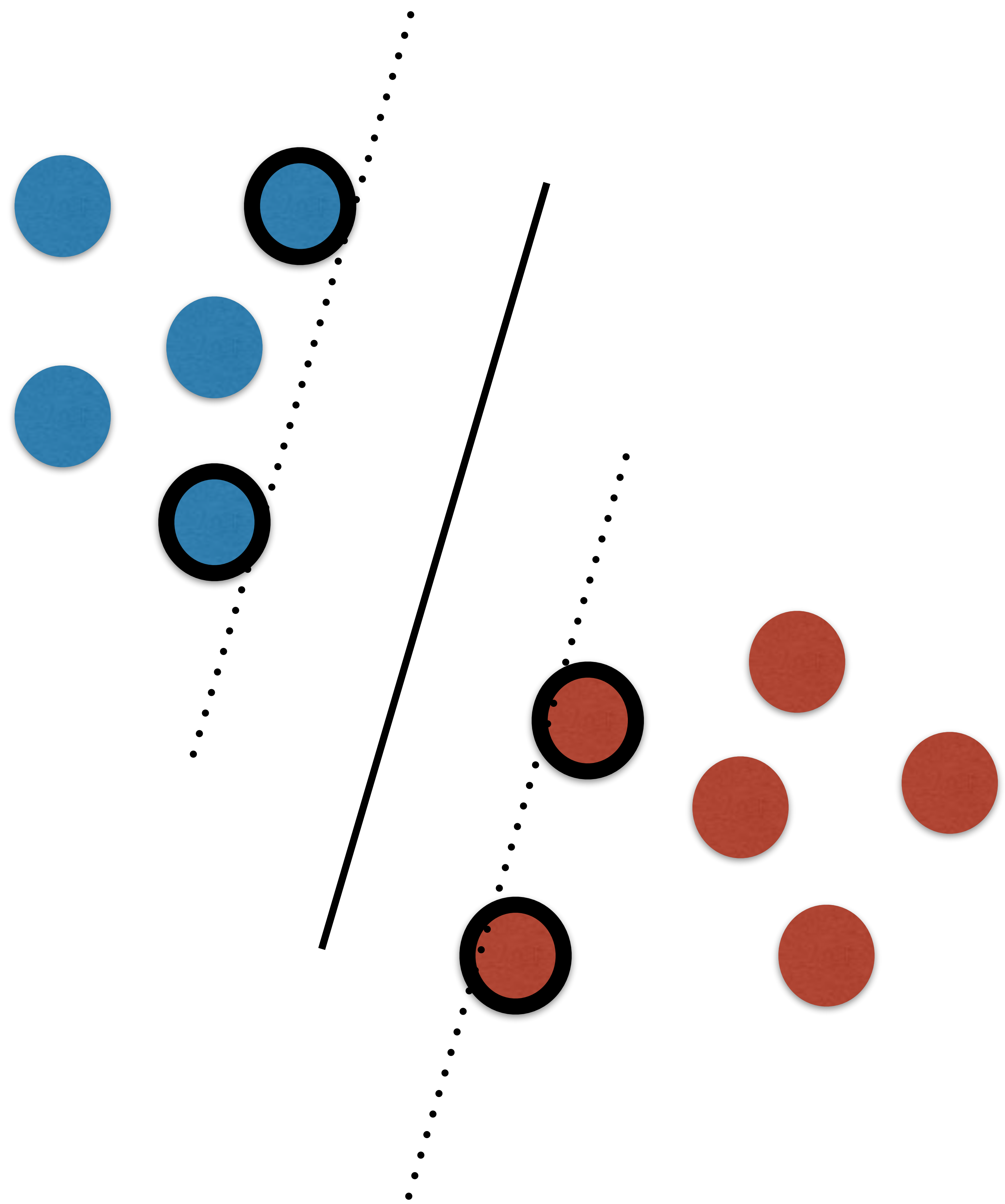
$$P(c|\mathbf{x}) = \frac{P(c)}{P(\mathbf{x})}\sum_{i=1}^{d} P(x_i|c)$$

Gaussian Naive Bayes Classifier:

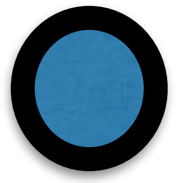
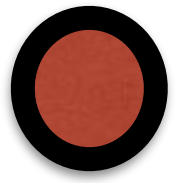
$$p(x_i|c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left[-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}\right]$$

- NB: Assume each feature affects classification independently
- GNB: For continuous features, using probability density distribution.

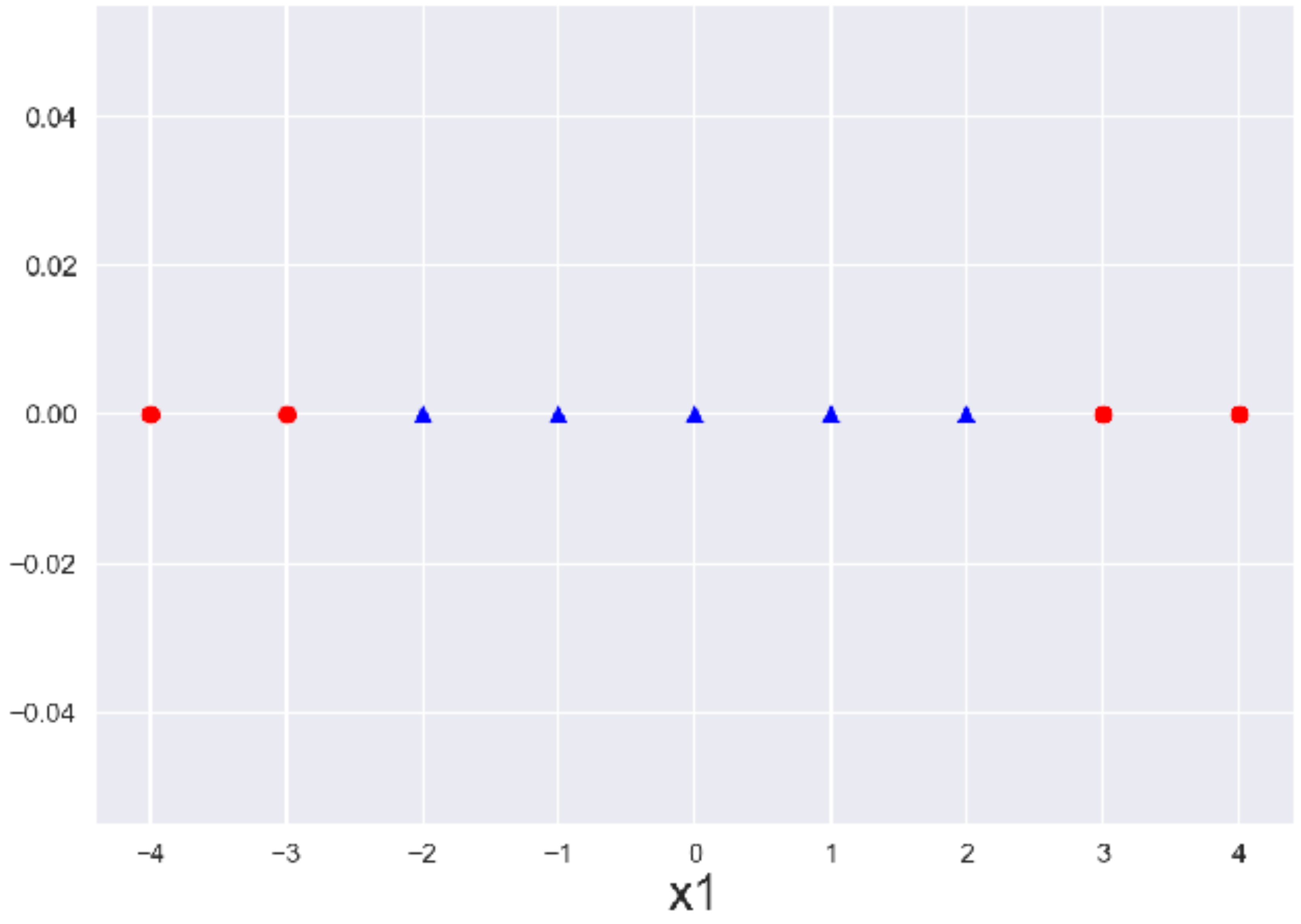
Linear Support Vector Machine Classifier

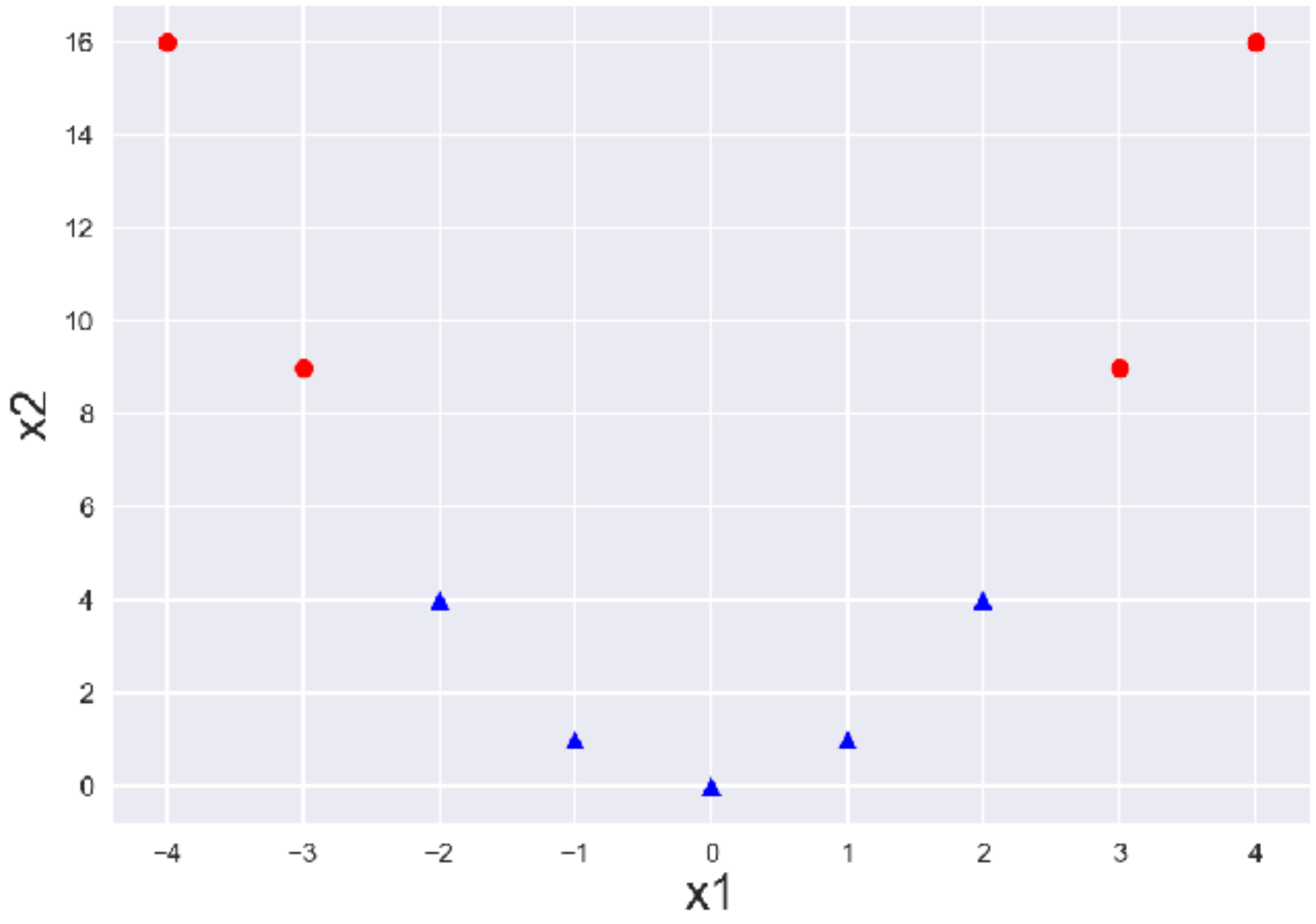


- SVM: Looking for the widest street that can separate 2 classes.
- Each data point is a n-dimensional vector
- The decision boundary is one n-1 dimensional hyper surface.

 **and**  **are support vectors for classification.**

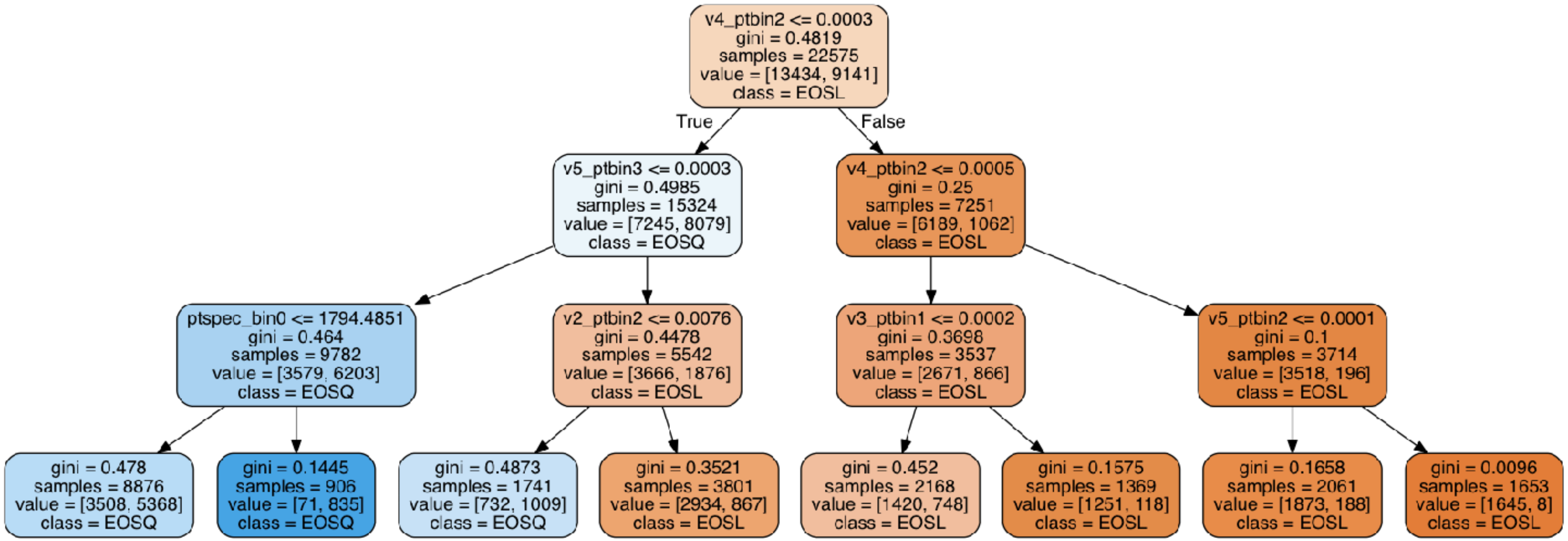
Support Vector Machine with non-linear kernels





- Left: dataset with one feature x1, not linearly separable
- Right: define x2 = x1 * x1, now linearly separable
- kernels are easier ways to introduce this non-linearity

Decision Tree



- Poor decision tree (accuracy 57% for Group1 and 64.9% for Group2) — features are not robust.

- For this tree, the best result is determined by the right bottom block (Gini impurity = 0.0096).

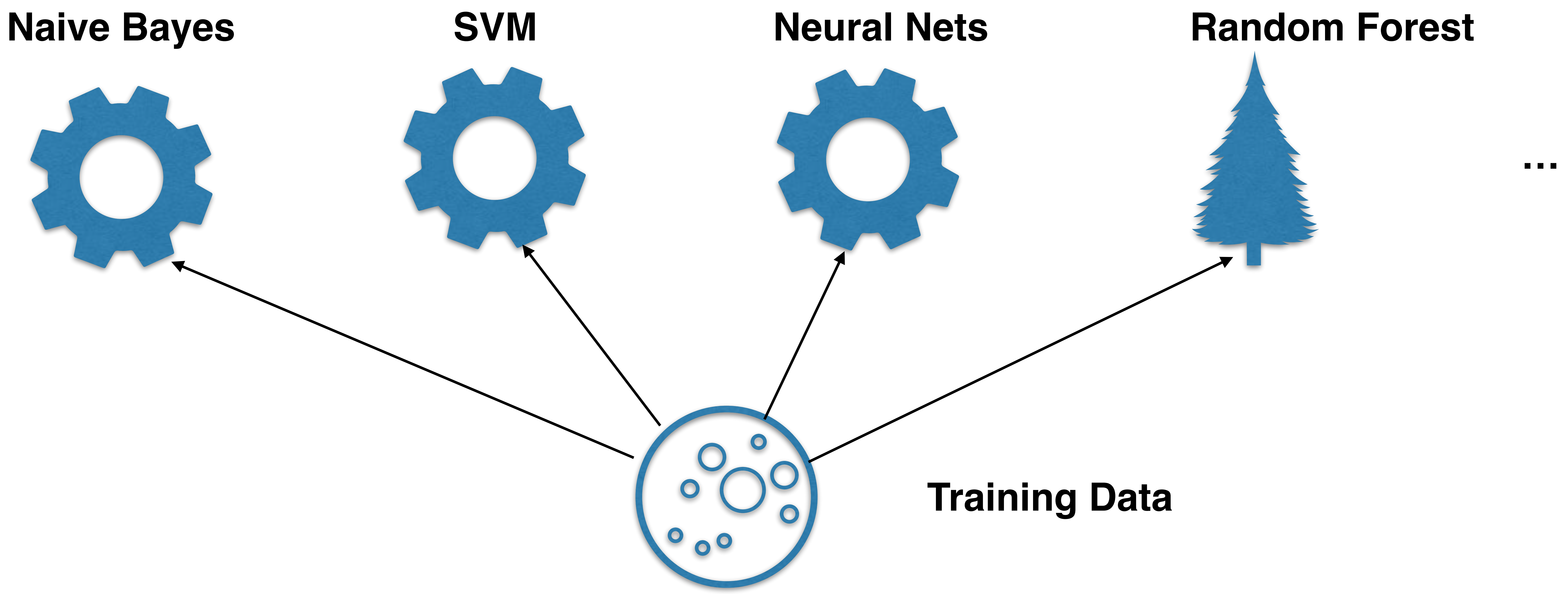
Ensemble Methods (1) Bagging and Stacking

三个臭皮匠，抵过诸葛亮

- Random Forest: each decision tree is a weak classifier, many diverse decision trees + majority voting = strong classifier whose accuracy is higher than the best classifier in the ensemble.

- Bagging: many different classifiers + majority voting (少数服从多数)

- Stacking: many different classifiers + learning to vote (真理可能掌握在少数人手中）



Ensemble Methods (2) Boosting

知错能改，善莫大焉

- Boosting: sequentially improve the classifier by paying more attention to misclassified samples

- Example: AdaBoost, XGBoost (many winners of Kaggle data science competing)