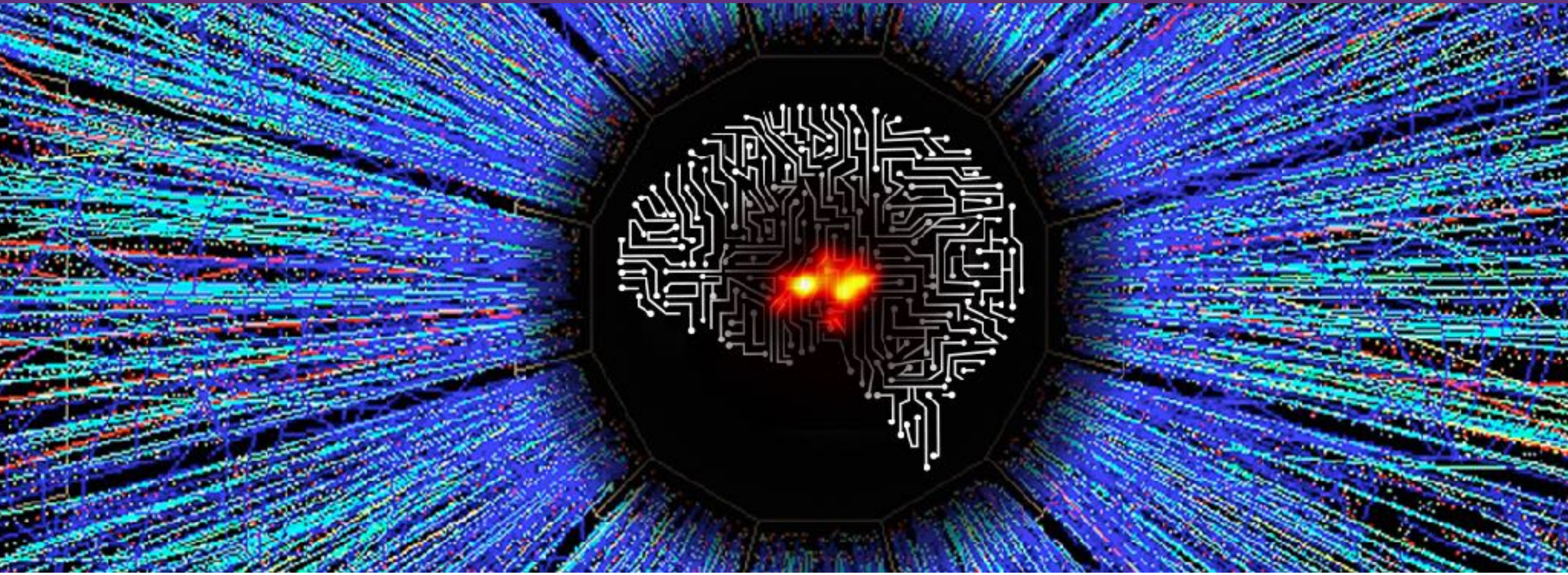
The application of deep learning to event-by-event simulations of relativistic hydrodynamics



LongGang Pang
UC Berkeley & Lawrence Berkeley National Laboratory

LG. Pang, K.Zhou, N.Su, H.Petersen, H. Stoecker, XN. Wang. Nature Communications 9 (2018) no.1, 210

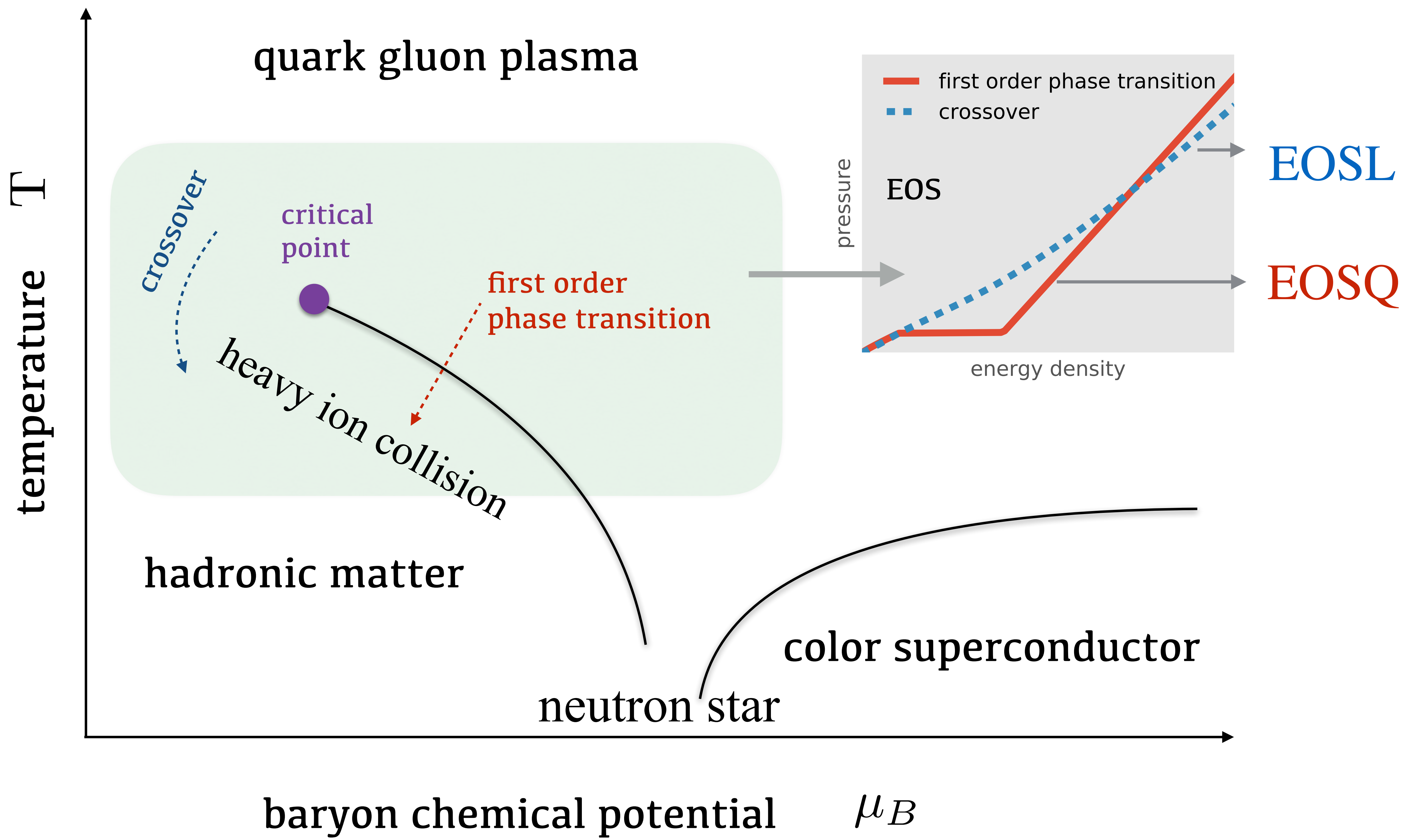
Berkeley UNIVERSITY OF CALIFORNIA

BERKELEY LAB

JETSCAPE

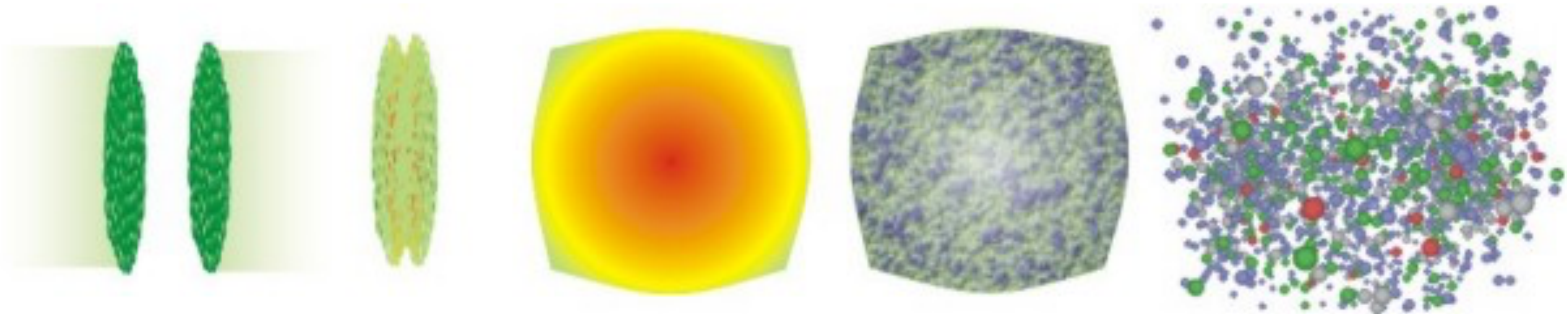
CIPANP
2018
Palm Springs, CA

Conjectured QCD phase diagram



Hot QCD matter (quark gluon plasma) in heavy ion collisions

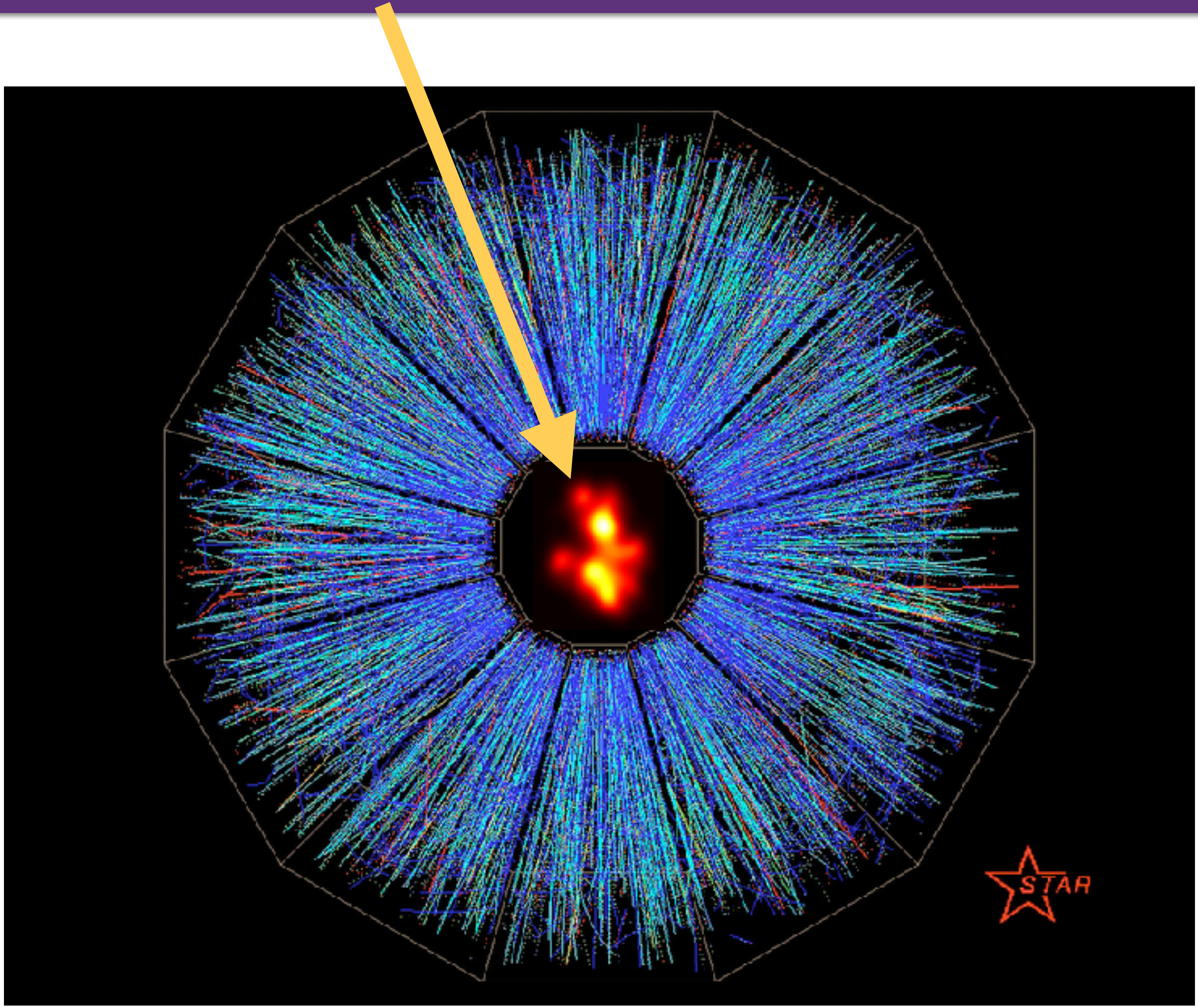
Fig originally from Steffen A. Bass



Given the EoS, we can predict the final state particle distribution.

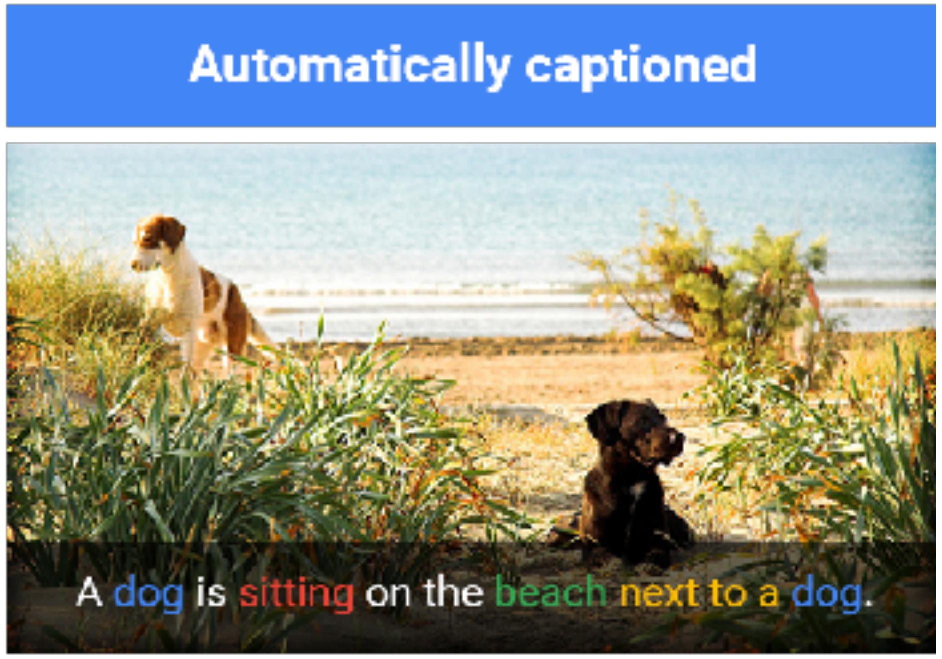
However, the final distribution is also affected by many other parameters and uncertainties in the initial state, the evolution and the particlization.

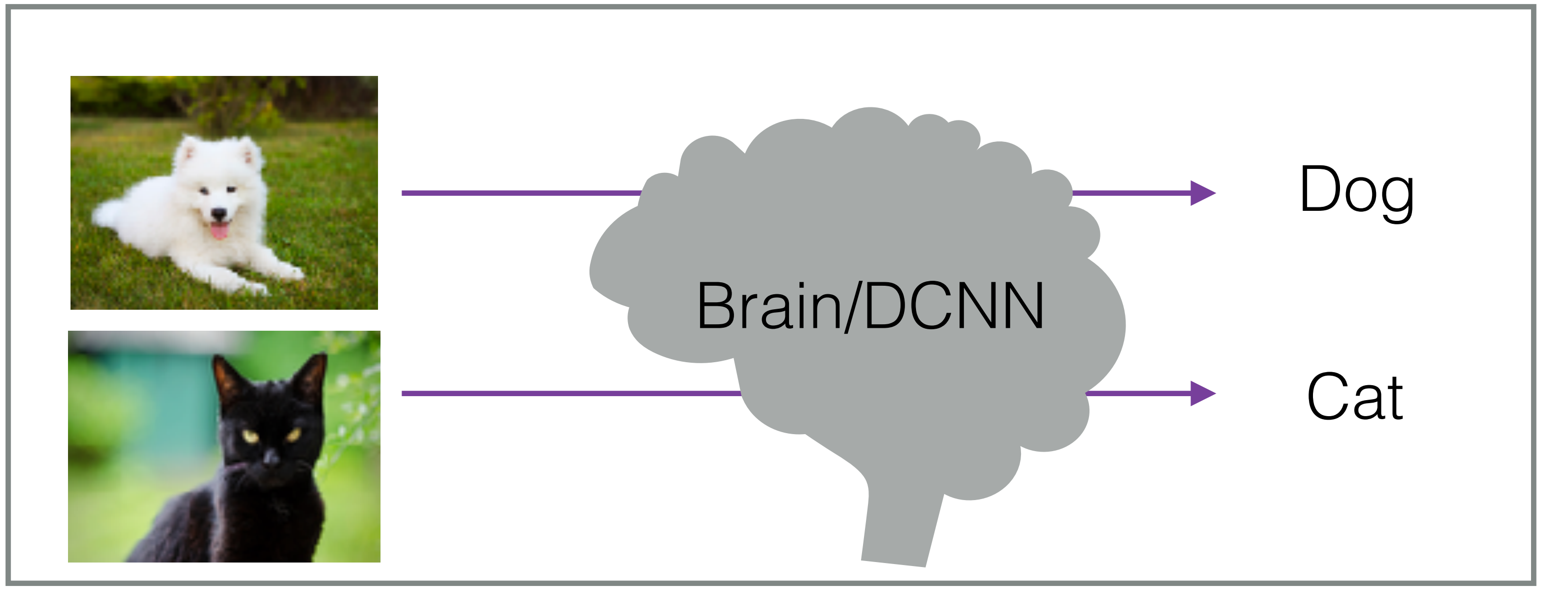
How to get EoS given the final state particles?



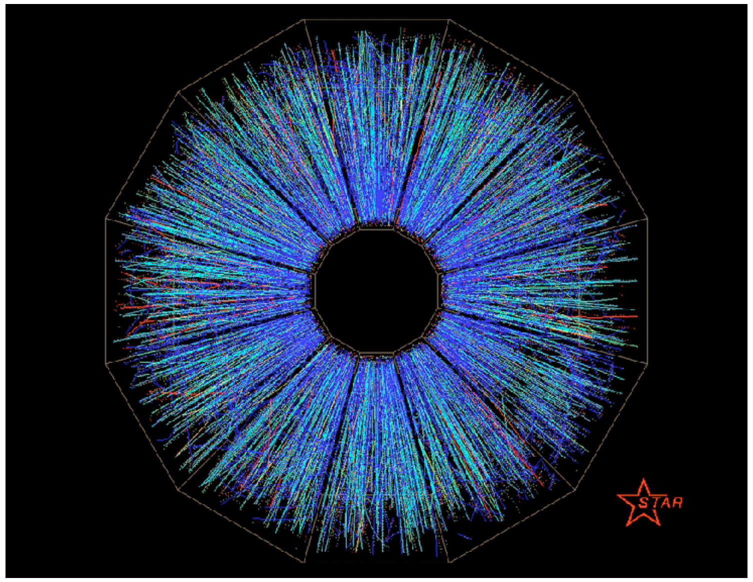
Deep neural network can make captions for images

Google, DeepMind

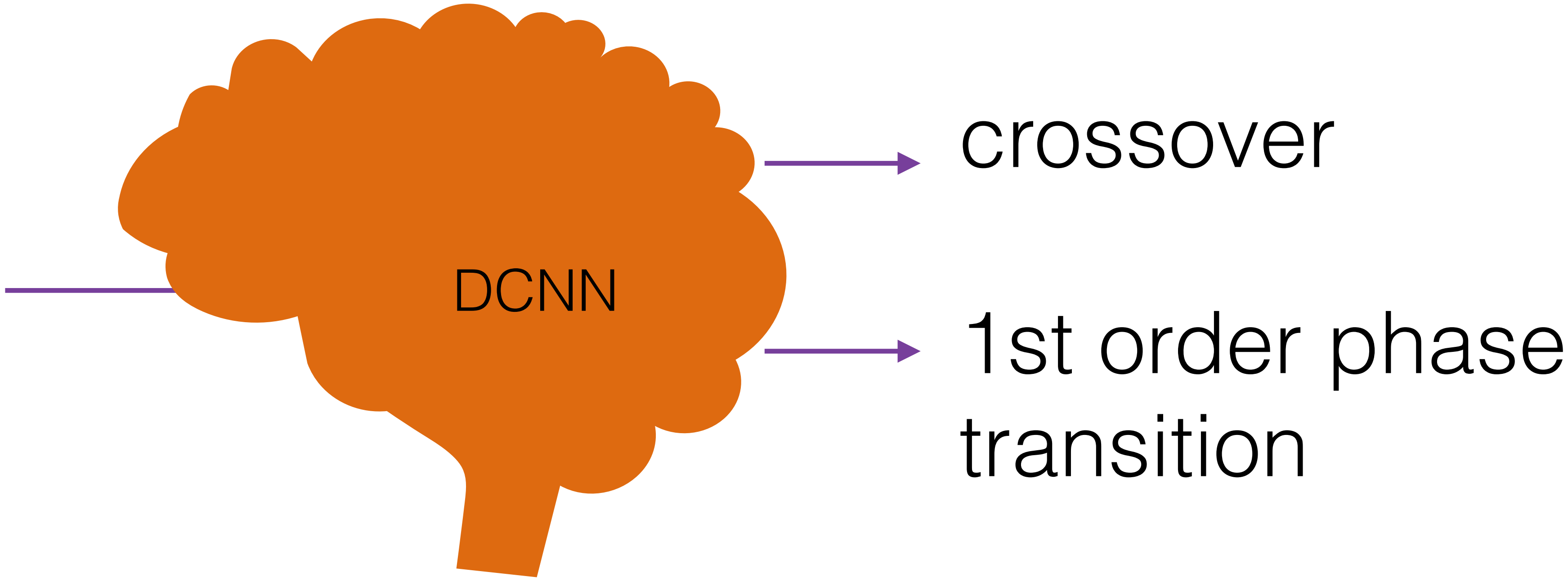




Classifying two phase transition regions



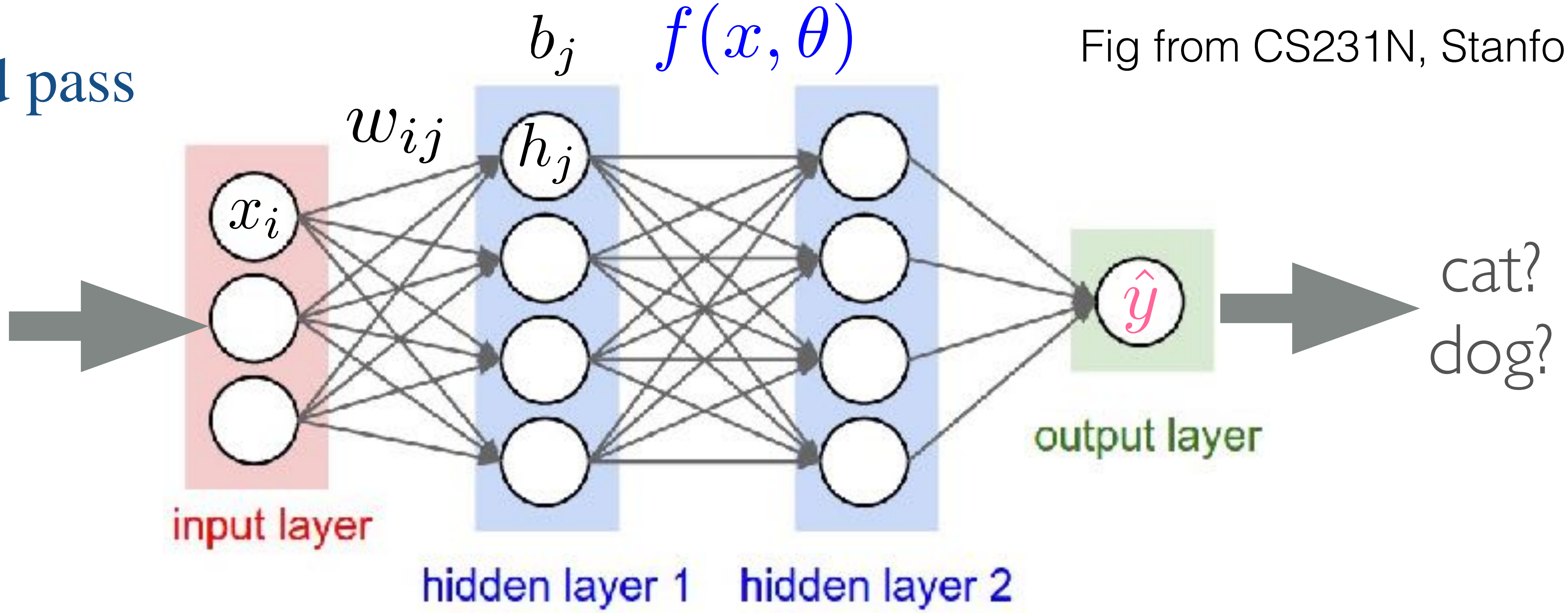
$\rho(p_T, \Phi)$



"hello world" example of deep neural network

Forward pass

Fig from CS231N, Stanford



Linear operation

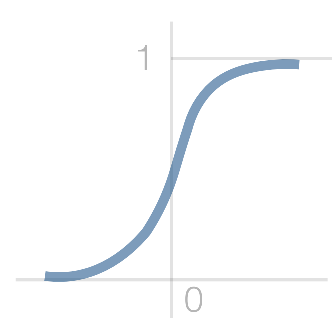
$$z_j = \sum_{i=1}^{N} x_i w_{ij} + b_j$$

scaling, rotating, boosting, changing dimensions

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

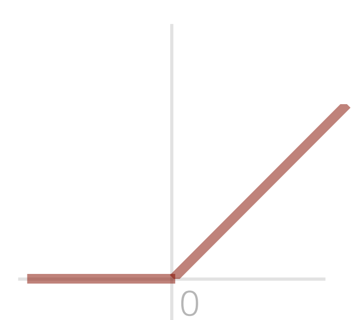
(a) Sigmoid

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



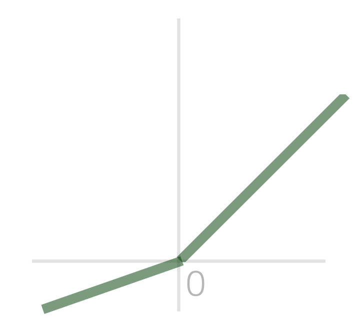
(b) ReLU

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

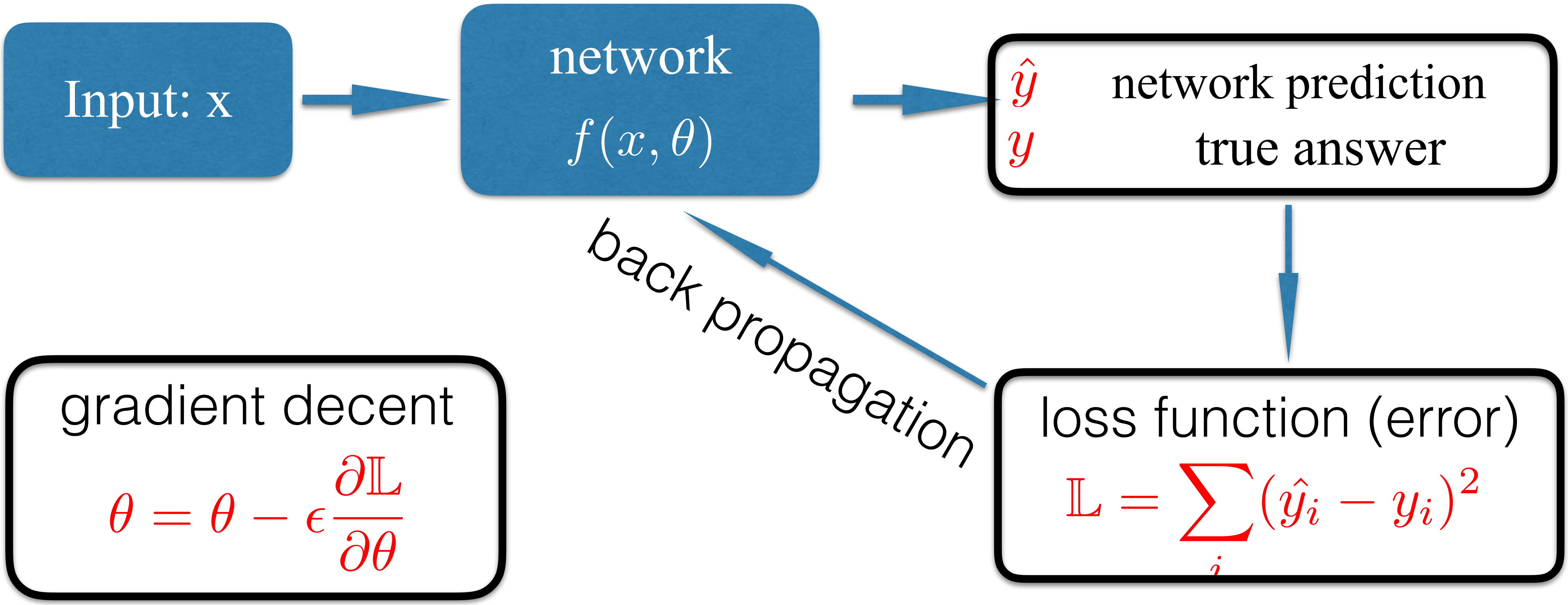


(c) PReLU

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

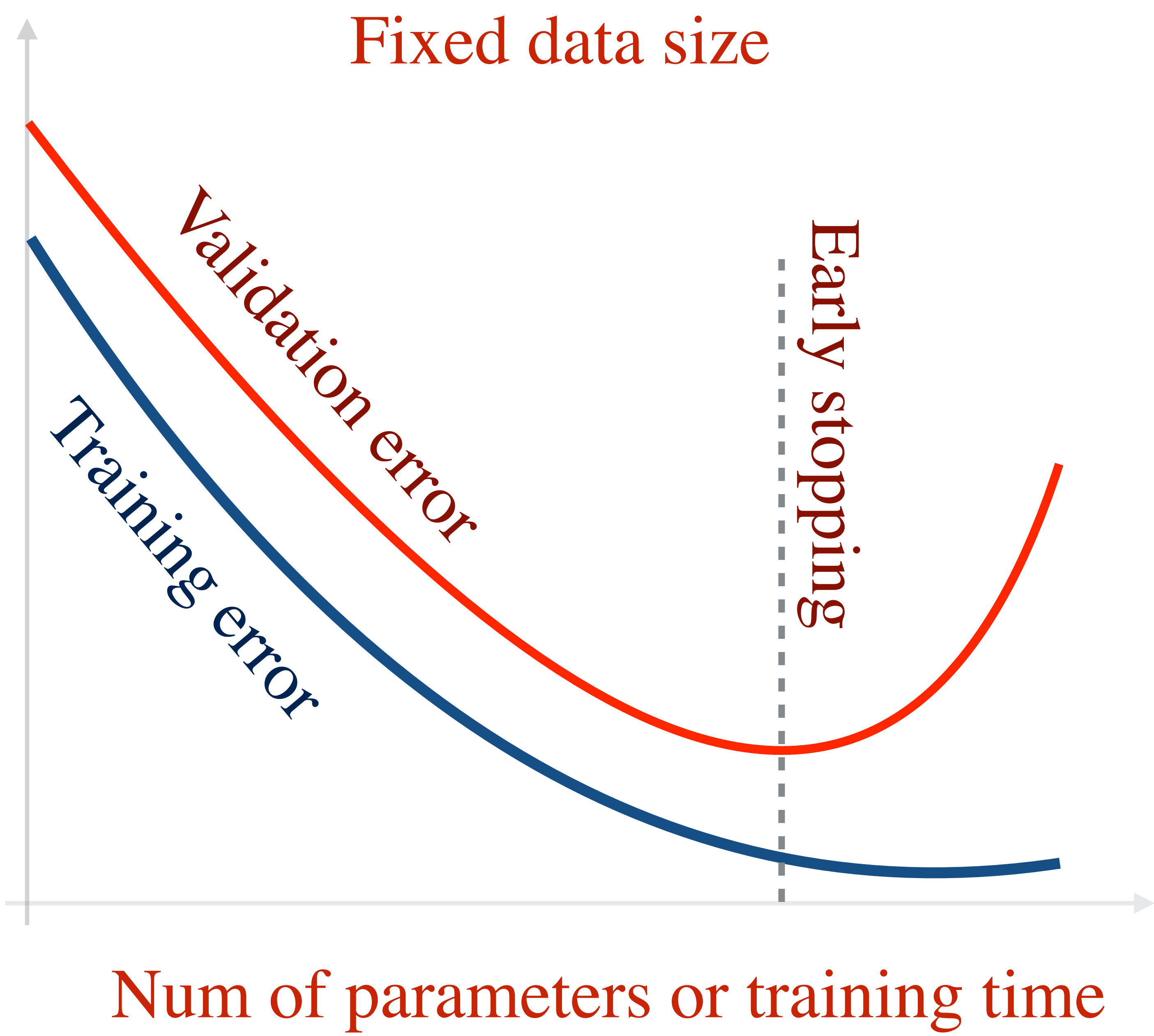


Back propagation and gradient decent



Human intelligence/artificial neural network can reduce fitting error by updating model parameters through back propagation and gradient decent.

Overfitting problem in fully connected network



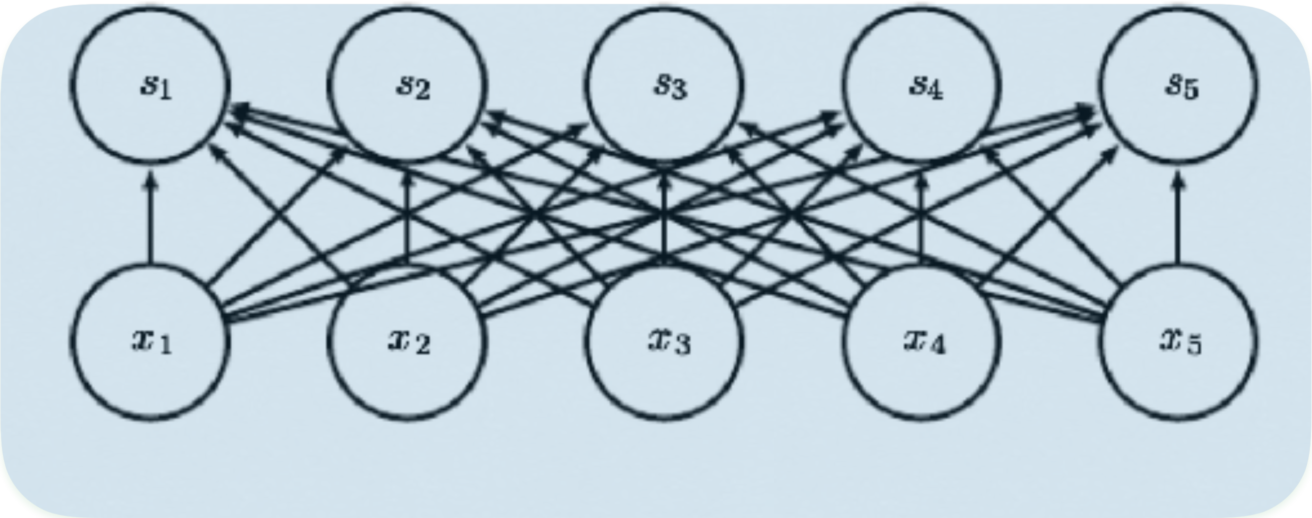
- Training error: prediction error rates on training data
- Validation error: prediction error rates on new data

Ways to reduce overfitting

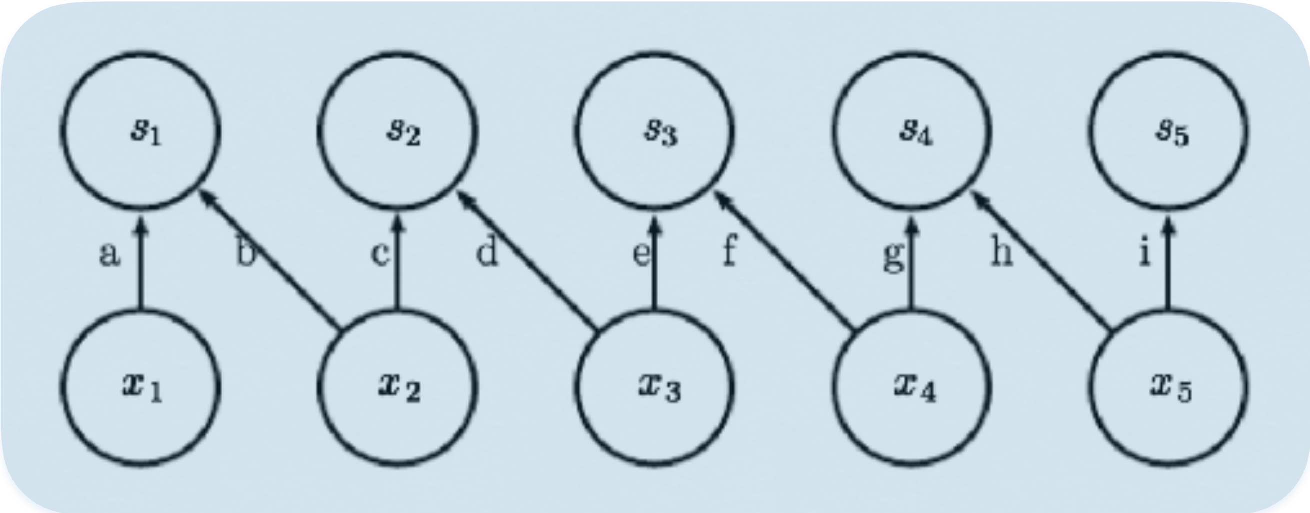
1. Early stopping
2. Increase training dataset by
   a. preparing big amount of data.
   b. data augmentation (crop, scale, rotate, flip …).
3. Reduce number of parameters
   a. Dropout: randomly discard neurons.
   b. Drop connection: randomly discard connections.
   c. CNN: locally connected to a small 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

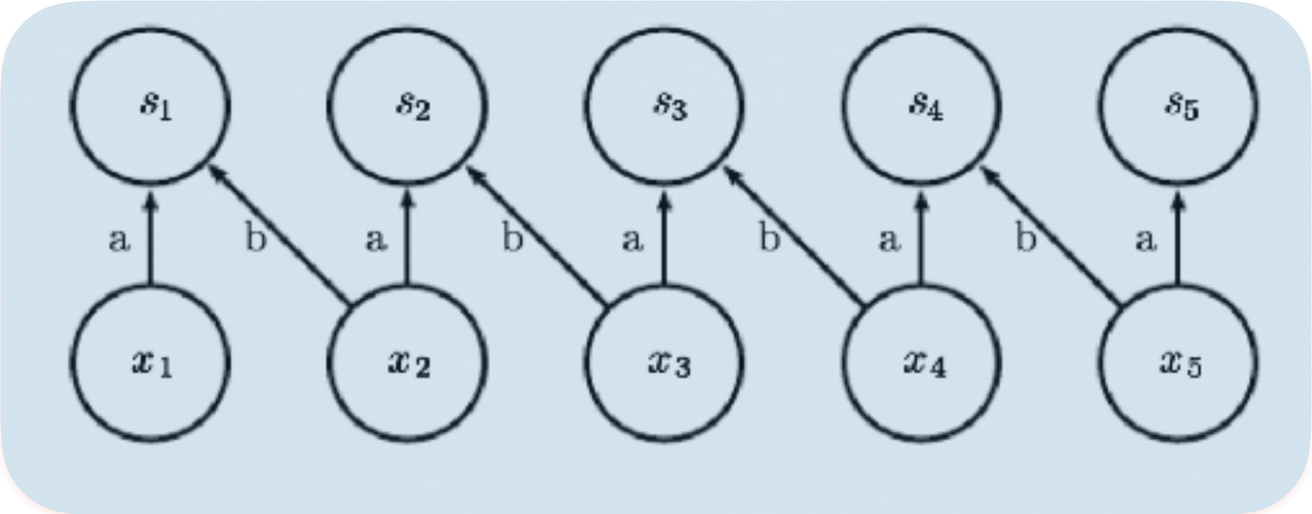
Fully Connected



Locally Connected



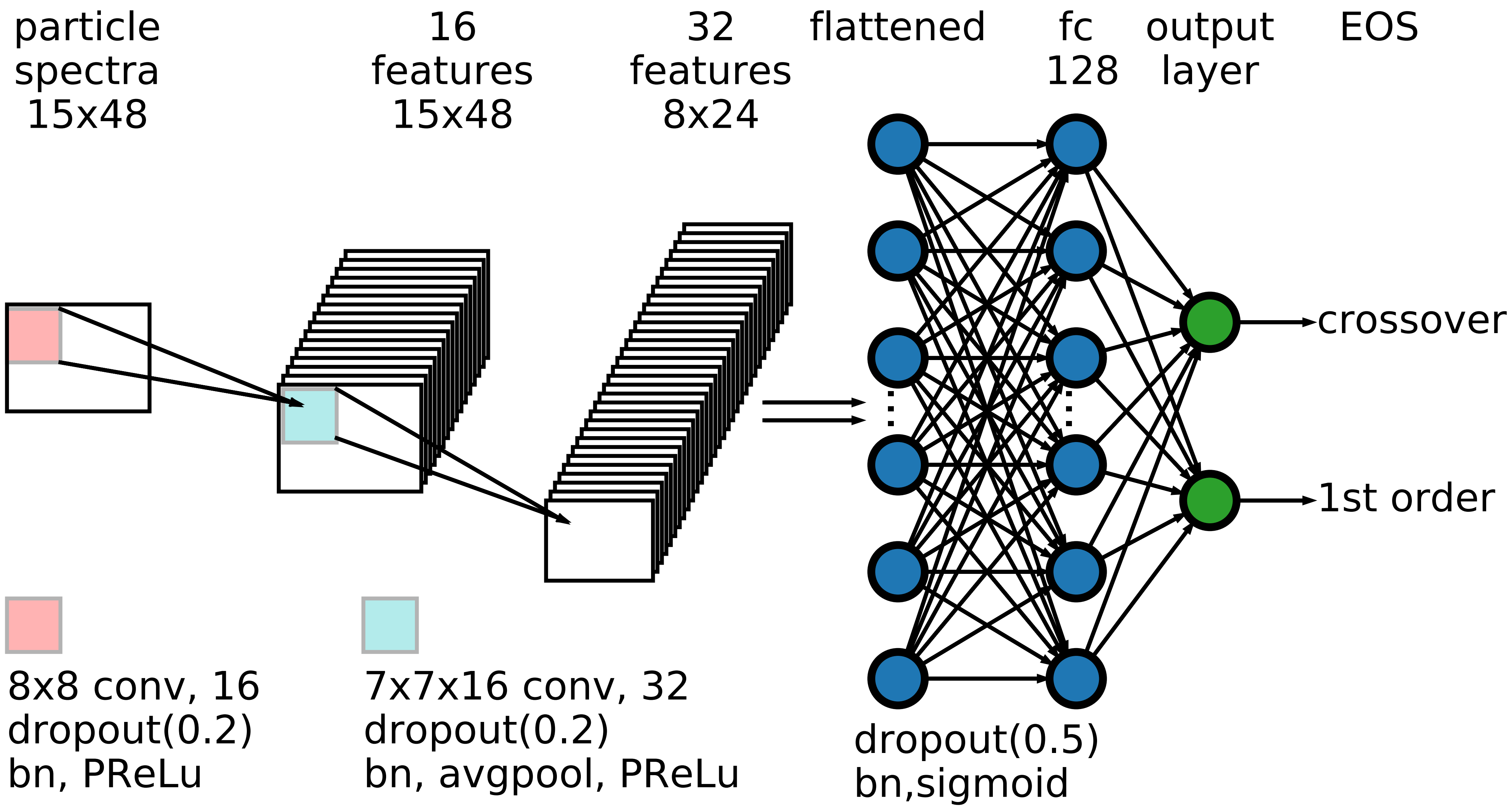
Locally Connected
+
Share Weights



Convolution ✓

From "Deep Learning" Book.

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

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.

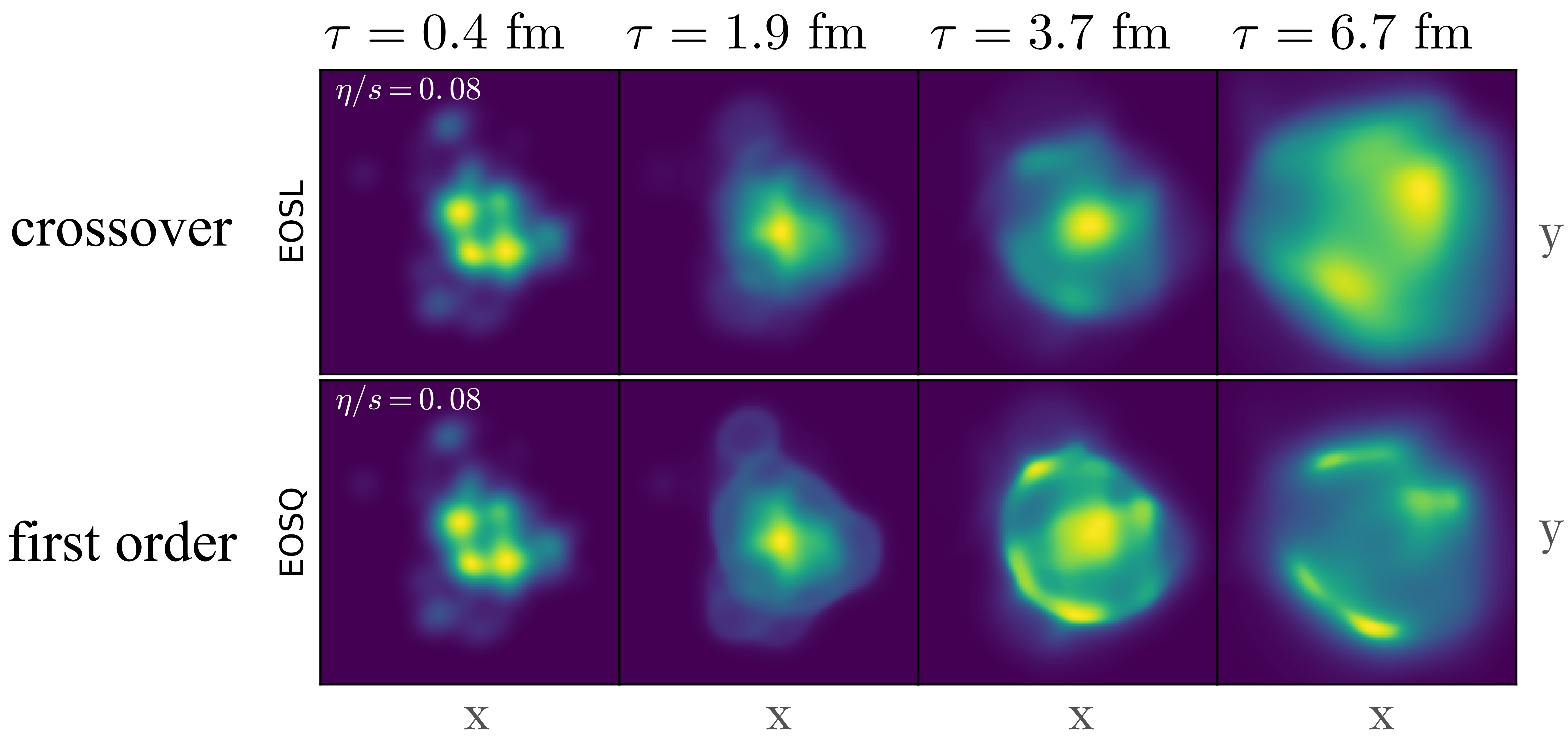
Big data from the relativistic hydrodynamics on GPU

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

$$\text{where } T^{\mu\nu} = (\epsilon + P)u^\mu u^\nu - P g^{\mu\nu} + \pi^{\mu\nu}$$

- CLVisc
  - A (3+1)D viscous hydrodynamic program to simulate high energy heavy ion collisions.
  - Parallelized on GPU using OpenCL ~ 60 times speed up.
  - git clone https://gitlab.com/snowhitiger/PyVisc.git
  - LG.Pang, H.Petersen, XN.Wang arXiv:1802.04449

The phase structure is encoded in the evolution history



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

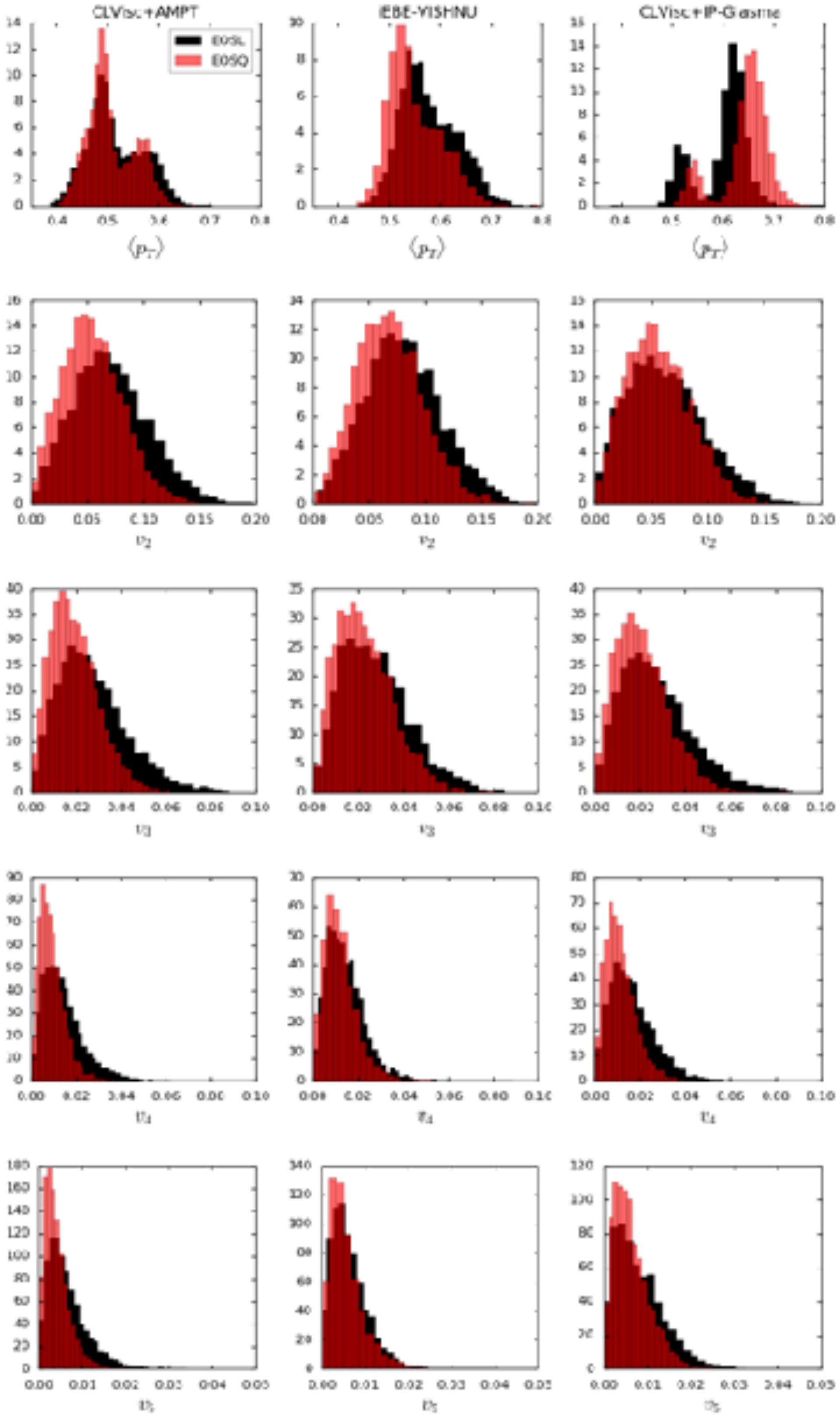
<pt>

v2

v3

v4

v5

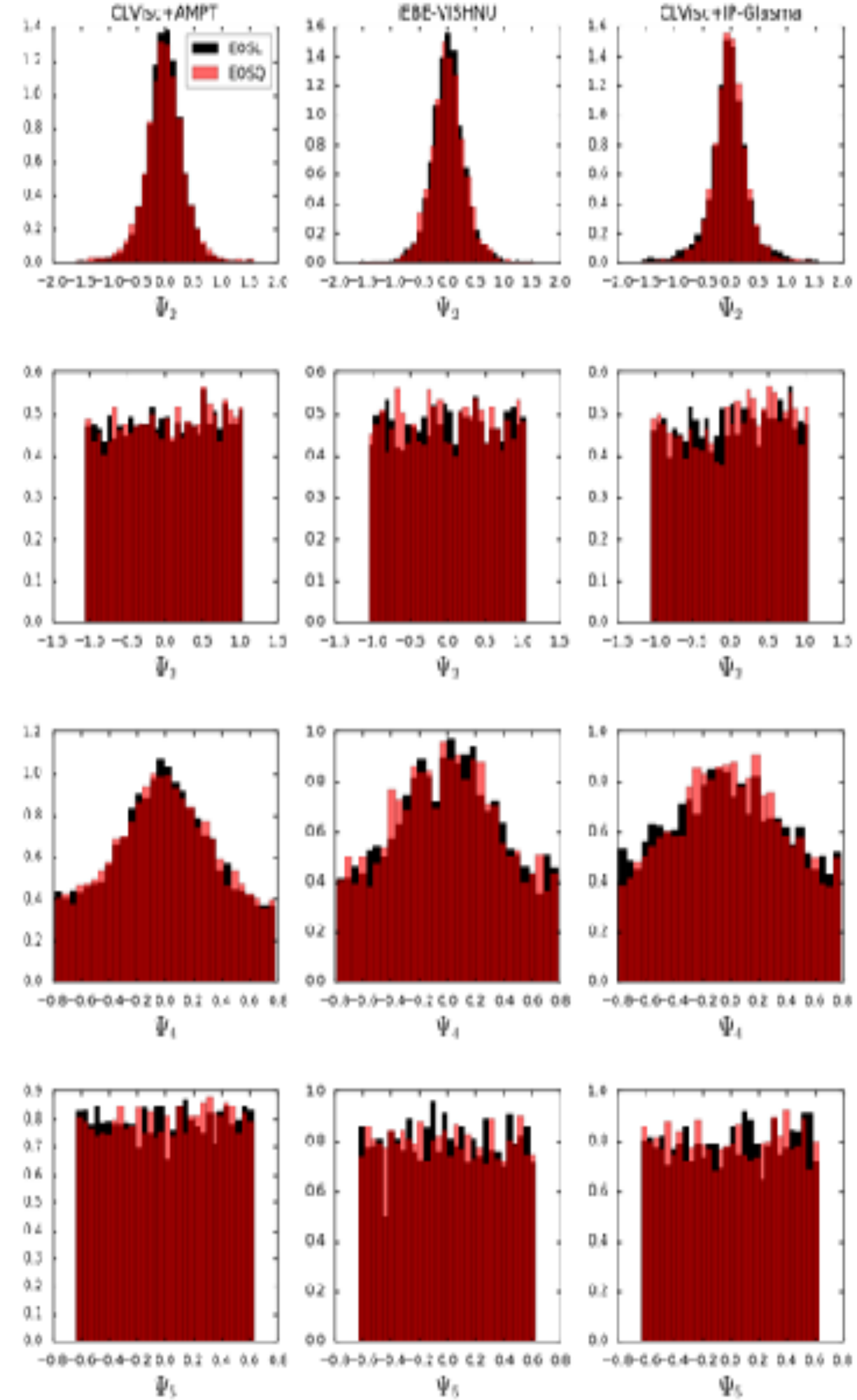


Ψ_2

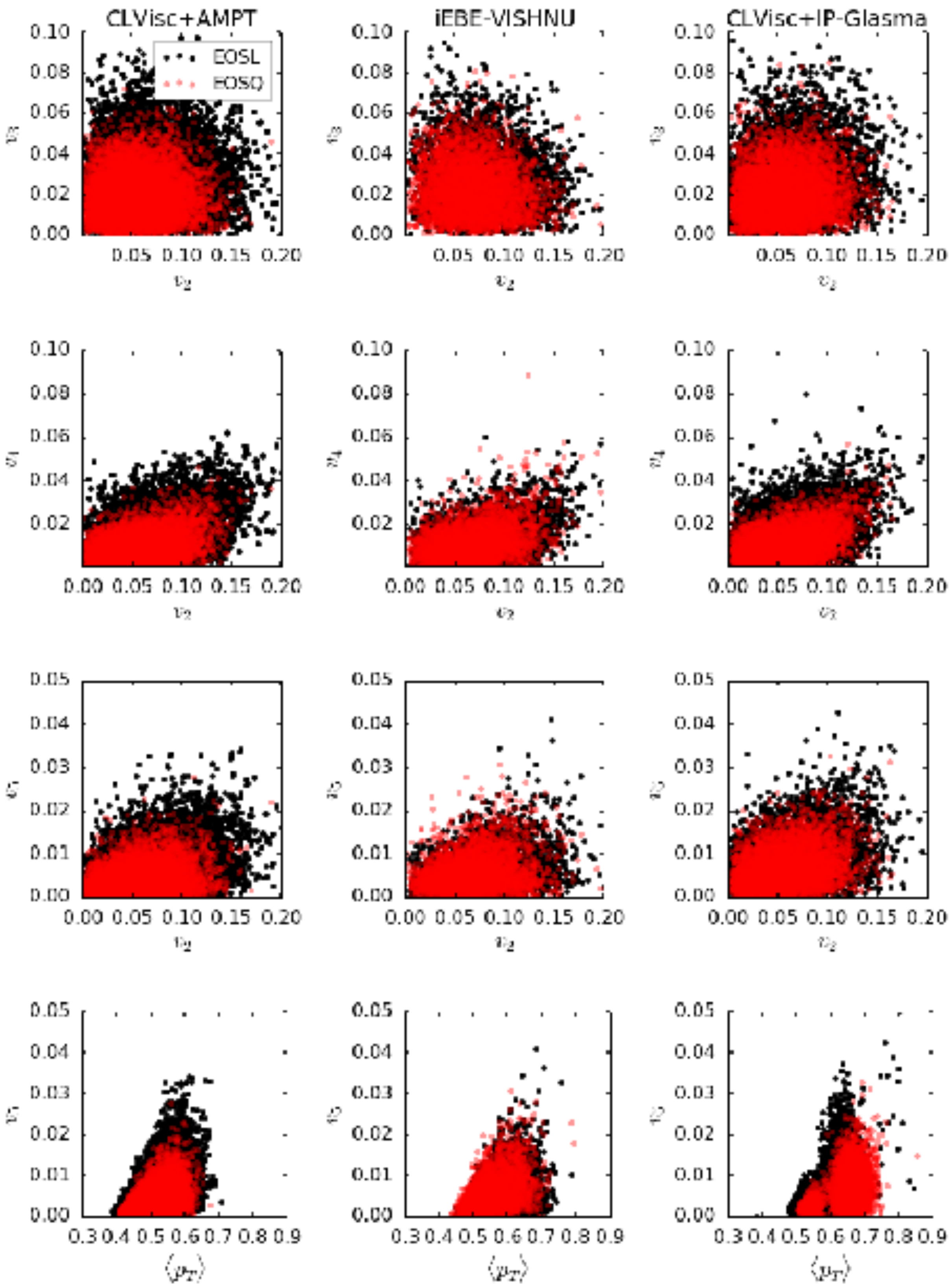
Ψ_3

Ψ_4

Ψ_5



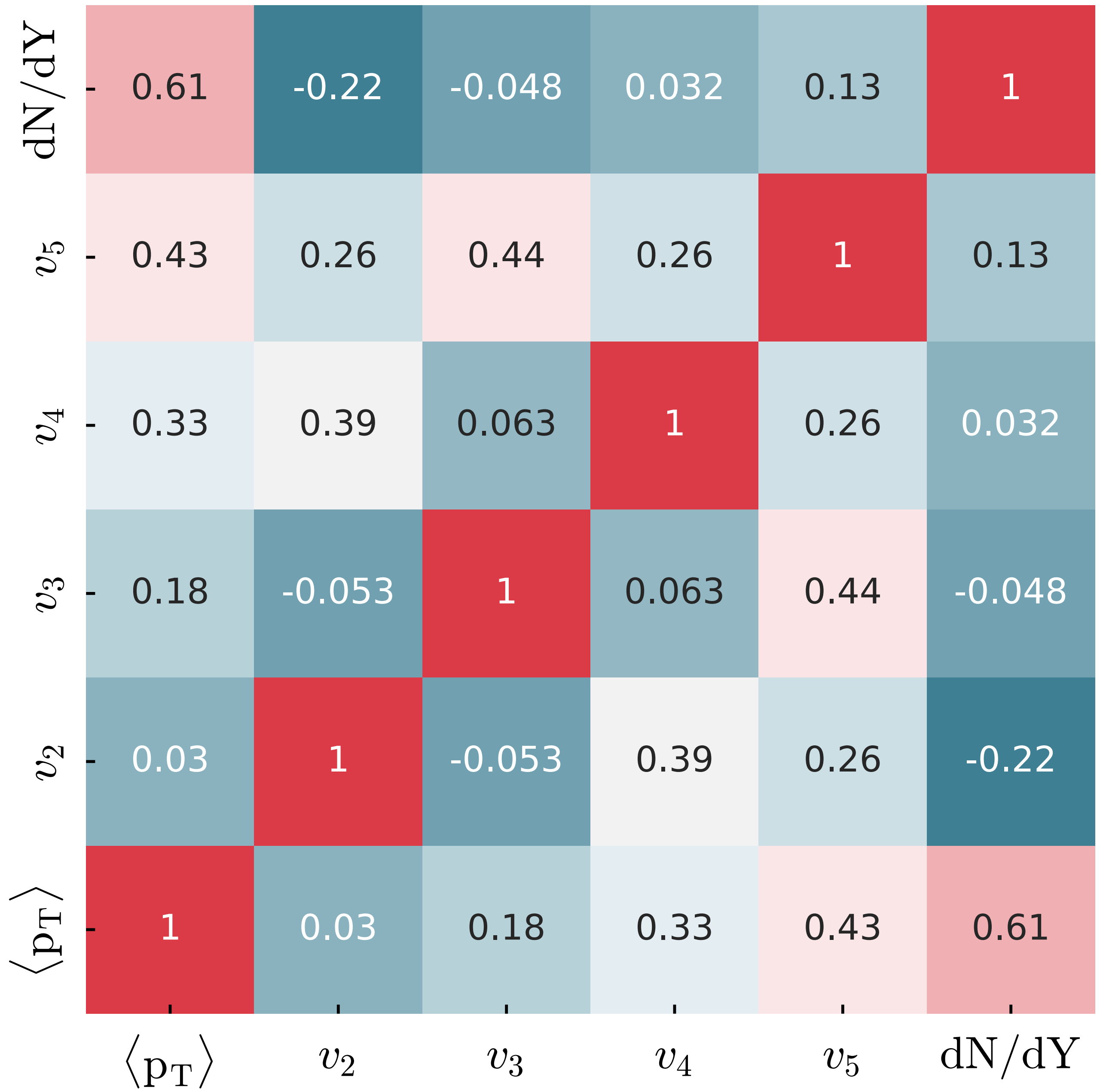
Correlations between several observables (black-EOSL, red-EOSQ)



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

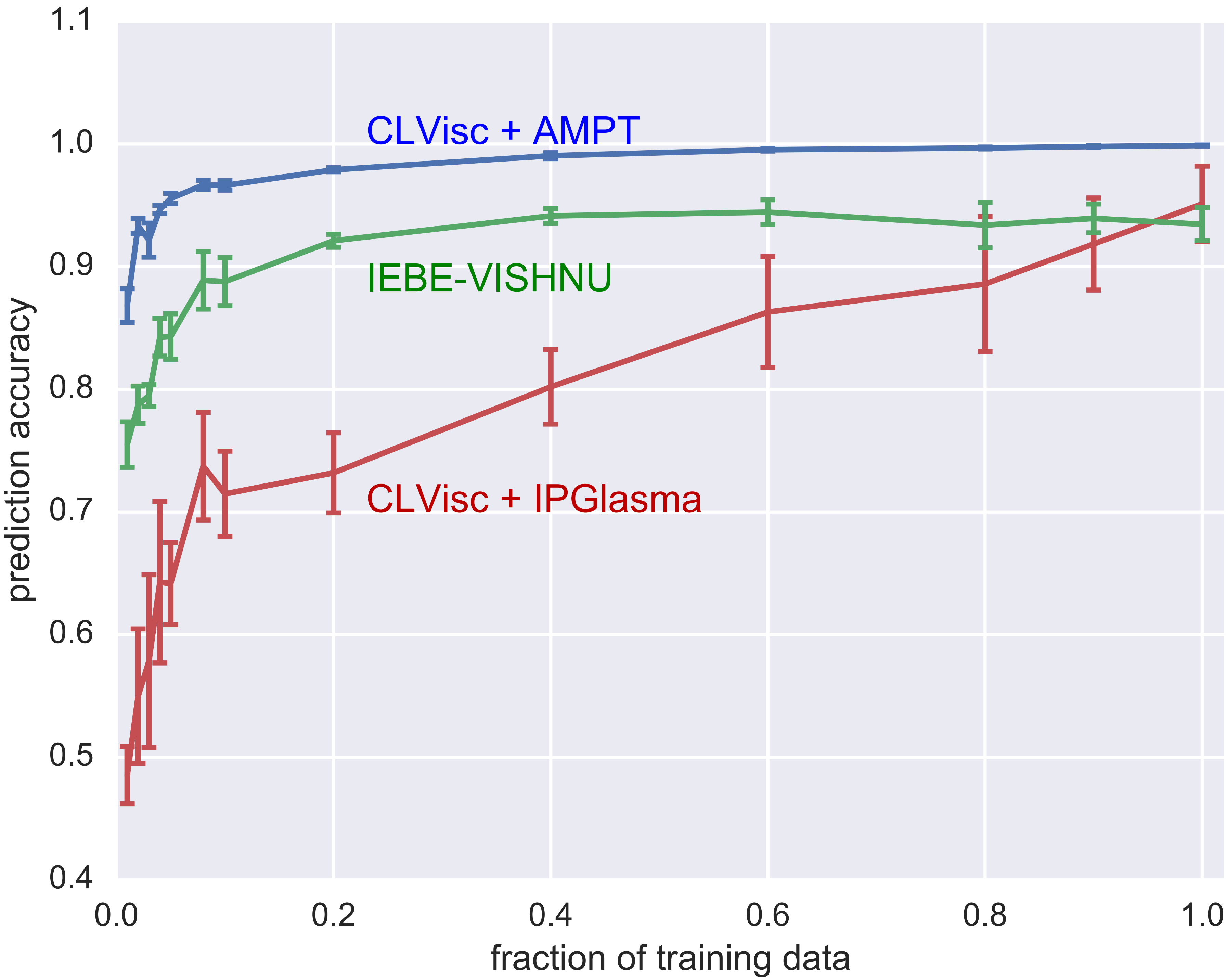
- The correlation between (v2, v3), (v2, v4), (v2, v5) and (<pt>, v5) fail to distinguish two different EoS.

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.

Results: classification accuracies

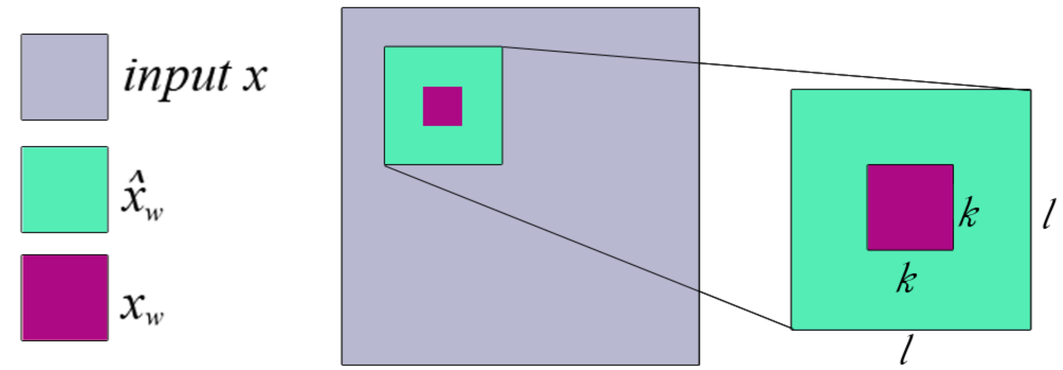


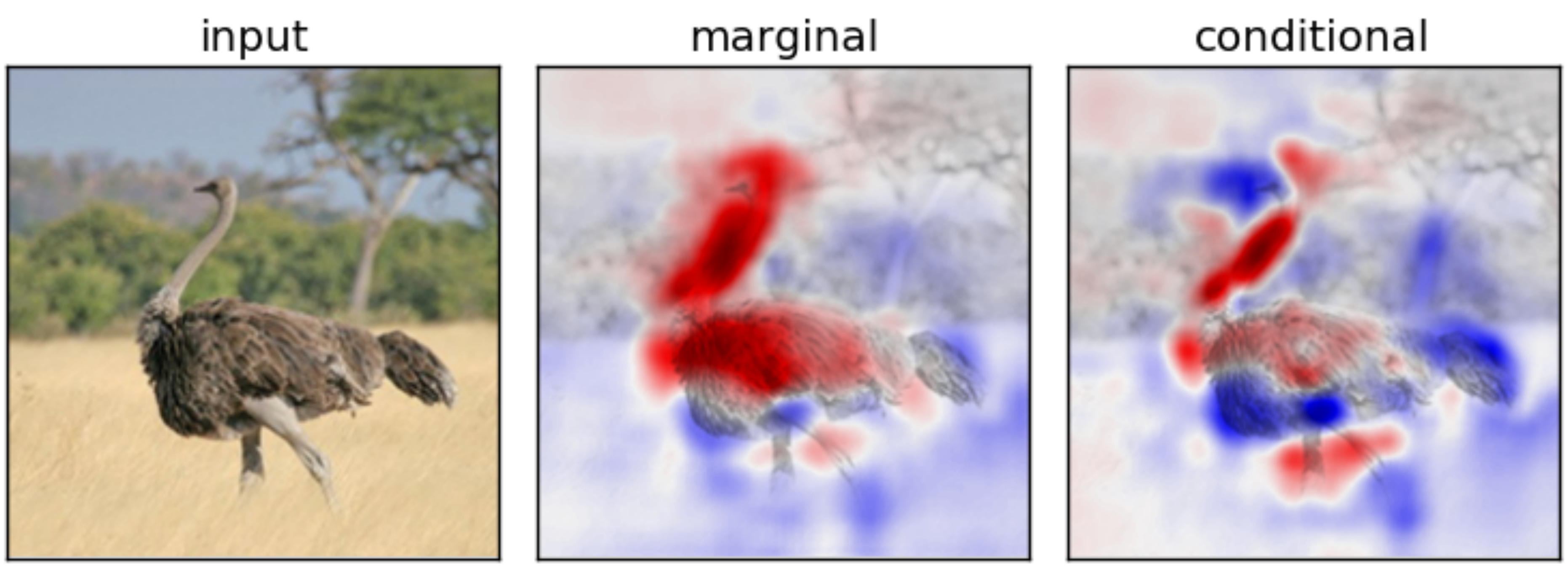
- 40000 events from CLVisc+AMPT model have been used for training
- Another 4000 events from CLVisc+AMPT have been used for testing
- 18000 events from another hydrodynamic model IEBE-VISHNU and CLVisc+IPGlasma model have been used for further testing

Prediction Difference Analysis

VISUALIZING DEEP NEURAL NETWORK DECISIONS: PREDICTION DIFFERENCE ANALYSIS

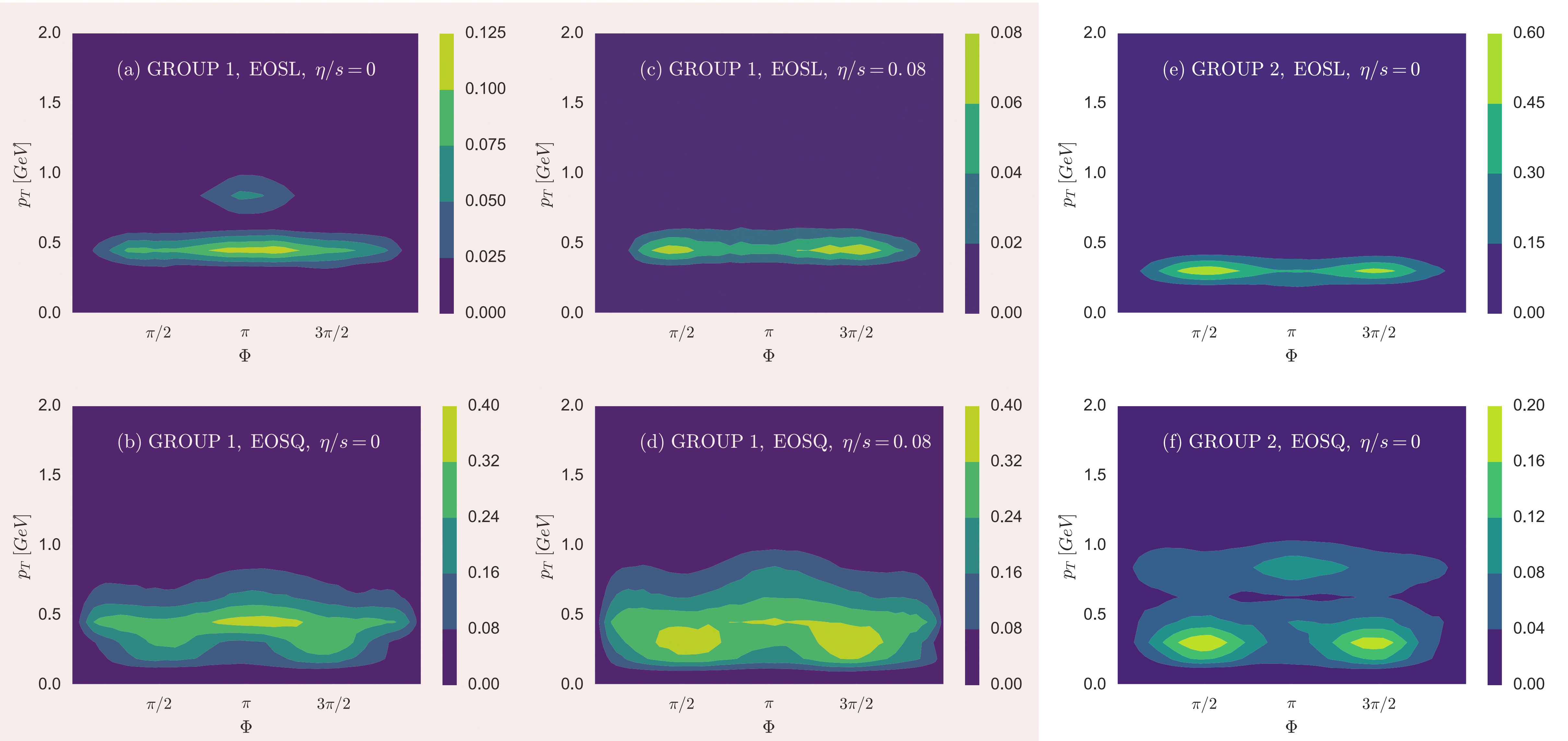
Luisa M Zintgraf[1,3], **Taco S Cohen**[1], **Tameem Adel**[1], **Max Welling**[1,2]
[1]University of Amsterdam, [2]Canadian Institute of Advanced Research, [3]Vrije Universiteit Brussel
{lmzintgraf,tameem.hesham}@gmail.com, {t.s.cohen, m.welling}@uva.nl





- Which region in the image is most relevant for classification

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.

$$C_{12} = < N_A N_B > - < N_A >< N_B >$$

$$N_A = N(p_T = 0.3, \phi = \pm\pi/2)$$

$$N_B = N(p_T = 0.8, \phi = \pi)$$

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%

Summary

- EoS and phase transition are very important in astrophysics and heavy ion collisions
- Deep convolution neural network is the state-of-the-art pattern recognition method in machine learning.
- CLVisc is efficient to provide big amount of training data.
- We demonstrated that a traceable encoder of the QCD phase structure survives the dynamical evolution and exists in the final snapshot of heavy ion collisions, one can efficiently and exclusively decode these information from the highly complex output using machine learning.

Future Challenge 1: hadronic afterburner

Particle production and equilibrium properties within a new hadron transport approach for heavy-ion collisions

J. Weil[1], V. Steinberg[1], J. Staudenmaier[1,3], L.G. Pang[1], D. Oliinychenko[1,2], J. Mohs[1,3], M. Kretz[1,4], T. Kehrenberg[1,3], A. Goldschmidt[1,5], B. Bäuchle[1], J. Auvinen[1,6], M. Attems[1,7] and H. Petersen[1,3,4]
[1] *Frankfurt Institute for Advanced Studies, Ruth-Moufang-Strasse 1, 60438 Frankfurt am Main, Germany*

- Hadronic cascade might be needed before comparing to exp. data
- SMASH: solves relativistic Boltzmann equations for hadron species i:

$$p^\mu \partial_\mu f_i(x,p) + F^\alpha \partial^p_\alpha f_i(x,p) = C^i_{\text{coll}}$$

where C^i_{coll} is the collision term and $F^\alpha = -\partial^\alpha U(x)$ is the force experienced by a individual particle and U(x) is the mean field potential.

Future Challenge 2: detector efficiency

- The detectors can only capture ~80% of the final state hadrons
- Experimental data are corrected with a efficiency factor
- Effect on the classification accuracy
    - Might be not important as animal brains are robust to the resolution and small missing patches of images
    - Using detector simulations and apply the same efficiency correction to the training data

Future Challenge 3: more realistic EoS

- Lattice QCD fails to provide EoS at finite net baryon chemical potential because of the fermion sign problem

- How to get first order phase transition EoS for the finite baryon chemical potential region?

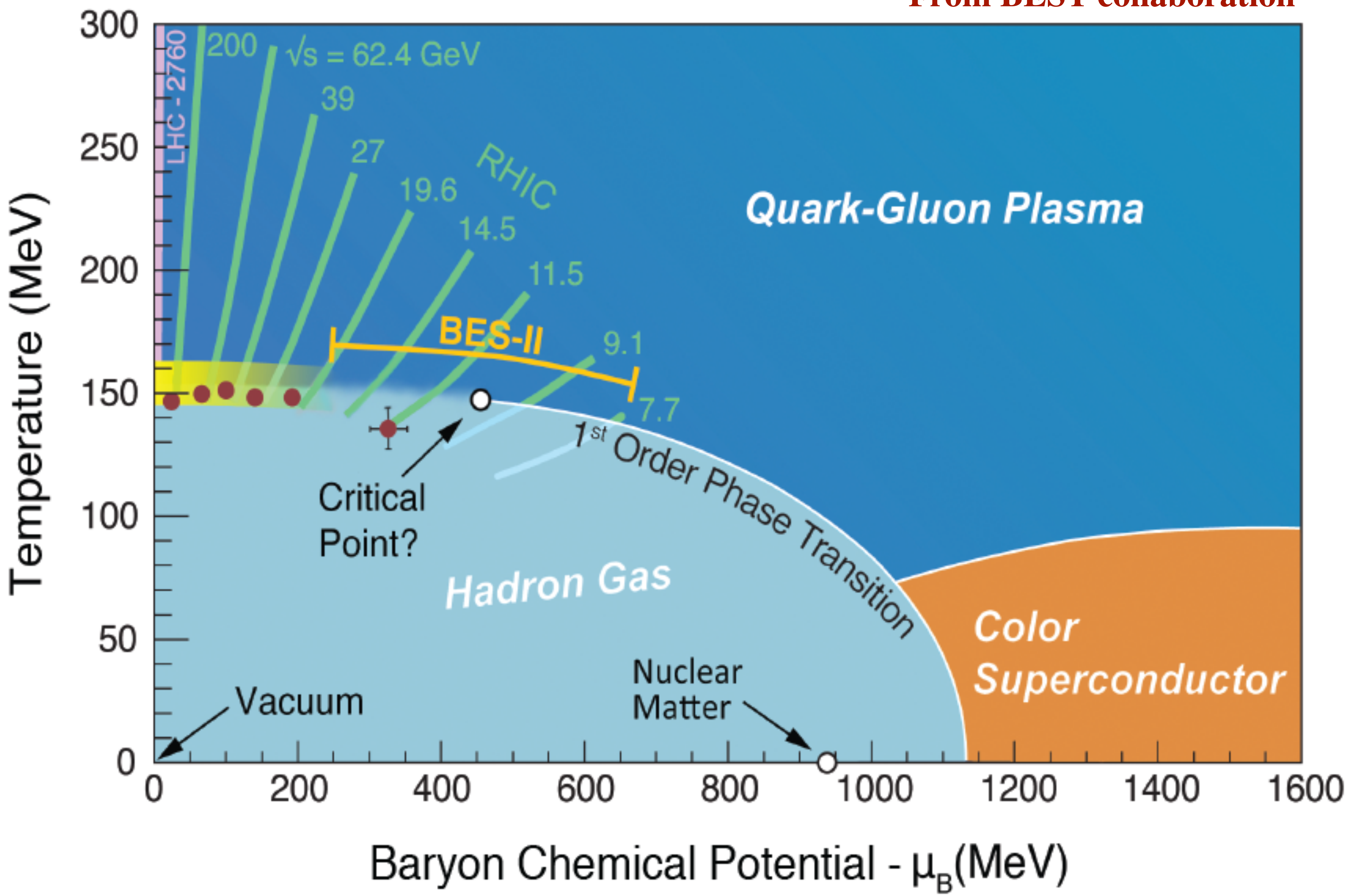
    - It might be possible to prepare millions of different EoS to get particle spectra, which can be used to train a deep neural network to fix the EoS parameters at first order phase transition region using regression.

Backups

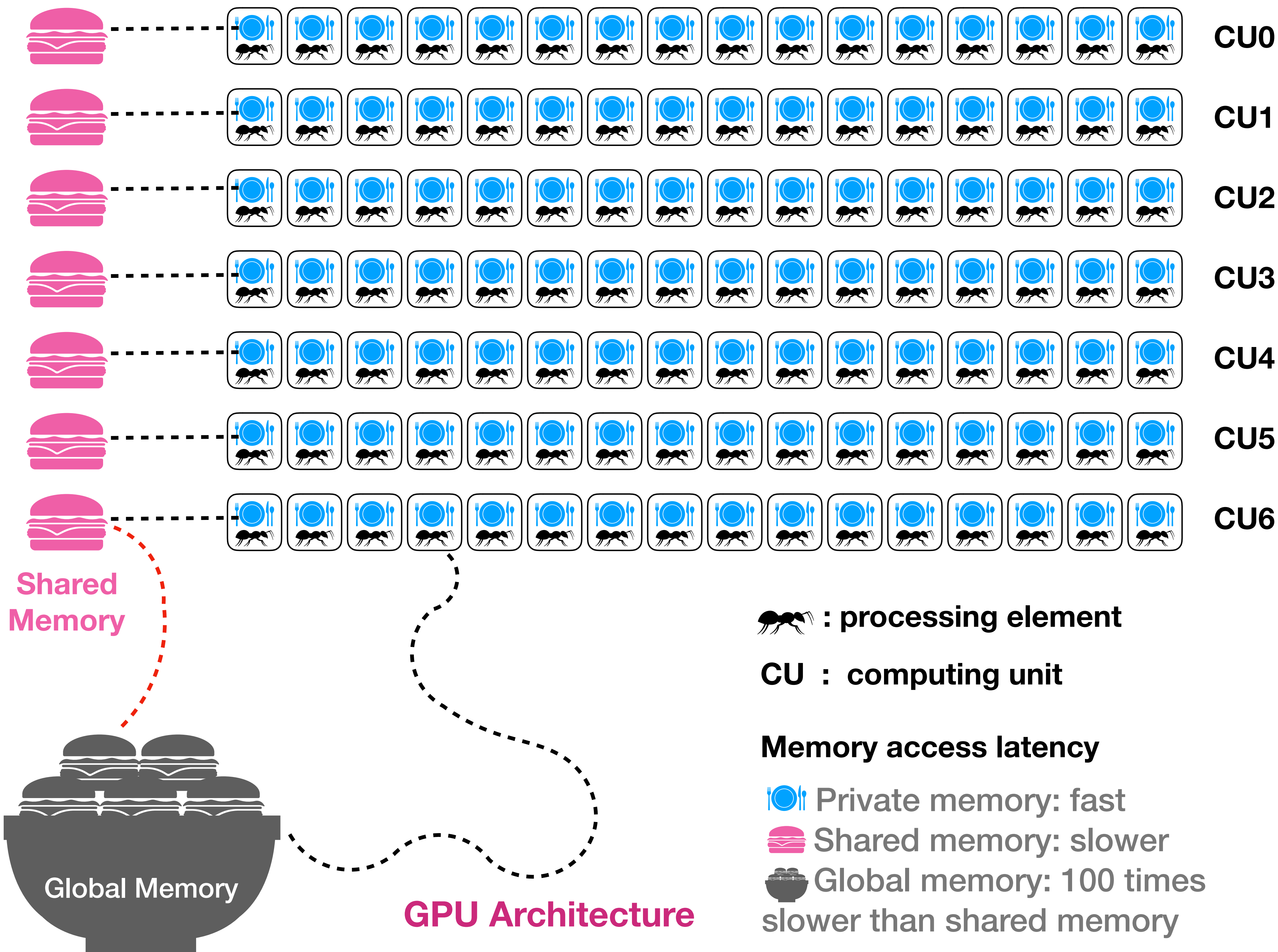
Thanks for your attention!

Beam Energy Scan project to locate the critical end point

From BEST collaboration

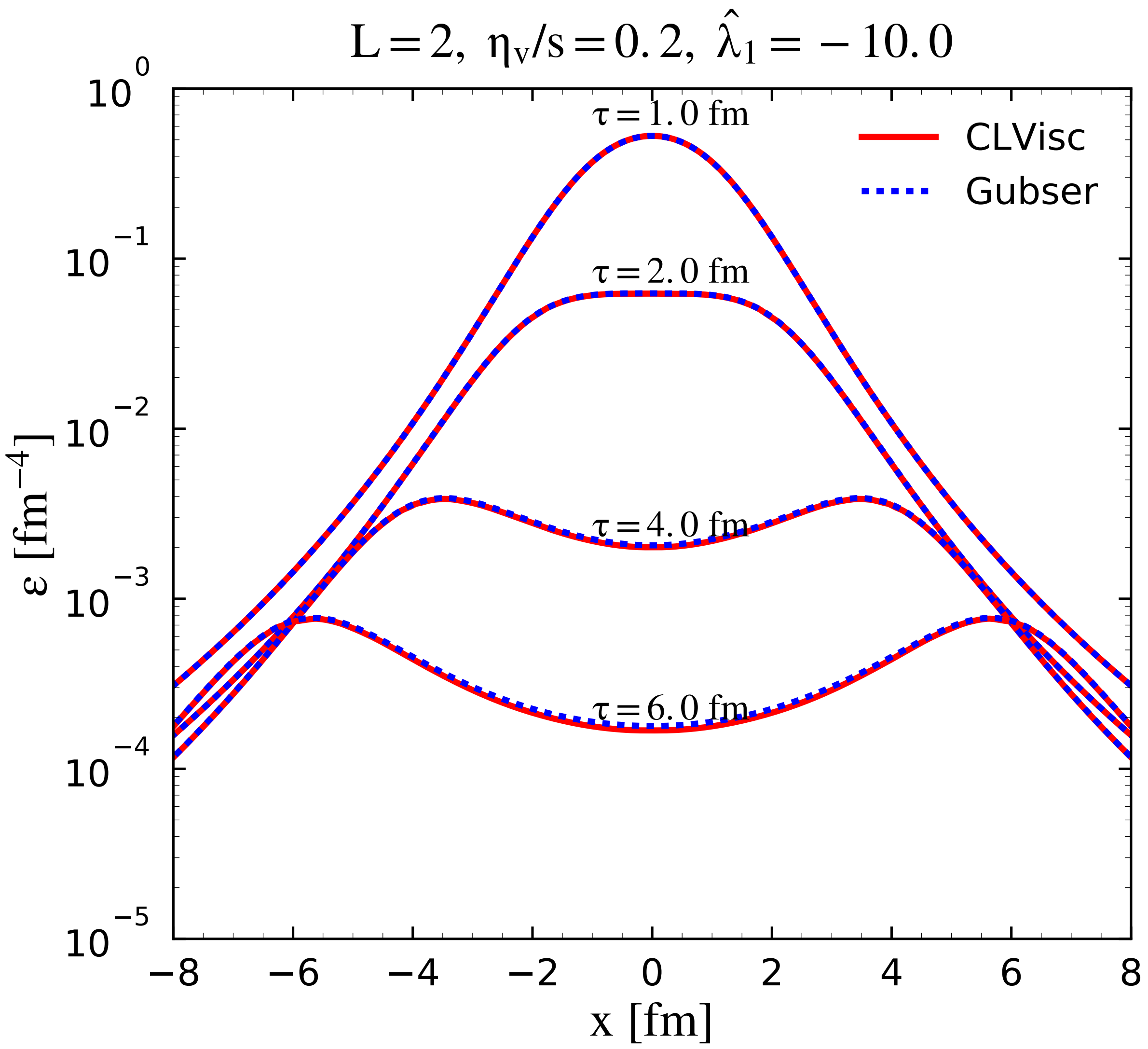


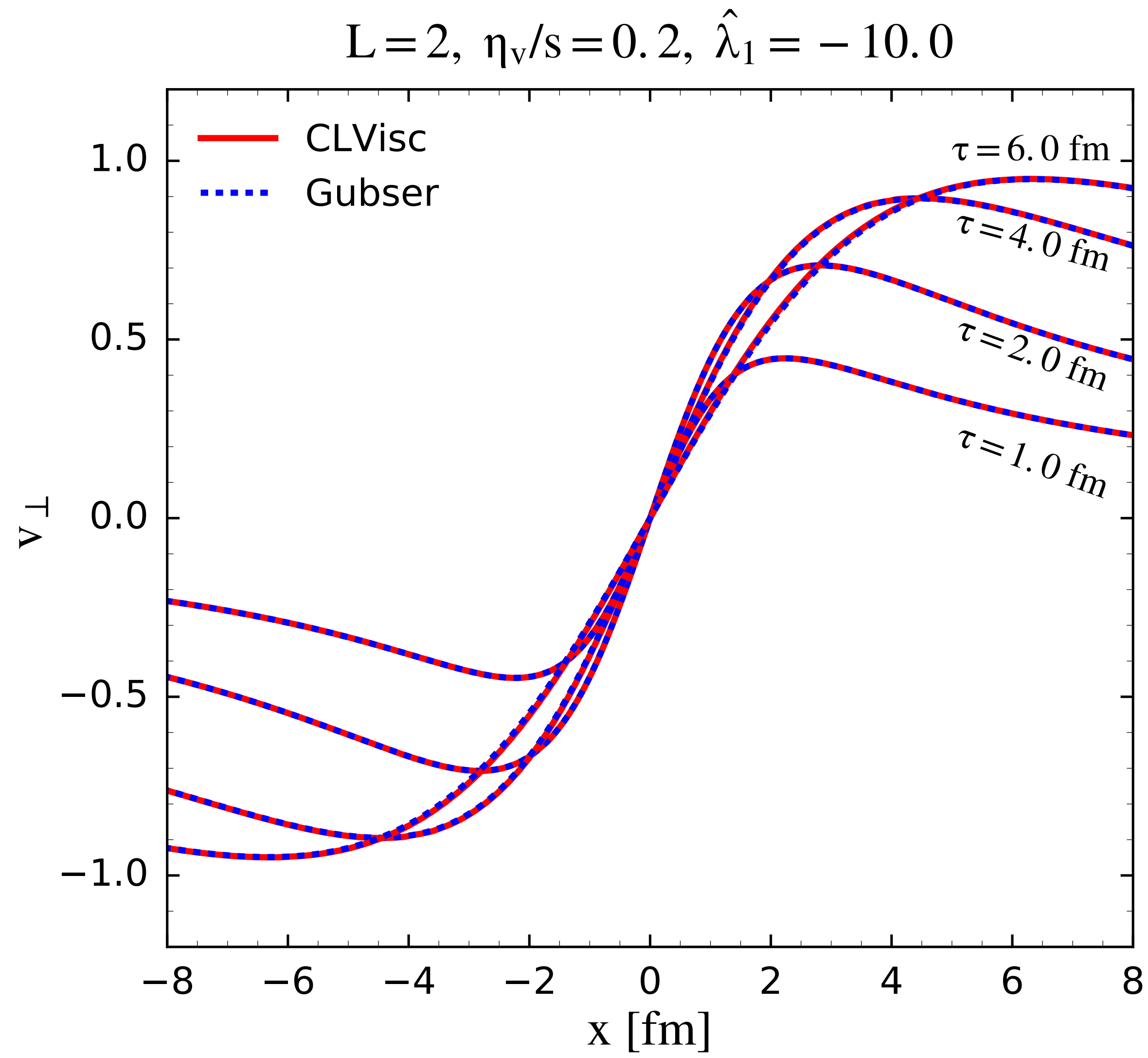
GPU parallelization



LG.Pang, H.Petersen, XN.Wang arXiv:1802.04449

Gubser Solution for 2nd order viscous hydrodynamics

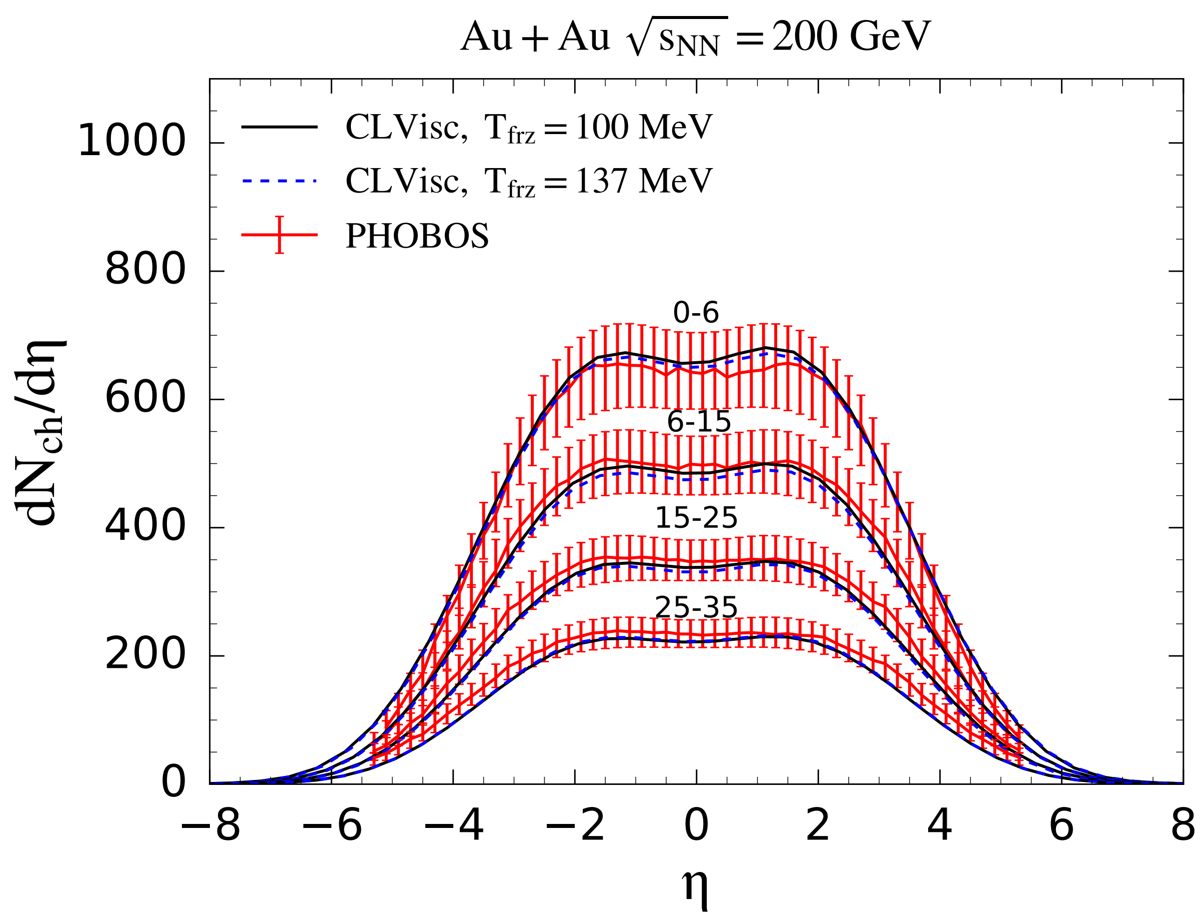


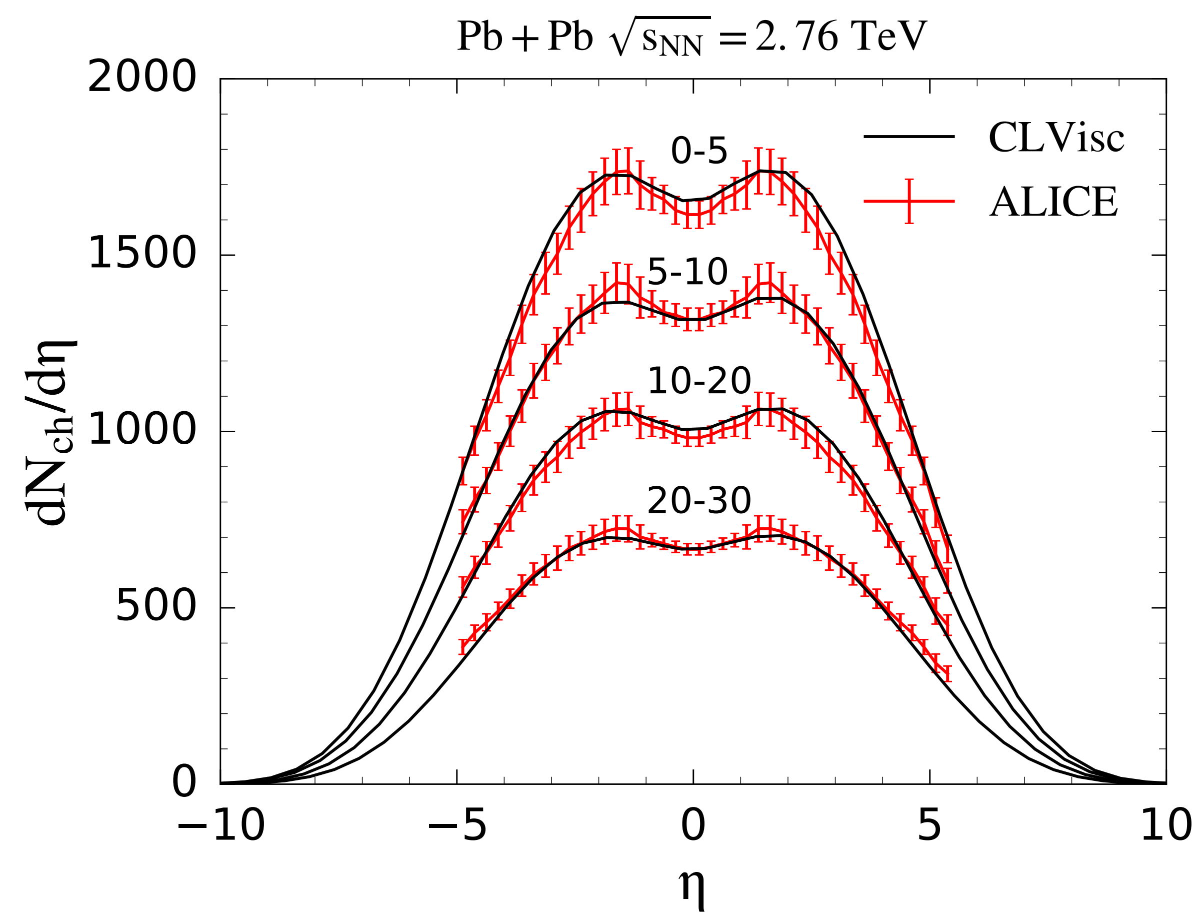


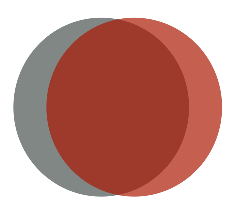
- Gubser solution for 2nd order viscous hydrodynamics, LG.Pang, Y.Hatta, XN.Wang & BW.Xiao Phys.Rev. D91 (2015) no.7, 074027
- Tested with Riemann solution, Bjorken solution and Gubser solution.

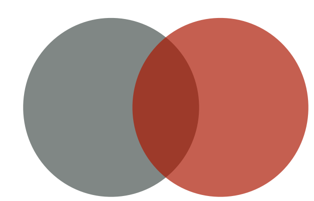
Compare with experiment: charged multiplicity

LG.Pang, H.Petersen, XN.Wang arXiv:1802.04449





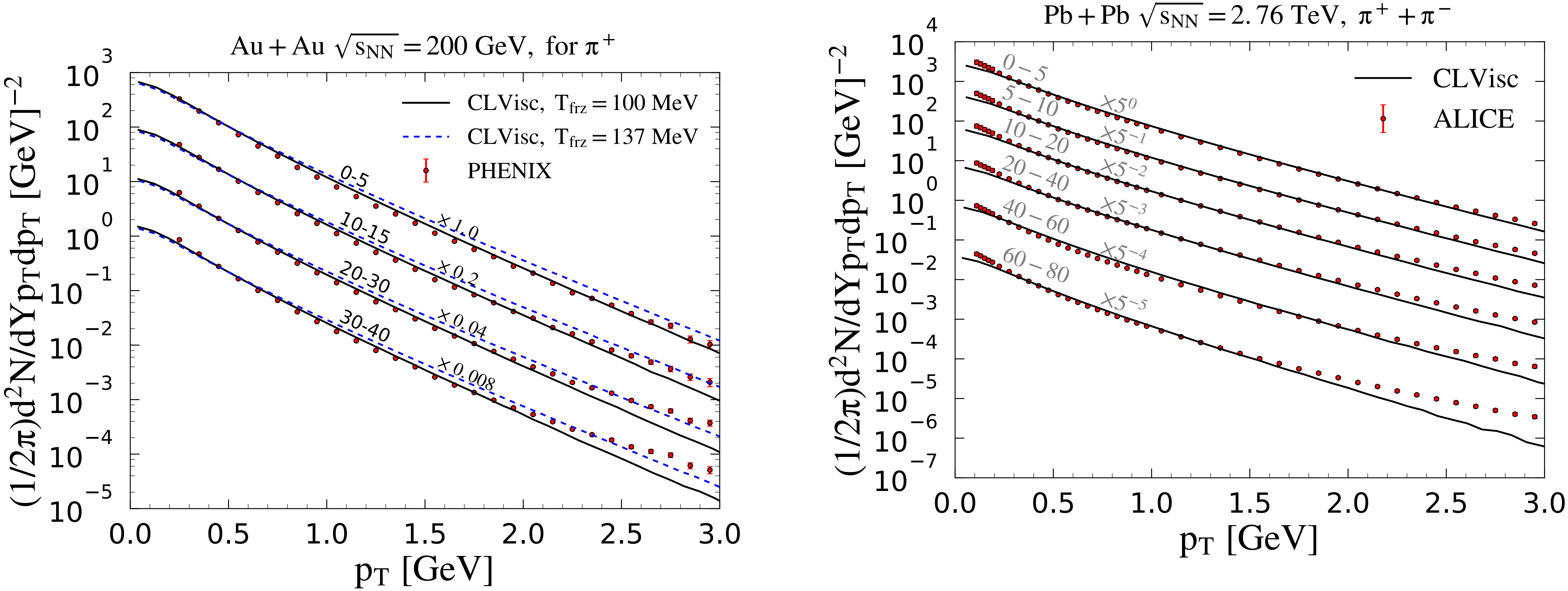
0-5% 

 20-30%

- Fitting with most central collisions, works for all the other centralities.
- Dimension reduction $\rho(\eta, p_T, \phi) \rightarrow \tilde{\rho}(\eta)$
- Where the pseudo-rapidity $\eta = \frac{1}{2} \ln \frac{|\mathbf{p}| + p_z}{|\mathbf{p}| - p_z}$

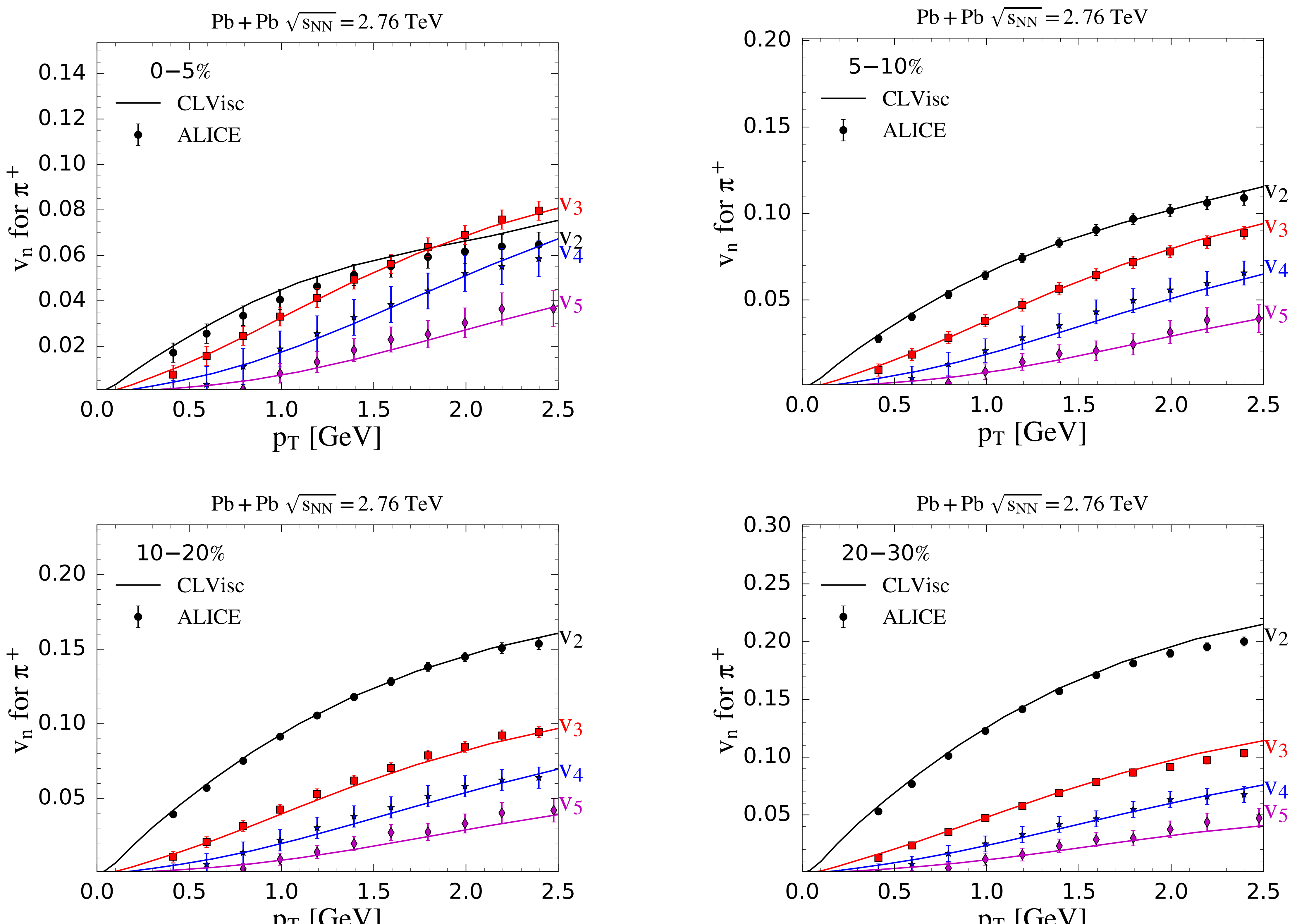
Compare with experiment: transverse momentum spectra

LG.Pang, H.Petersen, XN.Wang arXiv:1802.04449



- Dimension reduction $\rho(\eta, p_T, \phi) \to \tilde{\rho}(p_T)$

Compare with experiment: Fourier decomposition of azimuthal angle distributions



LG.Pang, H.Petersen, XN.Wang arXiv:1802.04449

Open Source Libraries

OPEN SOURCE LIBRARIES

Keras Chainer CNTK TensorFlow Caffe
H2O DEEPLEARNING4J theano torch
DSSTNE Scikit-learn AzureML neon
MXNet DMTK Spark PaddlePaddle WEKA

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.

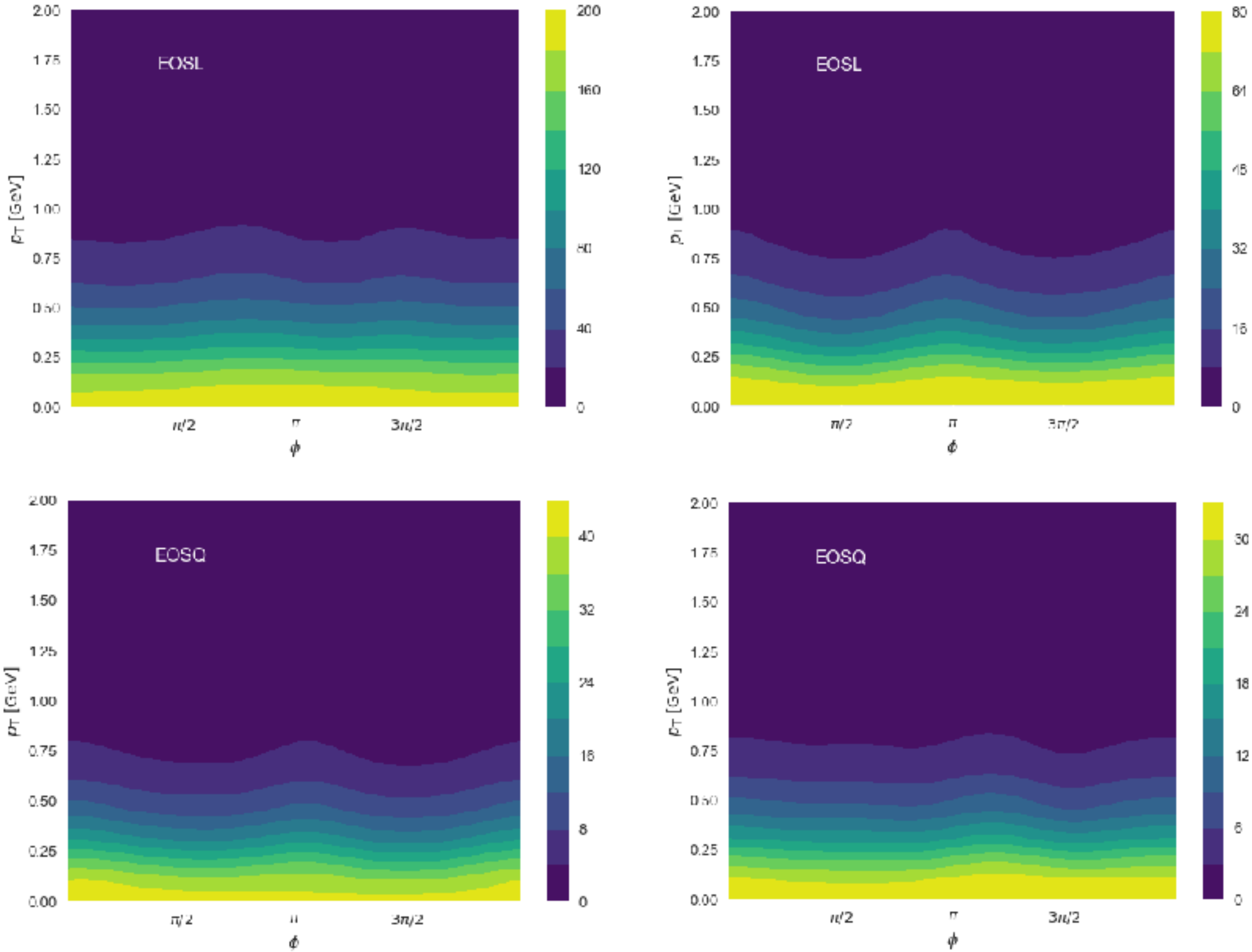
```
# Build one fully connected neural network (784->10->10 neurons) in Keras, for MNIST

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.

Some randomly selected particle spectra



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

Gaussian Naive Bayes Classifier

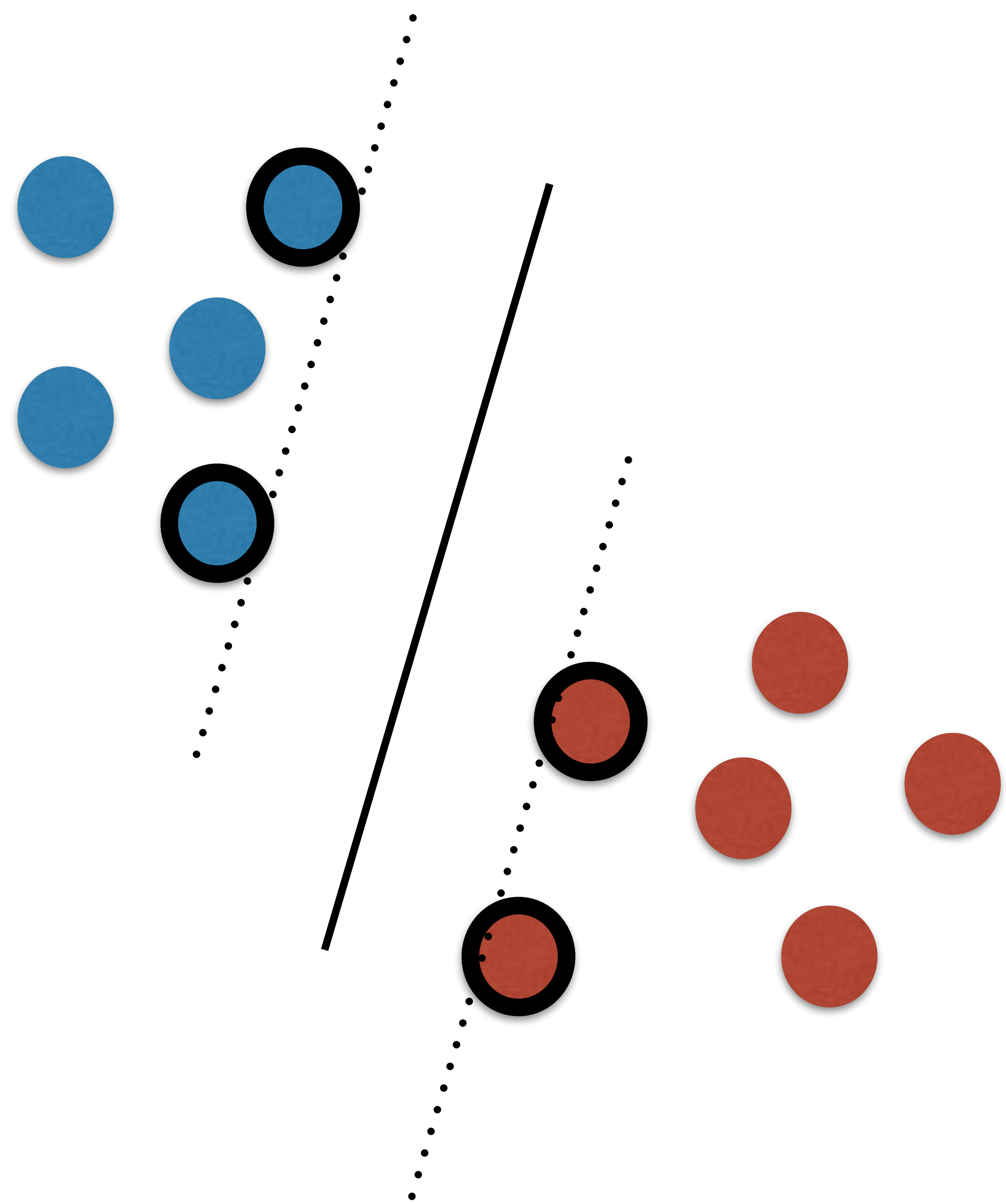
Bayes Classifier:
$$P(c|\mathbf{x}) = \frac{P(c)P(\mathbf{x}|c)}{P(\mathbf{x})}$$

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 affect classification independently
- GNB: For continuous features, using probability density dist.

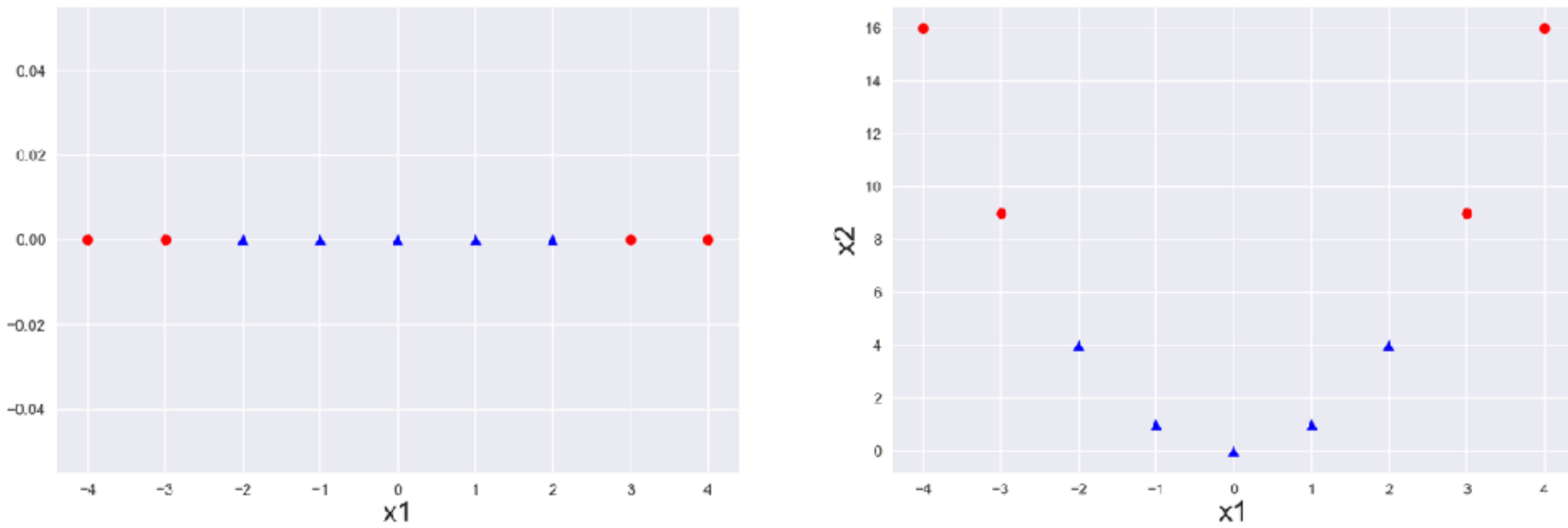
Linear Support Vector Machine Classifier



- 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

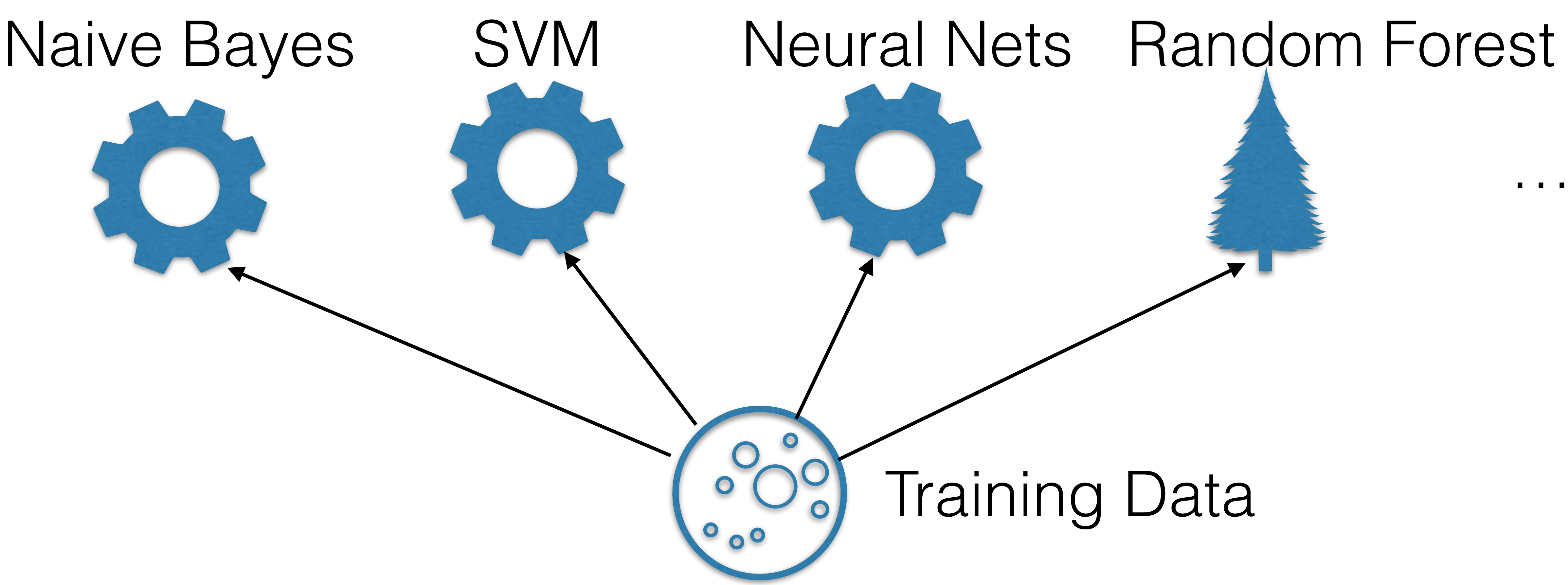


- Left: dataset with one feature x1, not linearly separable
- Right: define x2 = x1 * x1, now linearly separable
- kernels are easier ways to introduce this non-linearity

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

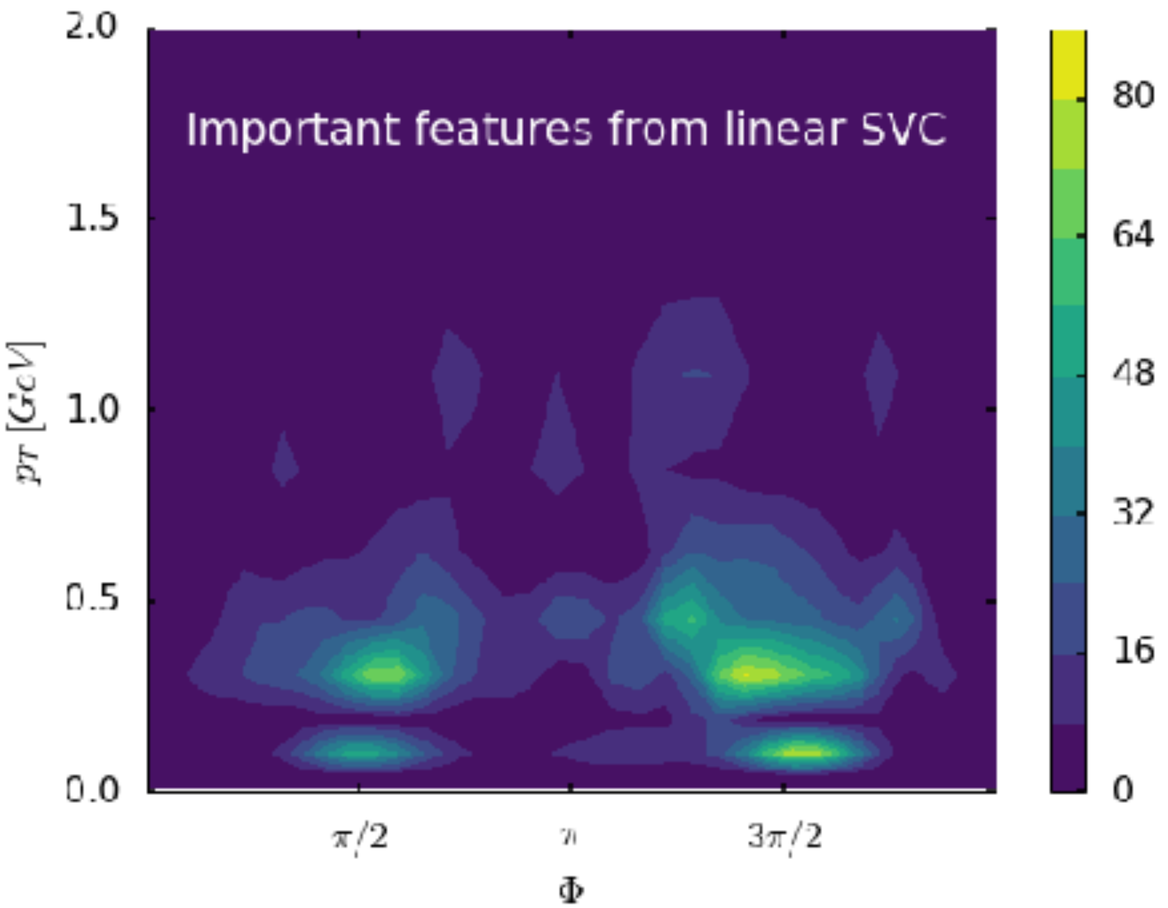
- Boosting: sequentially improve the classifier by paying more attention to misclassified samples
- Example: AdaBoost, XGBoost (many winners of Kaggle data science competing)

Important features from linearSVC

- linearSVC has the best generalization capability for this specific problem.

- If linearSVC trained with pre-defined observables, the most important features in descending order are: 'ptspec-bin4', ' ptspec-bin5', 'ptspec-bin8', 'ptspec-bin7','ptspec-bin6', 'ptspec-bin1', 'dndy', 'ptspec-bin2','ptspec-bin3', 'ptspec-bin11', 'v2-ptbin5', 'v2-ptbin6','v2-ptbin4', 'ptspec-bin9', 'v5-ptbin12', ‘ptspec-bin10', 'v5-ptbin11', 'ptspec-bin12', 'ptspec-bin0', 'v2-ptbin7'.

- If linearSVC trained with raw spectra, the important features are the following,



Hypothesis: The shape of the soft-particle pt spectra along the out-of-plane direction might be very important for EoS classification.

Reason: The expansion along out of plane is weaker and the effect of first order phase transition might be stronger.