

Deep learning in Lattice 1+1d Scalar Field Theory

Kai Zhou 周凯 (FIAS, Frankfurt)

Phys. Rev. D100 (2019) no.1, 011501

In collaboration with :

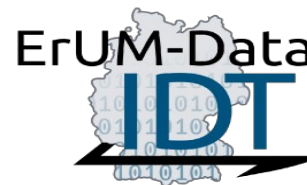
Gergely Endrődi (ITP, Frankfurt)

LongGang Pang (UC Berkeley, CCNU)

Horst Stoecker (ITP, FIAS, Frankfurt)

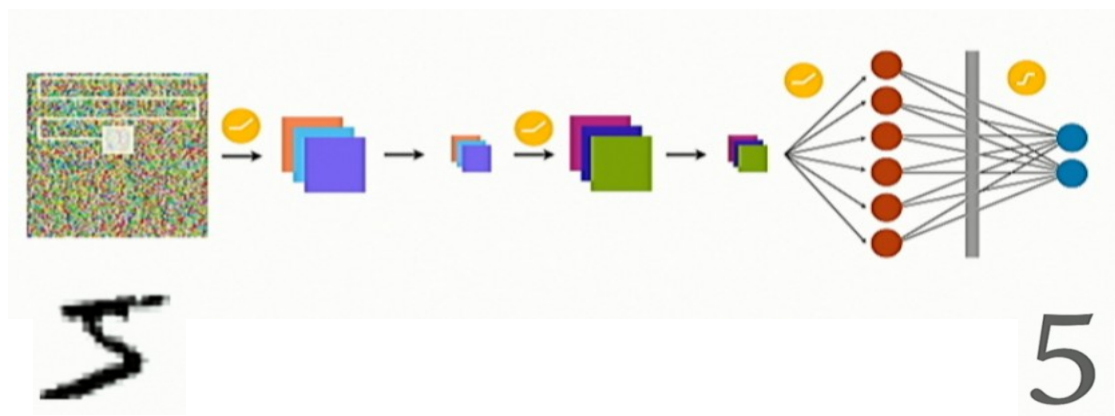
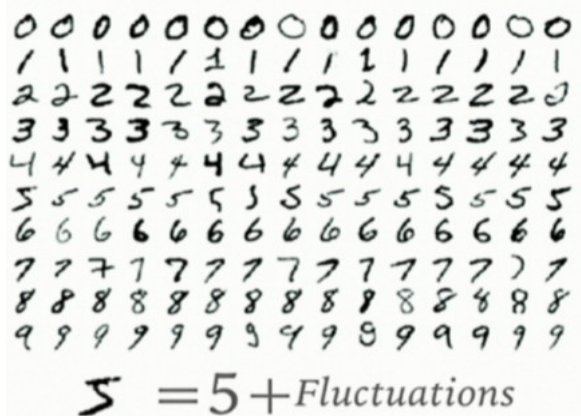


Bundesministerium
für Bildung
und Forschung



Introduction

- Convolutional Neural Network has proved to be extremely powerful in Pattern Recognition, Image Classification



- Discriminative learning (prediction) : Classification, Regression
- Generative modelling (generation) : RBM, VAE, GAN

1+1d $\lambda\phi^4$ field (prepare training set)

regularization of continuum Action:

$$S^{\text{lat}} = \sum_x \left\{ (4 + m^2) \phi^*(x) \phi(x) + \lambda [\phi^*(x) \phi(x)]^2 - \sum_{\nu=1,2} [e^{\mu\delta_{\nu,2}} \phi^*(x) + \hat{\nu}] + e^{-\mu\delta_{\nu,2}} \phi^*(x) \phi(x - \hat{\nu}) \right\}$$

Partition sum :

$$\mathcal{Z} = \int D[\phi] \exp(-S^{\text{lat}}[\phi])$$

Dualization approach :

$$\mathcal{Z} = \sum_{\{k,\ell\}} \prod_n \left\{ e^{\mu k_t(n)} \cdot W[s(n)] \cdot \delta[\nabla \cdot k(n)] \cdot \prod_{\nu} A[k_{\nu}(x), \ell_{\nu}(x)] \right\}$$

$$W[s(n)] = \int_0^{\infty} dr r^{s(n)+1} e^{-(4+m^2)r^2 - \lambda r^4}$$

$$s(n) = \sum_{\nu} [|k_{\nu}(n)| + |k_{\nu}(n - \hat{\nu})| + 2(\ell_{\nu}(n) + \ell_{\nu}(n - \hat{\nu}))]$$

$$A[k_{\nu}(x), \ell_{\nu}(x)] = \frac{1}{(\ell_{\nu}(n) + |k_{\nu}(n)|)! \ell_{\nu}(n)!}$$

C. Gattringer and T. Kloiber, Nucl. Phys. B869 (2013) 56-73
O. Orasch and C. Gattringer, Int. J.Mod.Phys.A33(2016) no.01,1650010,

Configurations - Dualization approach

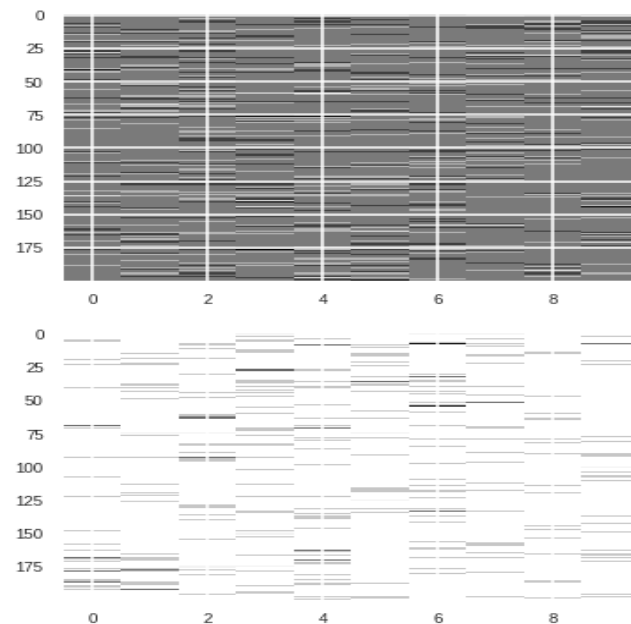
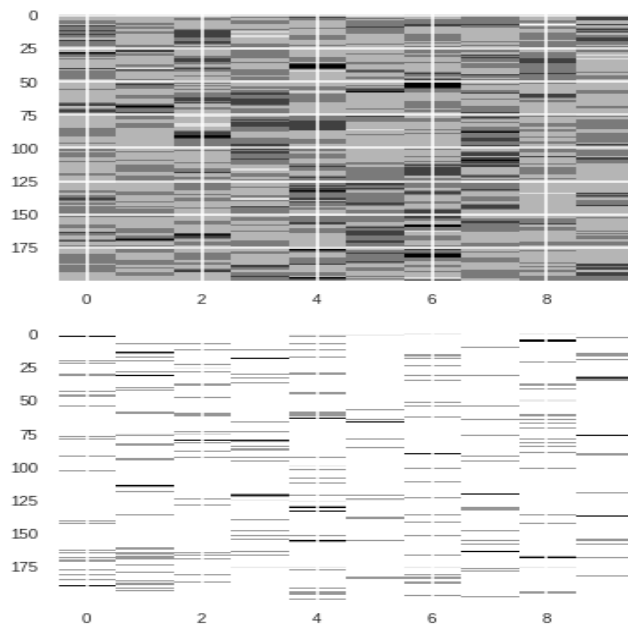
Configurations - 4 integer-valued variables : k_t , k_x , l_t , l_x

$$N_t = 200$$

$$N_x = 10$$

$$m = 0.1$$

$$\lambda = 1.0$$



Divergence constraint :

$$\nabla \cdot k(n) = \sum_{\nu} [k_{\nu}(n) - k_{\nu}(n - \hat{\nu})] = 0$$

Observables : n and $|\phi|^2$

Grand canonical ensemble

$$\langle n \rangle = \frac{T}{L} \frac{\partial \log \mathcal{Z}}{\partial \mu}$$

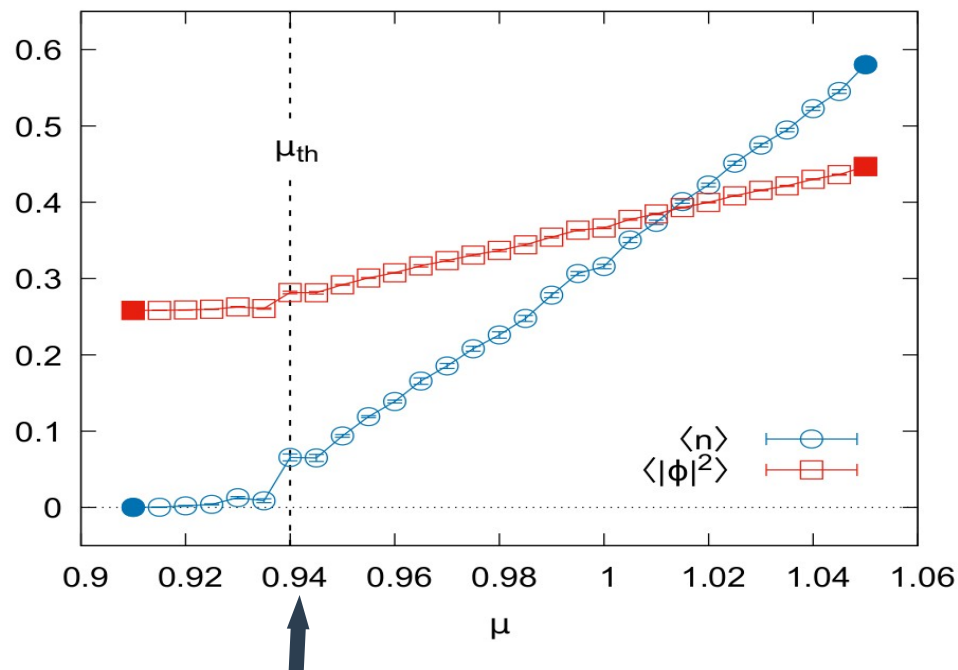
$$n = \frac{1}{N_x N_t a} \sum_n k_t(n)$$

$$\langle |\phi|^2 \rangle = \frac{T}{L} \frac{\partial \log \mathcal{Z}}{\partial (m^2)}$$

$$|\phi|^2 = \frac{1}{N_x N_t} \sum_n \frac{W[s(n) + 2]}{W[s(n)]}$$

Condensation sets in at

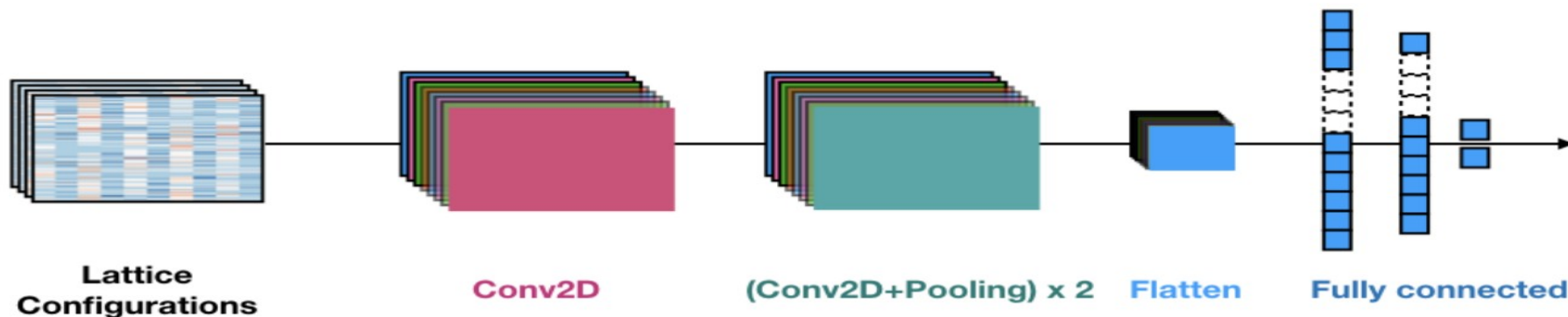
$$\mu_{th} \sim m_{phys} \sim 0.94$$



Exploring NN application here

- (1) Classification : detect 'phase transition' status based on configs
(identify **order parameter**)
- (2) Regression : physical observables regression
(identify **thermodynamics**)
- (3) GAN(generate) : Learn to generate new configurations
Generate configs with proper **distribution**
(identify **partition function**)

DCNN Architecture - Classification

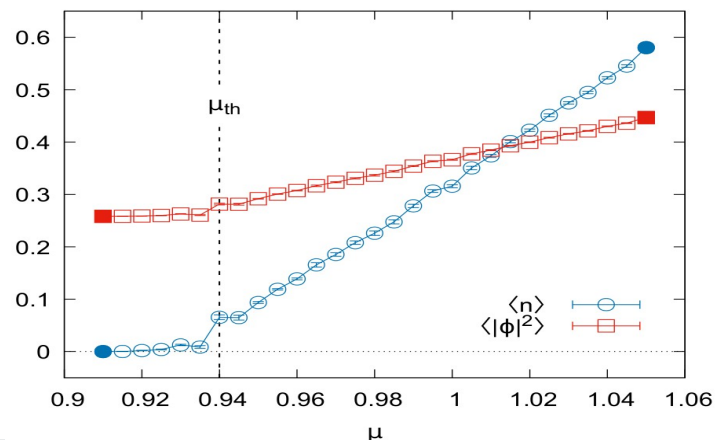


Training set consists of two ensembles of configurations @

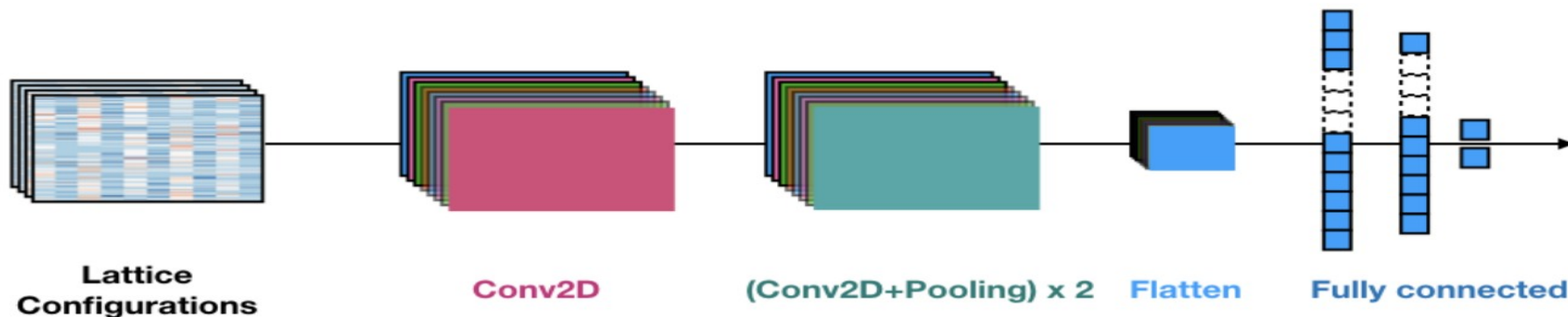
$\mu = 0.91$ with label $y = (0, 1)$

and

$\mu = 1.05$ with label $y = (1, 0)$

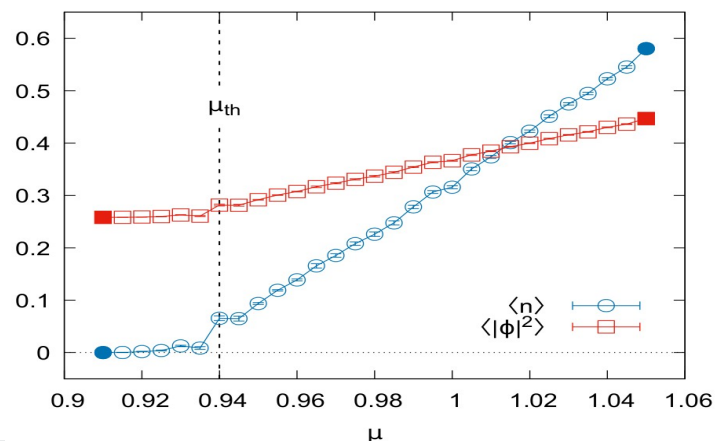


DCNN Architecture - Classification

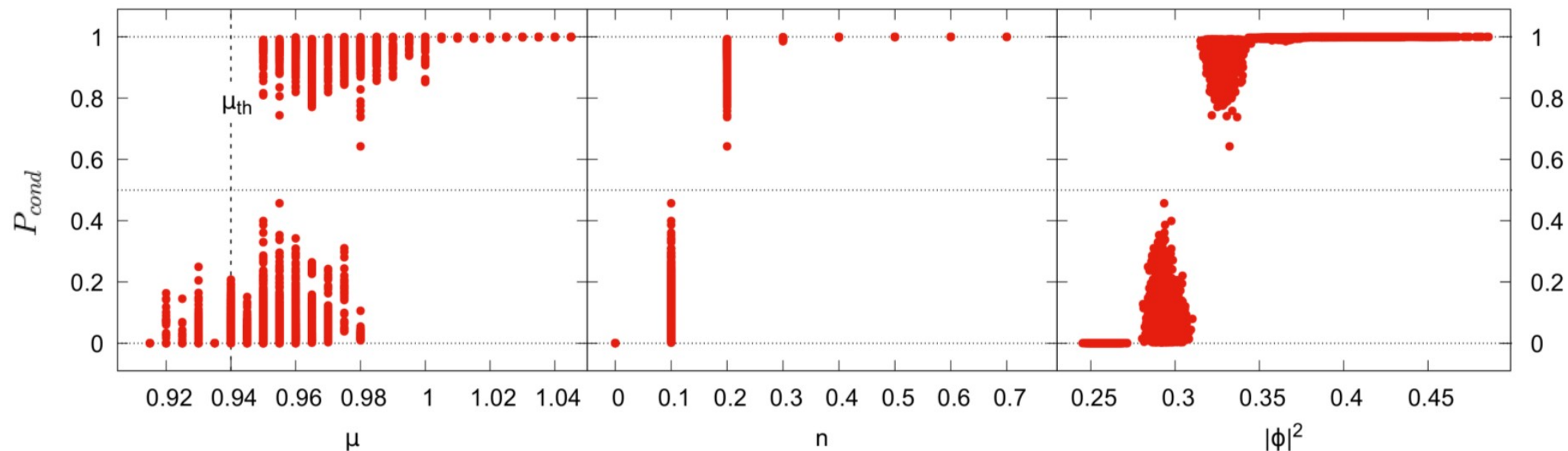


Testing set consists of different ensembles of configurations @ different chemical potential

$$0.91 < \mu < 1.05$$



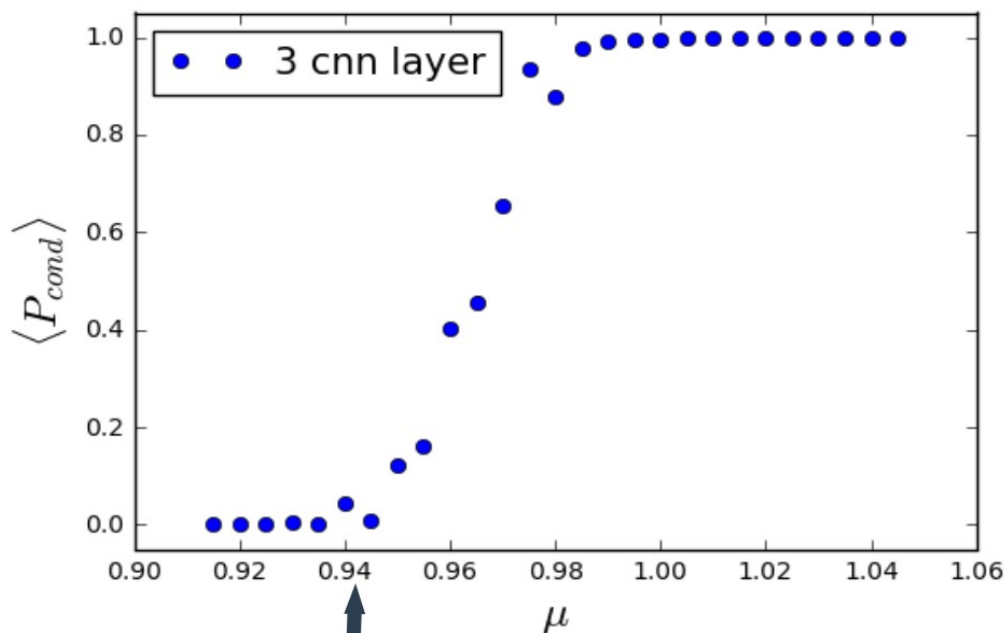
Condensation probability from DCNN



Strong correlation between P_{cond} and observables :
 n , squared field

Ensemble average cond-probability

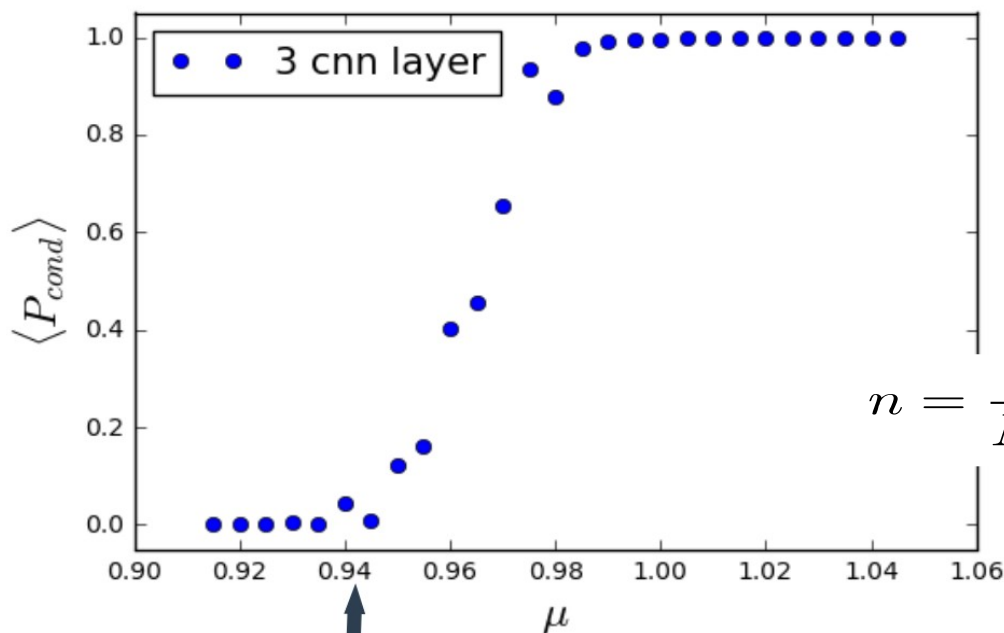
Classifier of the phases : $\langle n \rangle = 0$ and $\langle n \rangle \neq 0$



$$\mu_{th}(\langle P_{cond} \rangle > 0) \sim \mu_{th}(\langle n \rangle > 0)$$

Ensemble average cond-probability

Classifier of the phases : $\langle n \rangle = 0$ and $\langle n \rangle \neq 0$



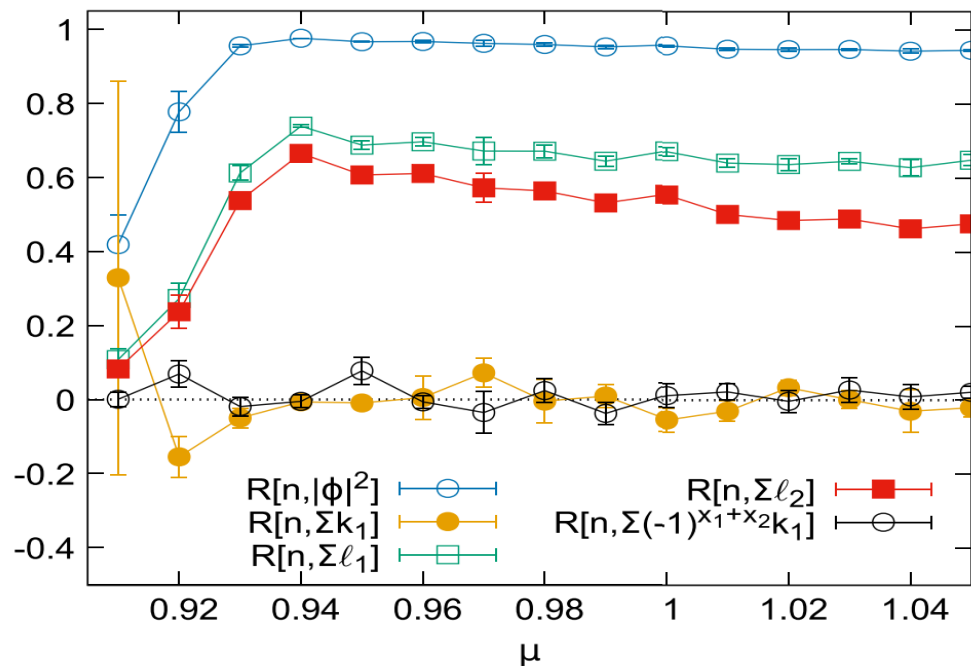
$$n = \frac{1}{N_x N_t a} \sum_n k_t(n)$$

$$\mu_{th}(\langle P_{cond} \rangle > 0) \sim \mu_{th}(\langle n \rangle > 0)$$

Discard kt information

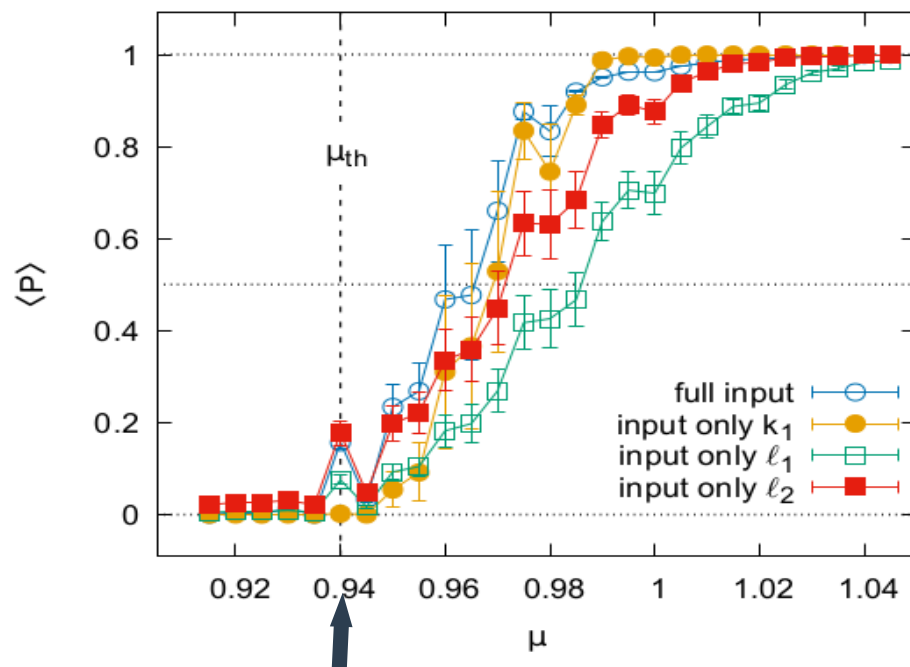
There's finite correlation between kt and l variable, But,
No correlation between kt and kx

$$R[A, B] \equiv \frac{\langle AB \rangle - \langle A \rangle \langle B \rangle}{\sqrt{\langle A^2 \rangle - \langle A \rangle^2} \sqrt{\langle B^2 \rangle - \langle B \rangle^2}}$$



Ensemble average cond-probability

The same transition point came out, even use **only kx** !



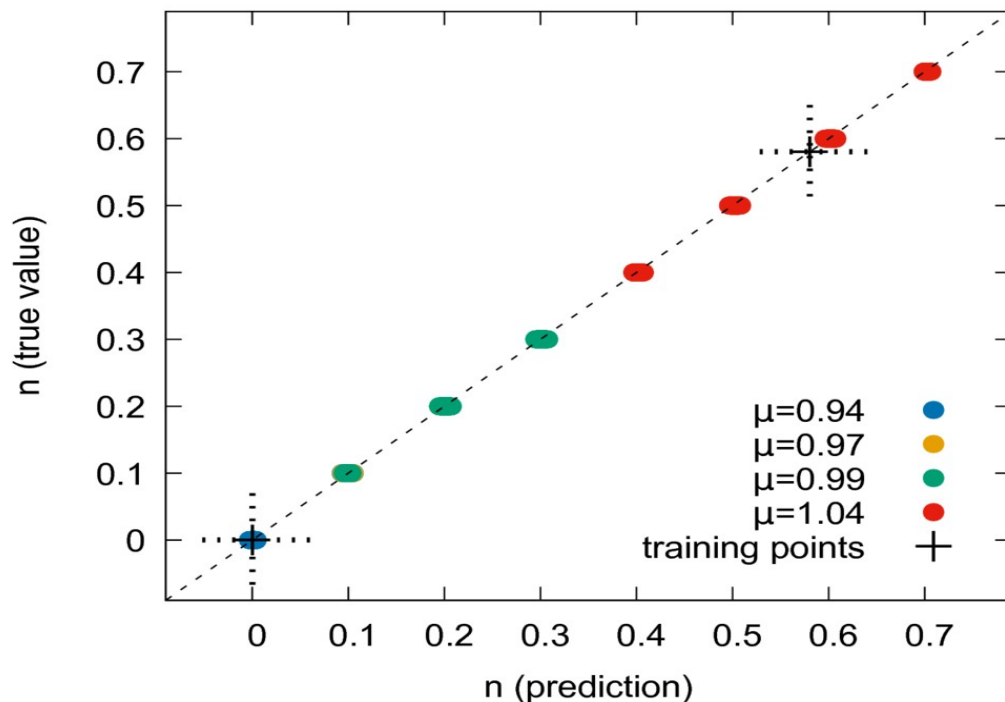
Beyond conventional analysis :

Indicating existence of novel correlation between kx and n

$$\mu_{th}(\langle P_{cond} \rangle > 0) \sim \mu_{th}(\langle n \rangle > 0)$$

Regression for particle density n

Note, for training, only used $\mu = 0.91$ and $\mu = 1.05$

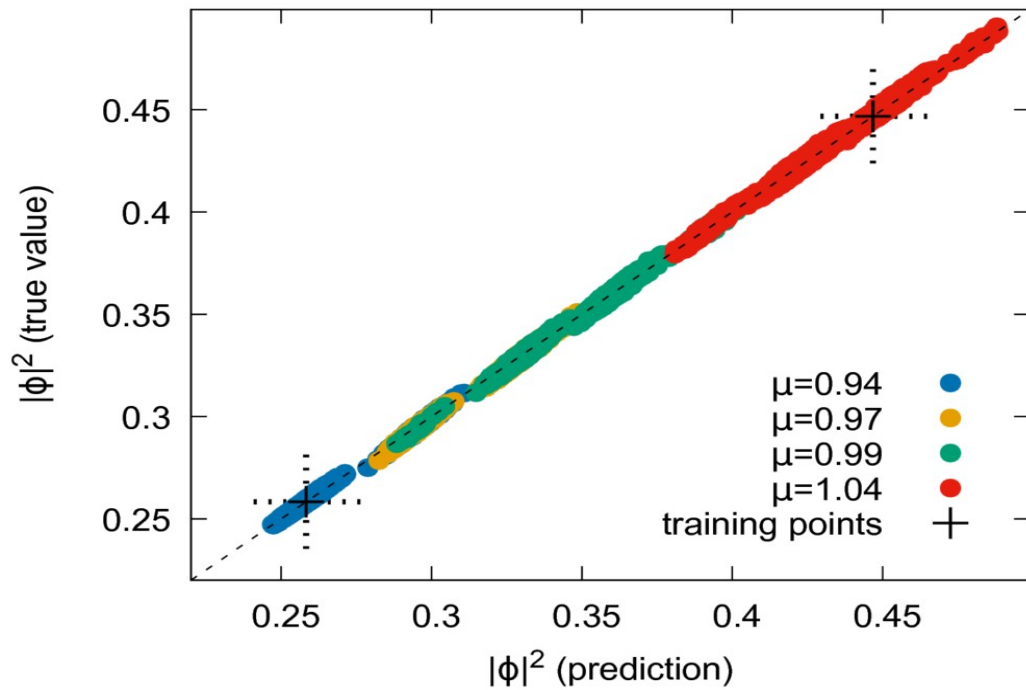


$$n = \frac{1}{N_x N_t a} \sum_n k_t(n)$$

$$RMSE < 0.003$$

Regression for squared field $|\phi|^2$

Note, for training, only used $\mu = 0.91$ and $\mu = 1.05$



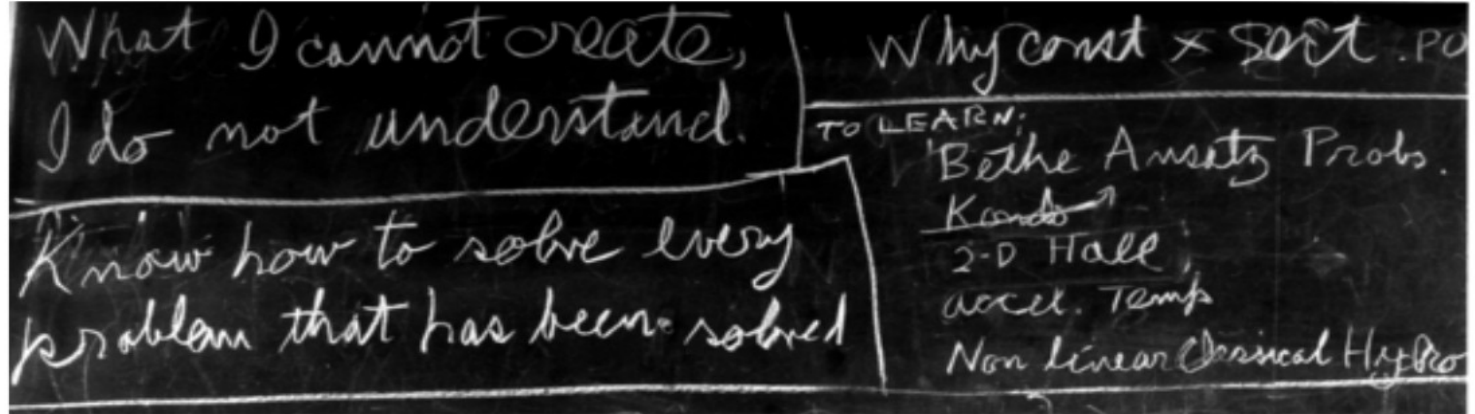
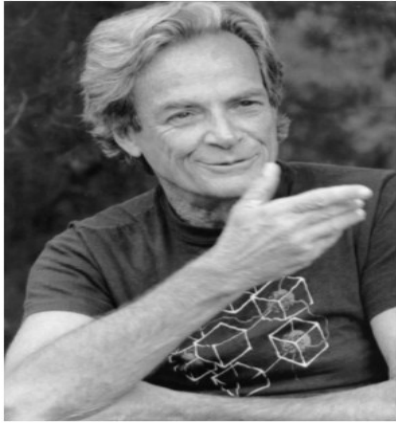
$$|\phi|^2 = \frac{1}{N_x N_t} \sum_n \frac{W[s(n) + 2]}{W[s(n)]}$$

$$W[s(n)] = \int_0^\infty dr r^{s(n)+1} e^{-(4+m^2)r^2 - \lambda r^4}$$

$$s(n) = \sum_\nu [|k_\nu(n)| + |k_\nu(n - \hat{\nu})| + 2(\ell_\nu(n) + \ell_\nu(n - \hat{\nu}))]$$

$$RMSE < 0.005$$

Generate new ones



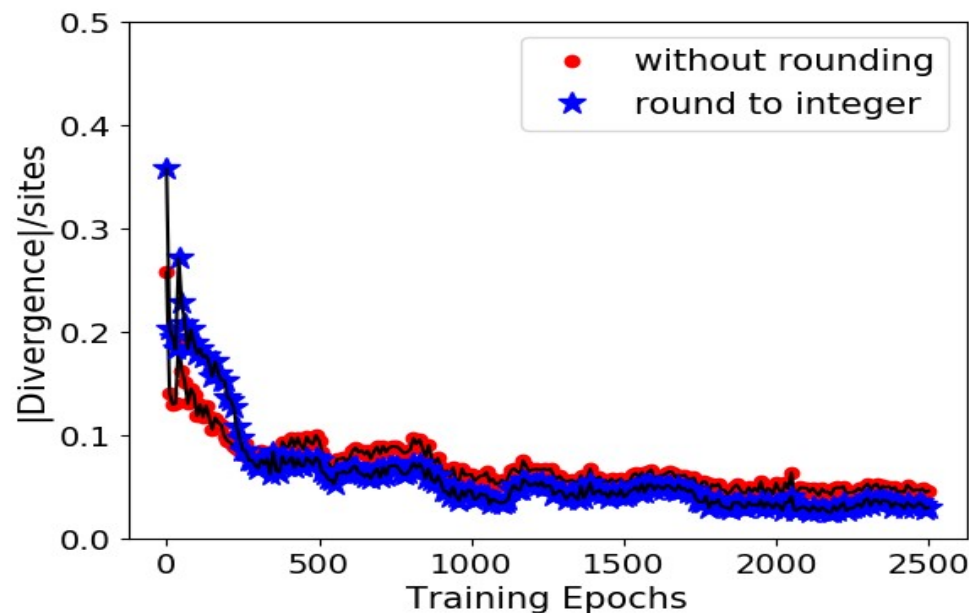
“What I can not create, I do not understand”

GAN - generate proper configurations

The divergence condition get learned automatically:

'Physical' configs
can be generated

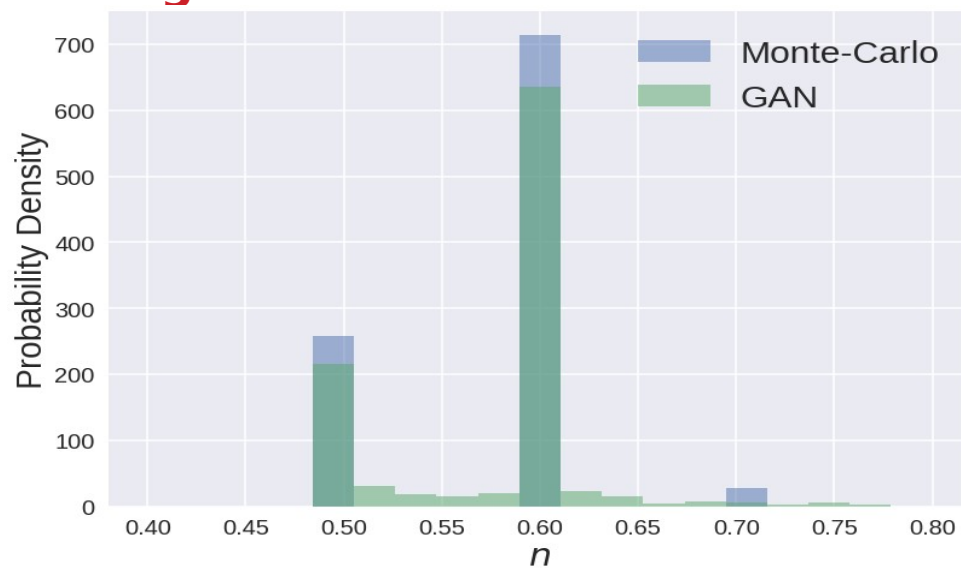
$$\nabla \cdot k(n) = \sum_{\nu} [k_{\nu}(n) - k_{\nu}(n - \hat{\nu})] = 0$$



Automatically capture the **implicit physical constraint!**

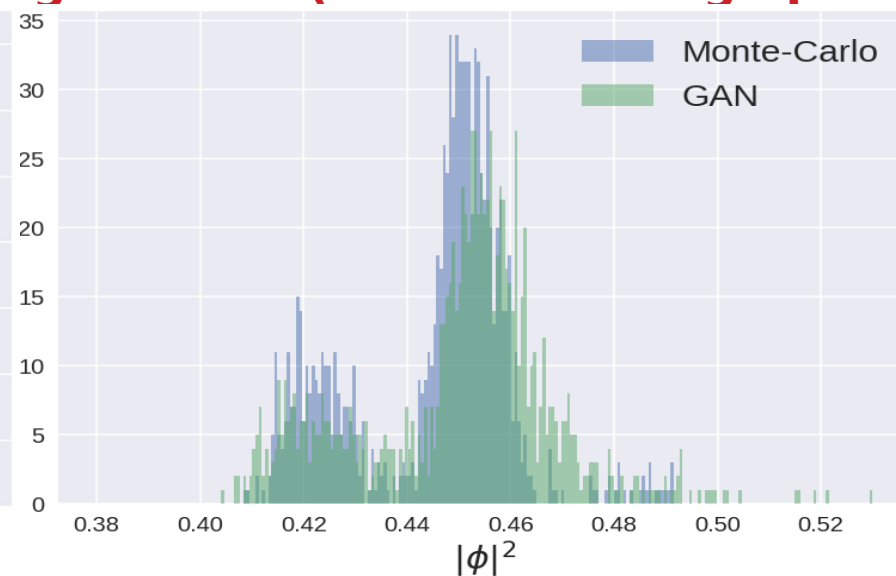
Distribution for observables

Histogram distribution with 1k configurations (after 6k training epochs):



$$\langle n \rangle_{MC} = 0.580$$

$$\langle n \rangle_{GAN} = 0.578$$



$$\langle \phi^2 \rangle_{MC} = 0.447$$

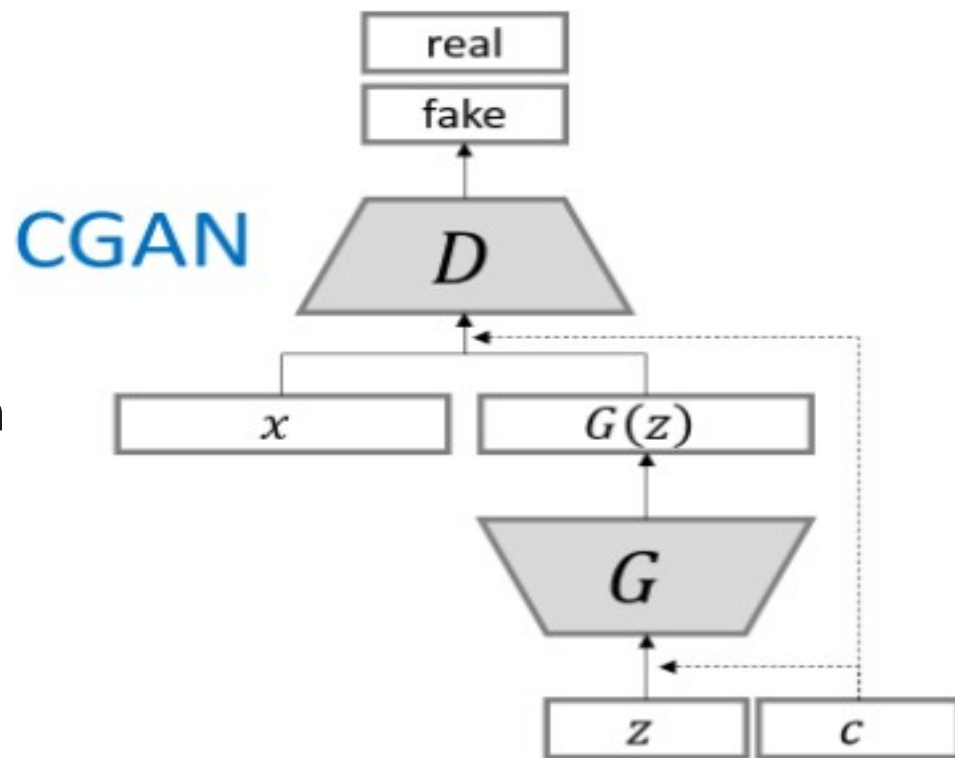
$$\langle \phi^2 \rangle_{GAN} = 0.449$$

add conditional information of n

make GAN conditional on particle density n :

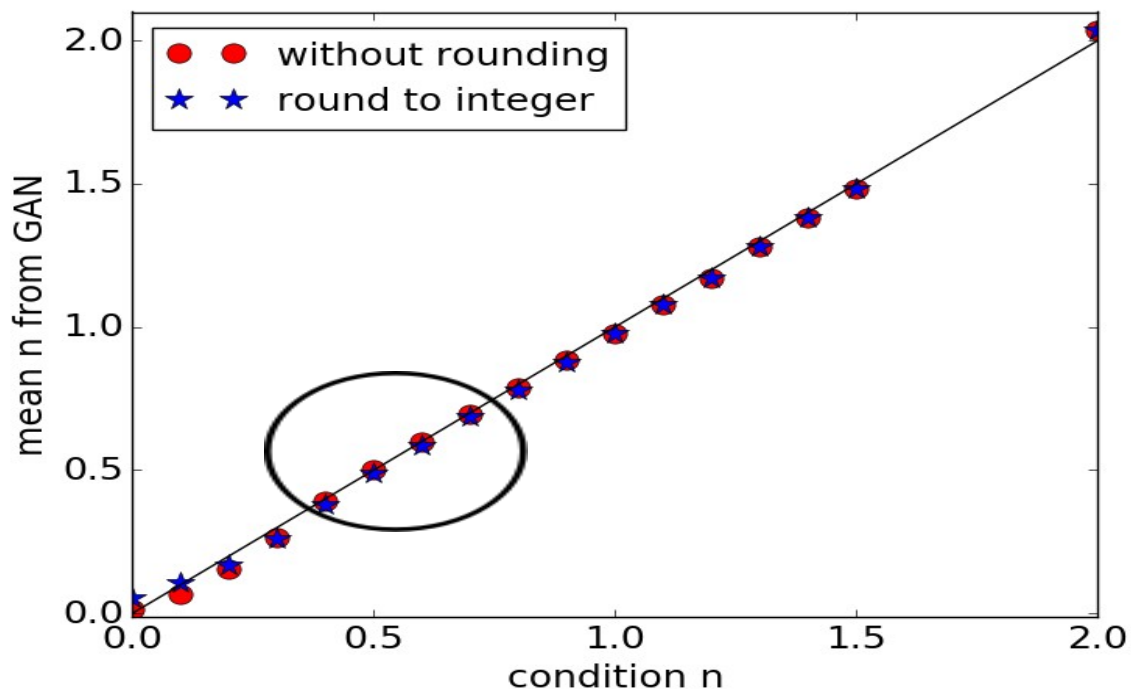
We train GAN using one ensemble with $\mu = 1.05$ labeled as well with n (including $n=0.4, 0.5, 0.6, 0.7$)

Once trained, we specify different n values in generation stage



Conditional GAN - control n

mean value for n and squared field of generated ensemble is controlled by condition in c-GAN.



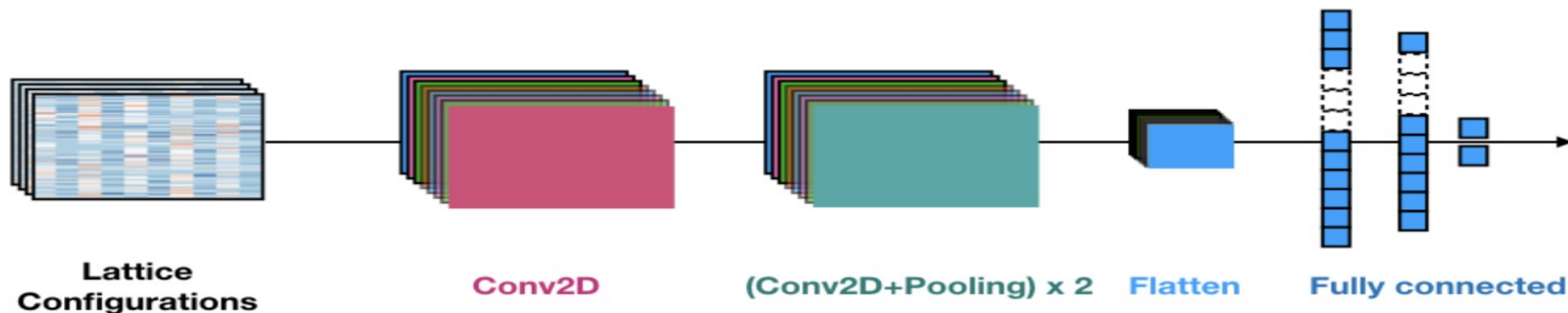
Summary

- (1) Classification : pin down phase 'transition' point
(identify **order parameter**)
- (2) Regression : learn physical observable (**non-linear regression**)
(identify **thermodynamics**)
- (3) GAN(generative): use **GAN** to **generate** new physical configuration
Capture/reproduce data distribution (**use for storage**)
limited grand canonical --> canonical ensemble
limited configs. --> explore phase diagram
(identify **partition function**)

Needs more exploration !

Thanks!

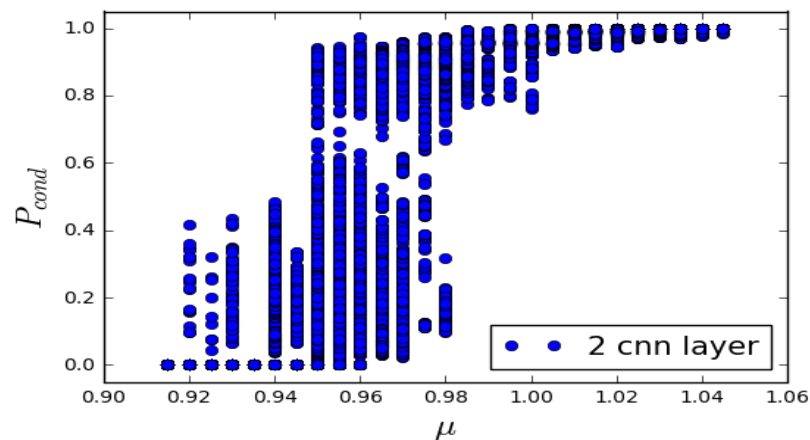
DCNN Architecture - Classification



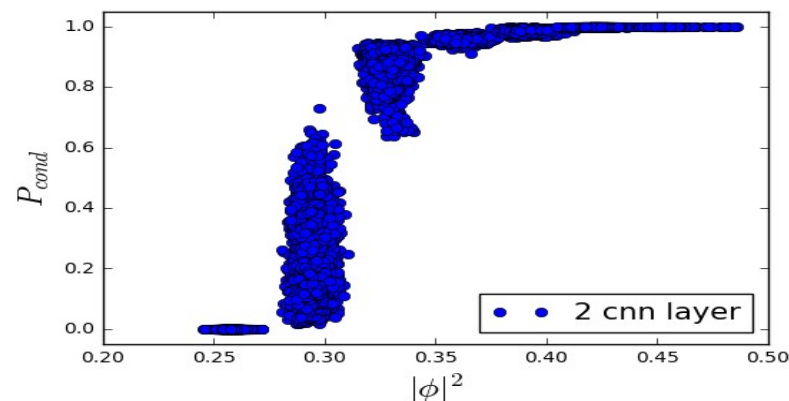
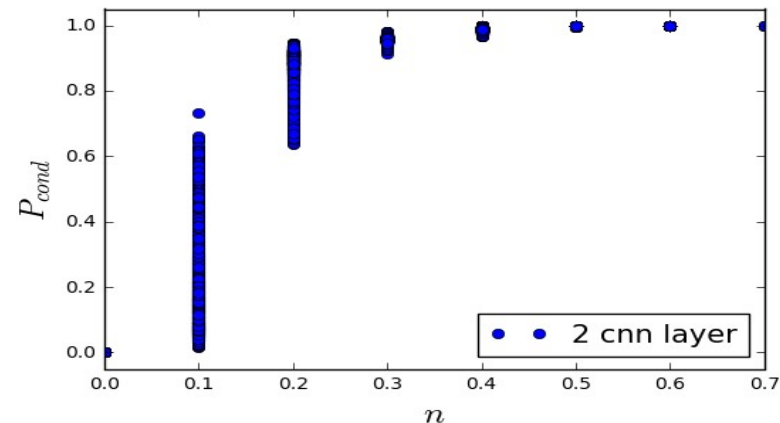
$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] + \lambda \|\theta\|_2^2$$

$\alpha_{lr} = 0.0001$ **With** *AdaMax* **optimization scheme**

Condensation probability from CNN



Deleting one CNN layer
off gives *slightly worse*
distinguish ability



GAN - Nash Equilibrium

Zero-sum game - Nash equilibrium

$$G^* = \arg \min_G \max_D (-\mathcal{L}_D(G, D))$$

$$\mathcal{L}_D = -\mathbb{E}_{\hat{x} \sim p_r(\hat{x})} [\log(D(\hat{x}))] - \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))]$$

$$\mathcal{L}_G = \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))]$$

$$D^*(\hat{x}) = \frac{p_r(\hat{x})}{p_r(\hat{x}) + p_g(\hat{x})}$$

GAN - distribution

