Dualities in and from Machine Learning

Sven Krippendorf

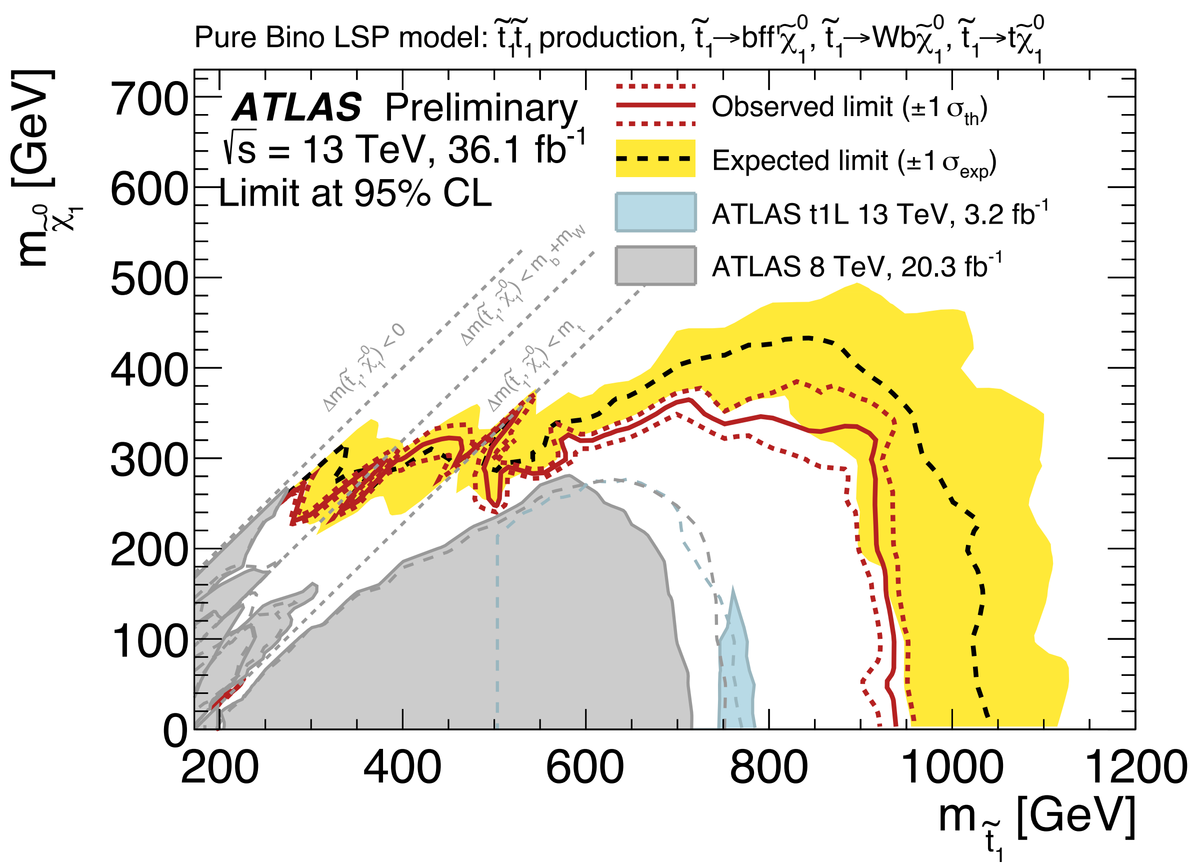
LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

String Phenomenology 2019
CERN, June 29th 2019

Current ML applications in high energy

Improve sensitivity

- Widely used in LHC particle identification, data analysis:



Atlas: 1711.11520

- We can use this for improved bounds on axion-like particles compared to previous bounds in (1704.05256)…

WIP with Francesca Day

Other avenues?

“Don’t ask what ML can do for you,
ask what you can do for ML.”

– *Gary Shiu*

What are the coolest, most powerful methods we have in string theory?

Dualities

Betzler, SK: 190x.xxxxx

Dualities

- Multiple EFTs with different DOF describing the same system.
- Dynamical systems: condensed matter physics, AdS/CFT, string dualities
- Allow us to describe dynamics of strongly coupled systems via dual weakly coupled descriptions
- Allow us to get EFT-operators at higher accuracy than normally allowed (theory: large number of diagrams, experiment: large amount of data). Think about Yukawa couplings in heterotic standard embedding
- …

Dualities

- Multiple EFTs with different DOF describing the same system.
- Dynamical systems: condensed matter physics, AdS/CFT, string dualities
- Allow us to describe dynamics of strongly coupled systems via dual weakly coupled descriptions
- Allow us to get EFT-operators at higher accuracy than normally allowed (theory: large number of diagrams, experiment: large amount of data). Think about Yukawa couplings in heterotic standard embedding
- …

What’s your favourite application?

How are dualities useful in practice?

Dualities

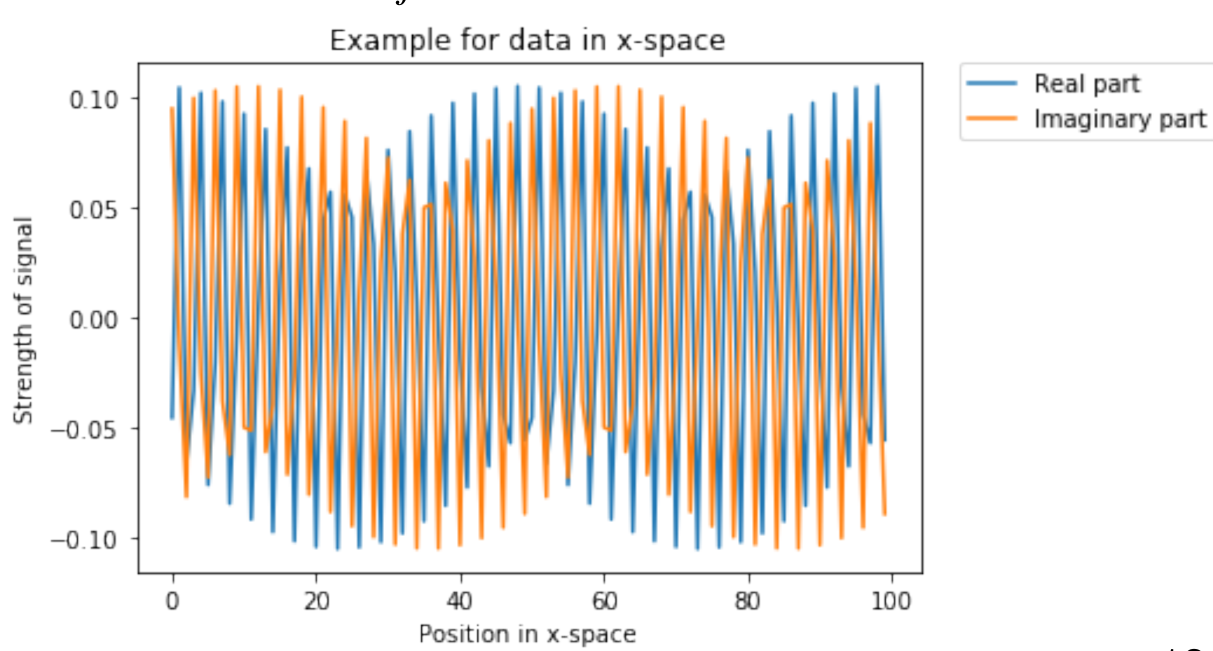
- What's the use of these multiple EFT descriptions?
- Different EFTs describe different properties (correlators) more efficiently, i.e. determined by `leading order' behaviour in one set of fields and complicated functions in dual set of fields.
- Understand the EFT in one frame better (i.e. higher order corrections) due to constraints from dual description.
- **Good news:** plenty of well-studied examples to adapt/ develop ML techniques: AdS/CFT (η/s), 2D Ising model, Yukawa couplings in heterotic standard embedding (complex structure moduli metric)…
  Many open questions on how higher order EFT operators in string theory look like.

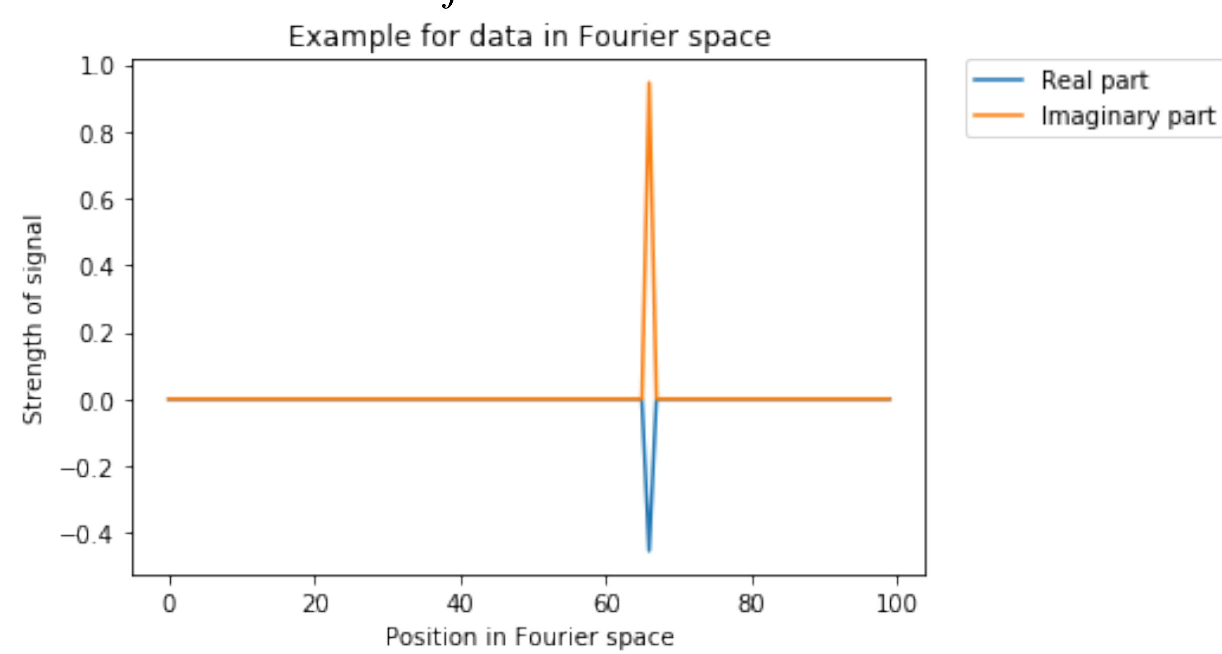
Why interesting for ML?

Why interesting for ML?

- (Deep) Neural networks can transform data into different representation.
- Finding good data representations is hard and often out of the realm of current optimisation strategies (see DFT below).
- Different EFTs describe different properties (correlators) more `efficiently'. Such representations are relevant!
- It allows us to determine a property of the data more easily.
- Example: Discrete Fourier Transformation

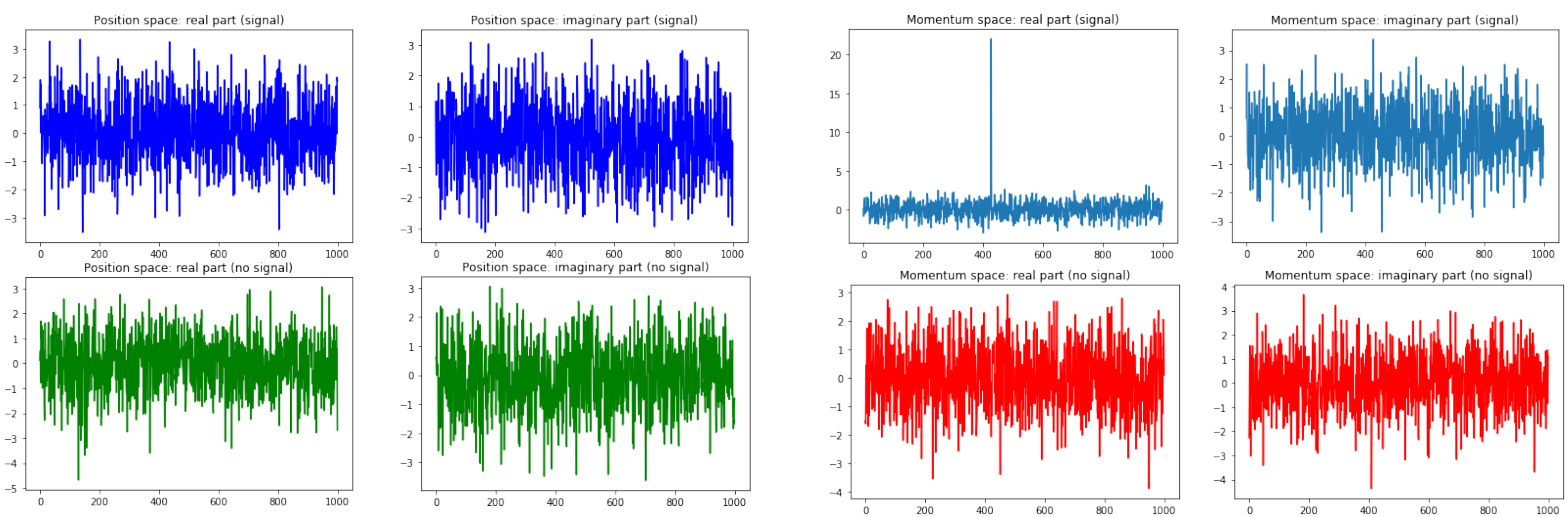
$$x_k = \frac{1}{n}\sum_{j=1}^{n} p_j \; e^{2\pi ijk/n} \qquad\qquad p_k = \sum_{j=1}^{n} x_j \; e^{-2\pi ijk/n}$$





Fourier transformation

- Let's confront this with a data question: Is there a signal in the noise?



- Let's check the performance on simple networks

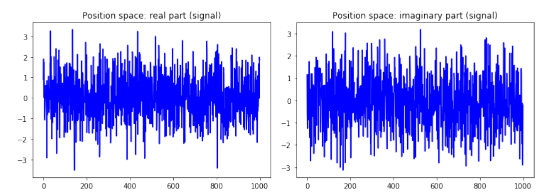
DFT: simple network

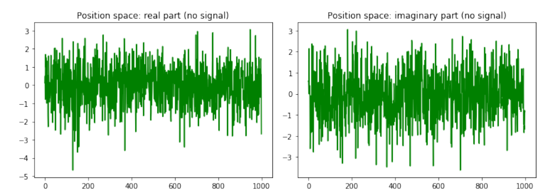
- Supervised learning task (binary classification):

N discrete values

$\{((\mathbf{x}_R, \mathbf{x}_I), y)\}$ $\{((\mathbf{p}_R, \mathbf{p}_I), y)\}$

$y = 1$ **noise + signal**





$y = 0$ **noise**

- Network

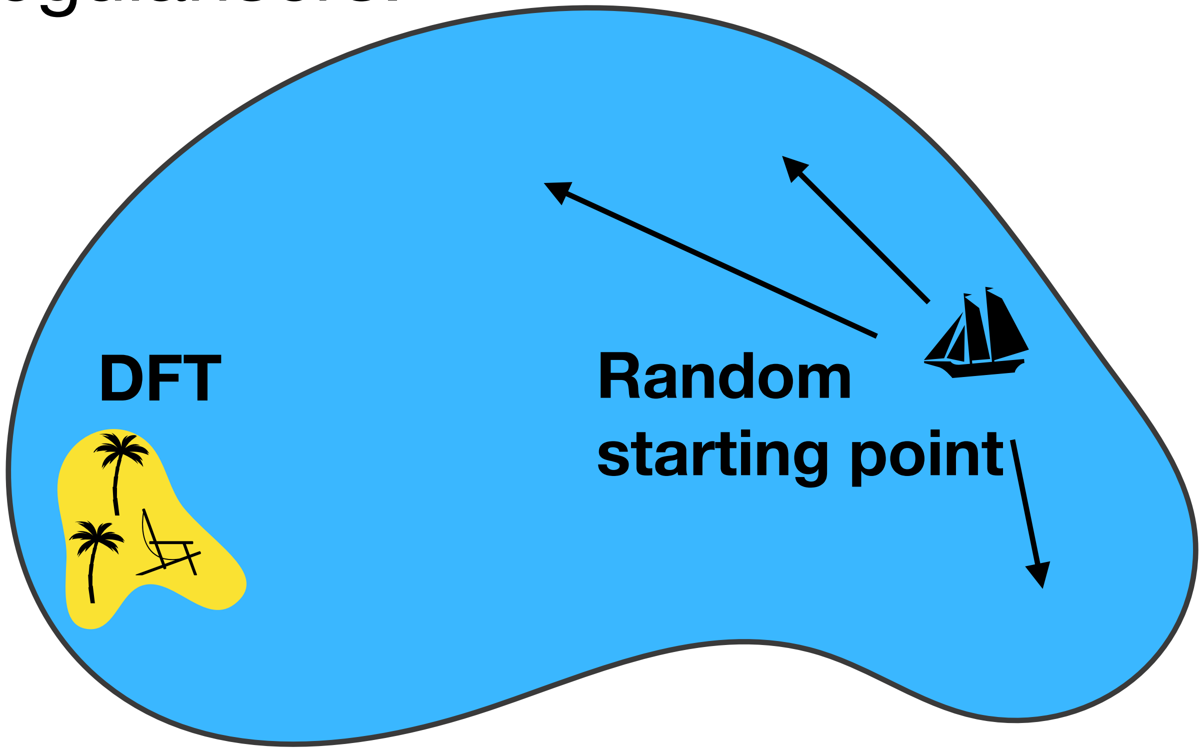
```
# ================================================
# Define simple 1-layer-CNN.
# ================================================

model = Sequential()
model.add(Conv1D(2, kernel_size=2,
                 activation='relu',
                 input_shape=(2*number_bins,1)))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
```

For this network classification works in momentum space, but not in position space.

Utilising dual representation

- Goal: improve performance on position space.
- Deeper network? Can do the job in principle
  [DFT can be implemented with a single dense layer]
- However finding it dynamically is `impossible' with standard optimisers, initialisations, and regularisers.

Layer	Shape	Parameters
Dense	(2000,2)	16000000
Conv1D	(2000,2)	4
Activation	(2000,2)	-
Dense	1	4001
Activation	1	-



DFT from modified loss

- Adapt loss function to achieve feature separation, i.e. separating the two classes of data (inspired by triplet loss) [towards generating dual representations dynamically]

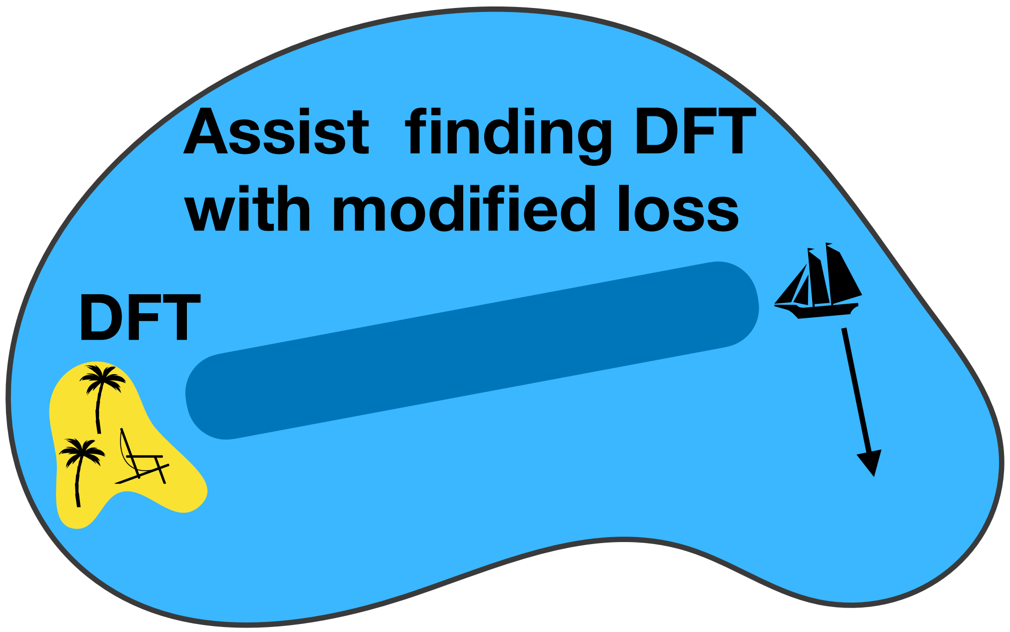
1503.03832

Separation

$$\text{Loss} = |y_{\text{noise}}|^2 - |y_{\text{signal}}|^2 + \alpha$$

Decorrelation via weight regularisation

$$\text{Loss} = \max\left\{0,\ \beta - \sum |w|^2\right\} + \sum_{i \neq j} \max\{0, (w_i \cdot w_j)\}$$



- Note: different data question (signal injected in position and momentum space) can lead to multiple minima in the loss landscape, i.e. using momentum space and position space.

DFT from modified loss

- Adapt loss function to achieve feature separation, i.e. separating the two classes of data (inspired by triplet loss) [towards generating dual representations dynamically]

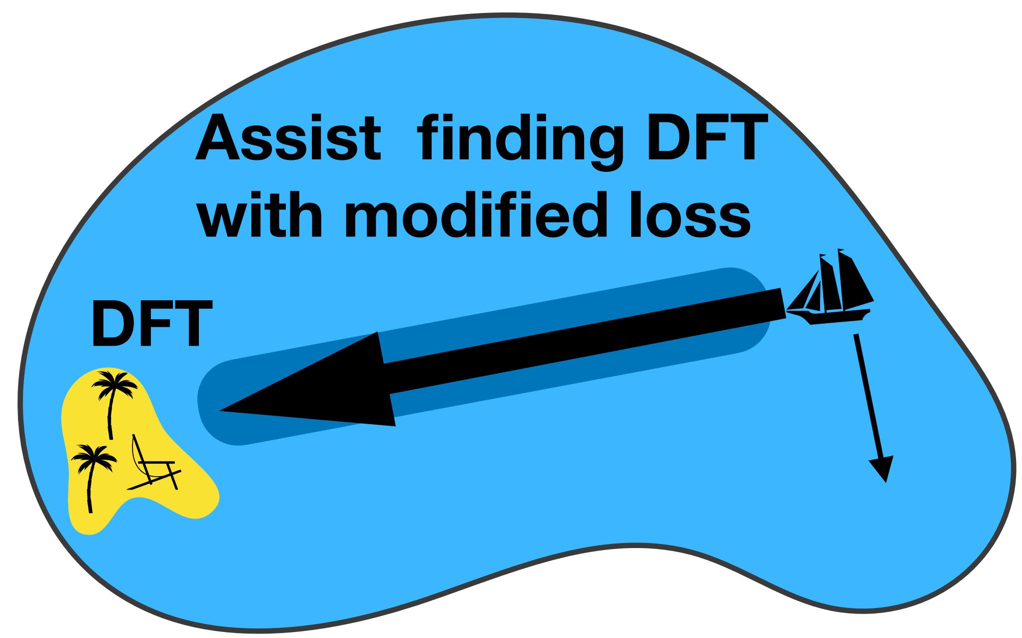
1503.03832

Separation

$$\text{Loss} = |y_{\text{noise}}|^2 - |y_{\text{signal}}|^2 + \alpha$$

Decorrelation via weight regularisation

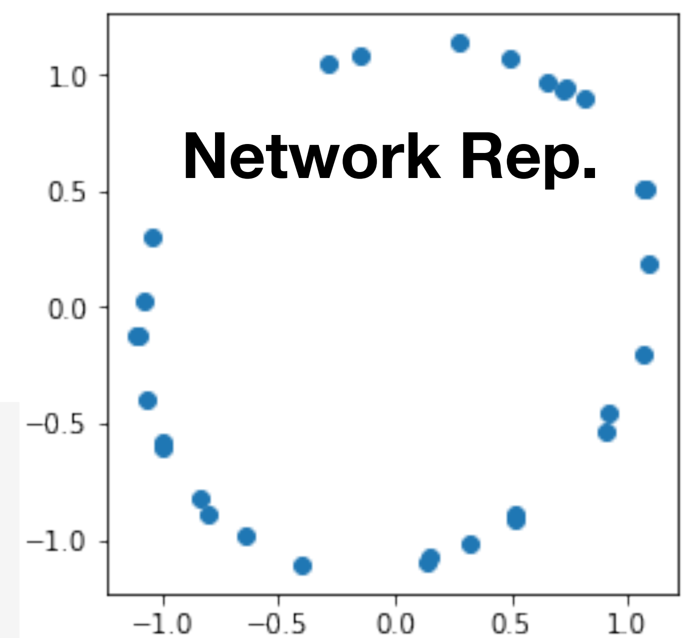
$$\text{Loss} = \max\left\{0,\ \beta - \sum |w|^2\right\} + \sum_{i \neq j} \max\{0, (w_i \cdot w_j)\}$$



- Note: different data question (signal injected in position and momentum space) can lead to multiple minima in the loss landscape, i.e. using momentum space and position space.

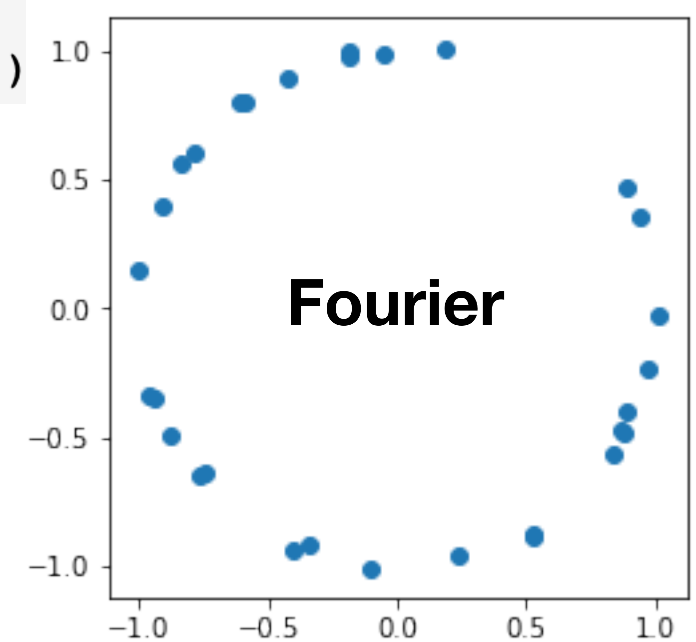
DFT from feature separation

- The final result is a simple network with few parameters utilising `dual' representation.

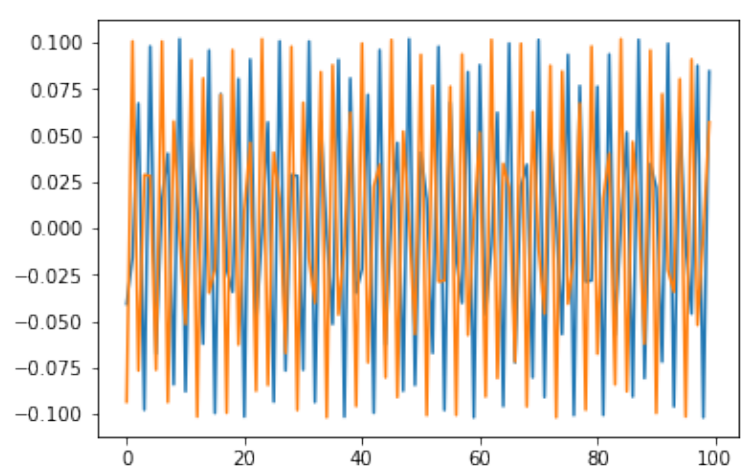



```
ourmodel=Sequential()
ourmodel.add(Dense(2,use_bias='False',activity_regularizer=customReg,
                   input_shape=(int(2*bins),)))
```

```
def customLoss(yTrue,yPred):
    margin = K.constant(1.)
    return K.maximum(K.constant(0),margin-K.sum(K.square(yPred)))

def customReg(weight_matrix):
    margin = K.constant(0.5)
    return K.maximum(K.constant(0.),margin-K.sum(K.square(weight_matrix)))
```

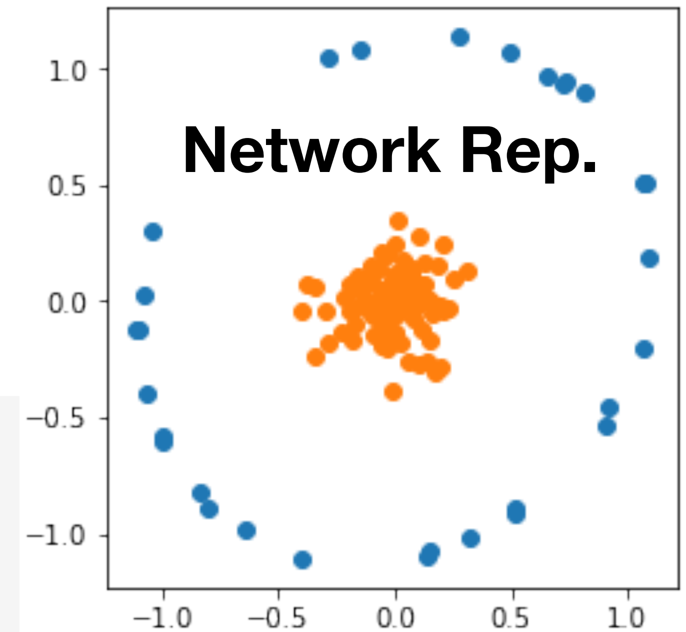


Input sample:



DFT from feature separation

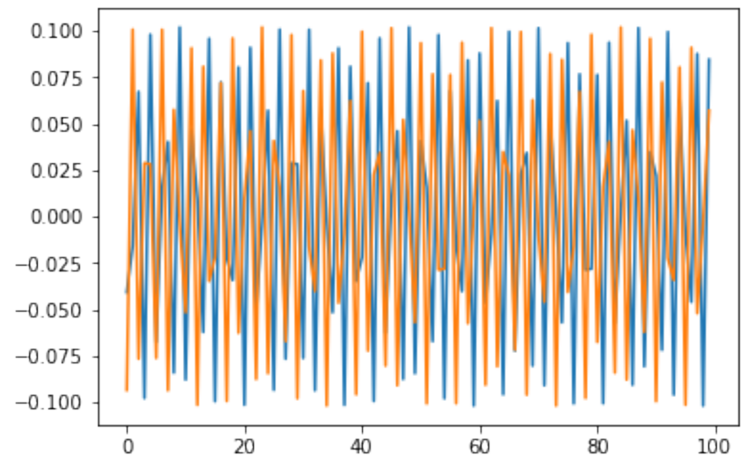
- The final result is a simple network with few parameters utilising `dual' representation.

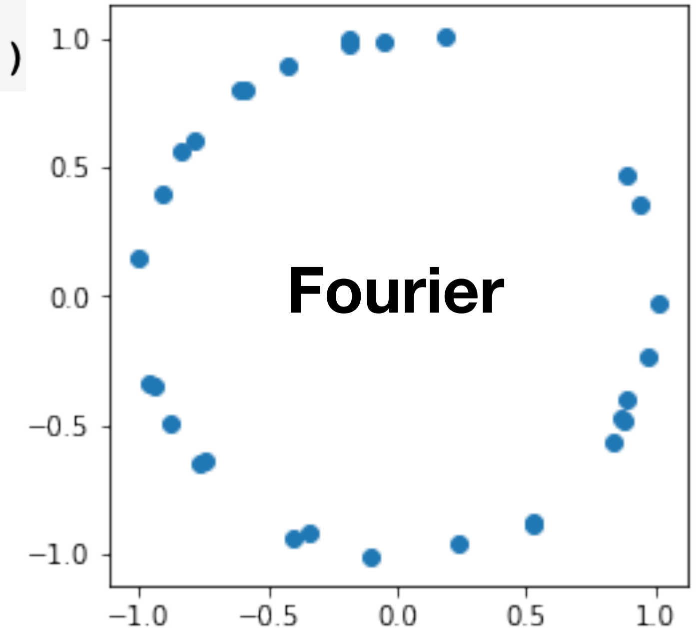



```
ourmodel=Sequential()
ourmodel.add(Dense(2,use_bias='False',activity_regularizer=customReg,
                   input_shape=(int(2*bins),)))
```

```
def customLoss(yTrue,yPred):
    margin = K.constant(1.)
    return K.maximum(K.constant(0),margin-K.sum(K.square(yPred)))

def customReg(weight_matrix):
    margin = K.constant(0.5)
    return K.maximum(K.constant(0.),margin-K.sum(K.square(weight_matrix)))
```

Input sample:





Dualities and Fourier transformation

- Review on dualities and global symmetries (Quevedo: hep-th/9706210), several dualities can be seen as Fourier transformation:

> "Therefore we can see that the dualities we have been dealing with for antisymmetric tensors are only particular cases of Fourier transforms and finding the dual action reduces to finding Fourier transforms. "

- For instance duality between massive antisymmetric tensors.

$$Z = \int \mathscr{D}\tilde{B}_{d-h}\mathscr{D}H_h \; e^{\int d^D x (H\cdot d\tilde{B}_{d-h} + G(H_h) + \tilde{F}(\tilde{B}_{d-h}))}$$

$$e^{\tilde{F}} \overset{\text{FT}}{\leftrightarrow} e^{F}$$

$\tilde{B}_{d-h}$ ↙ ↘ H_h

$$S = \int d^D x \left(F(\partial H_h) + G(H_h) \right)$$

$$\tilde{S} = \int d^D x \left(\tilde{F}(\tilde{B}_{d-h}) + \tilde{G}(\partial \tilde{B}_{d-h}) \right)$$

Dualities and Fourier transformation

- Fourier transformation ~ Duality transformation

  “Therefore we can see that the dualities we have been dealing with for antisymmetric tensors are only particular cases of Fourier transforms and **finding the dual action reduces to finding Fourier transforms.** “

- Here: network adapts dual representation by demanding feature separation.

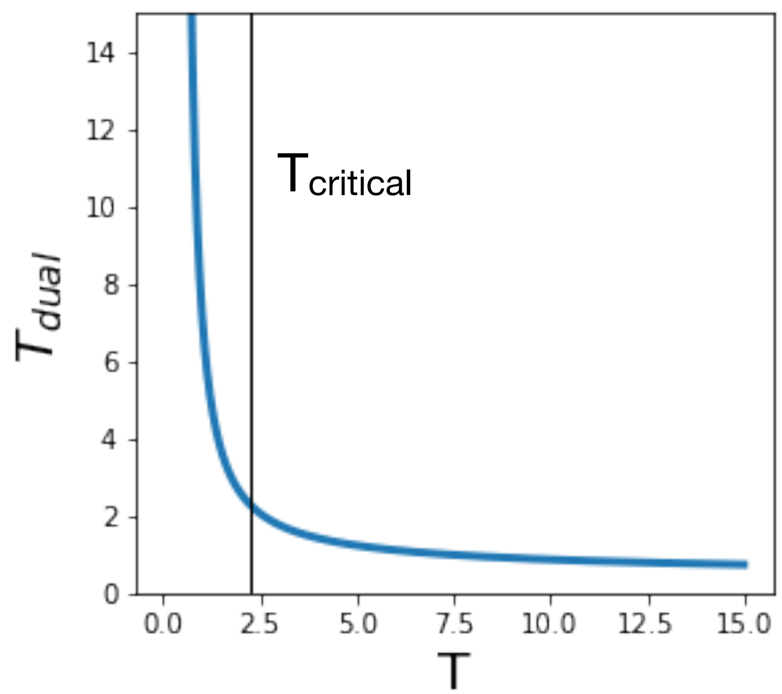
- Can we use this?

Let’s look for Physics examples

2D Ising Model

Duality in 2D Ising model

- High - low temperature self-duality



Ordered rep. ↔ Disordered rep.

Original

$$H = -J \sum_{\langle i,j \rangle} s_i s_j$$

$$Z = \sum e^{-\beta H(s)}$$

$$\beta = \frac{1}{k_B T}$$

Dual

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$$

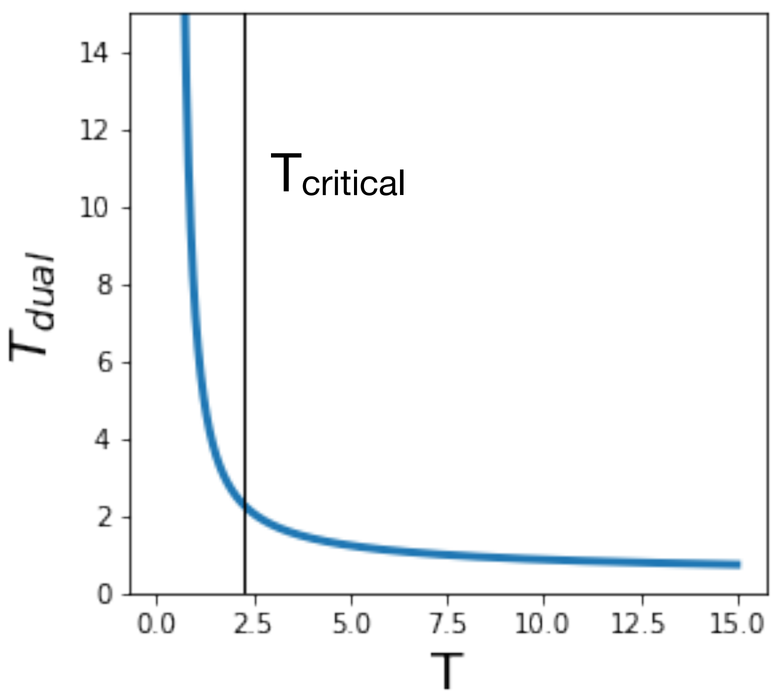
$$Z = \sum e^{-\tilde{\beta} H(\sigma)}$$

$$\tilde{\beta} = -\frac{1}{2} \log \tanh \beta$$

Krammers, Wannier 1941; Onsager 1943; review: Savit 1980

Duality in 2D Ising model

- High - low temperature self-duality



Ordered rep. ↔ Disordered rep.

Position space?

Momentum space?

Original

$$H = -J \sum_{\langle i,j \rangle} s_i s_j$$

$$Z = \sum e^{-\beta H(s)}$$

$$\beta = \frac{1}{k_B T}$$

Dual

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$$

$$Z = \sum e^{-\tilde{\beta} H(\sigma)}$$

$$\tilde{\beta} = -\frac{1}{2} \log \tanh \beta$$

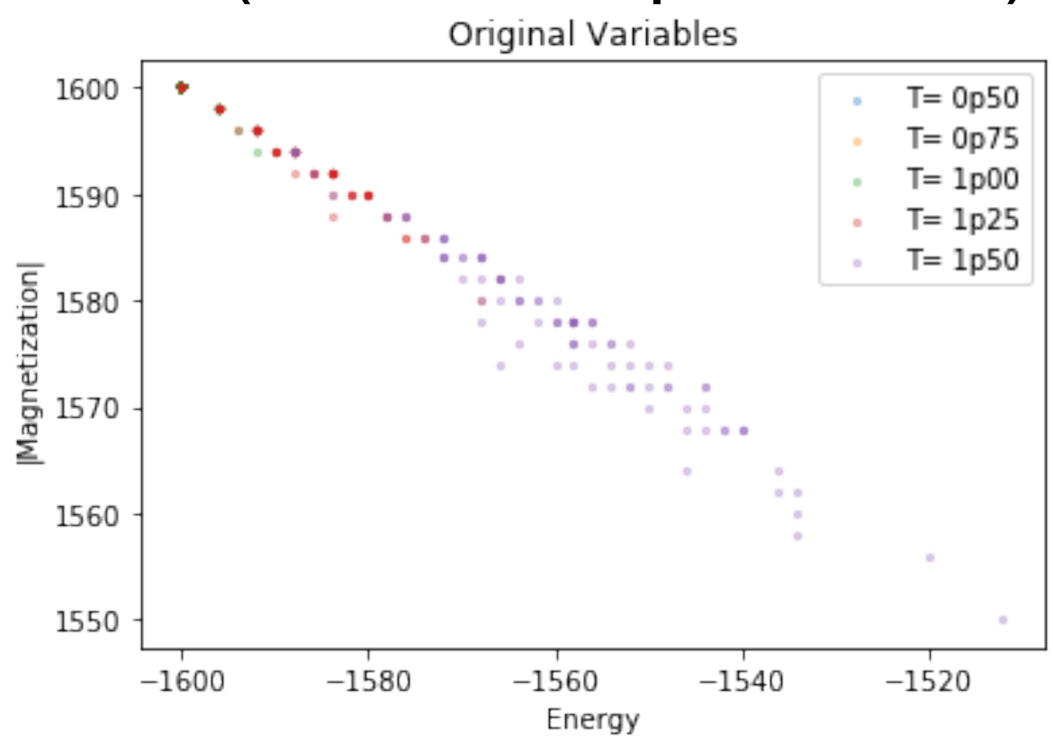
Krammers, Wannier 1941; Onsager 1943; review: Savit 1980

Which data problem?

- Some correlation function which is easier evaluated on dual variables.

$$\langle \sigma_i \sigma_j \rangle, \ \langle E(\sigma) \rangle, \ \langle M(\sigma) \rangle$$

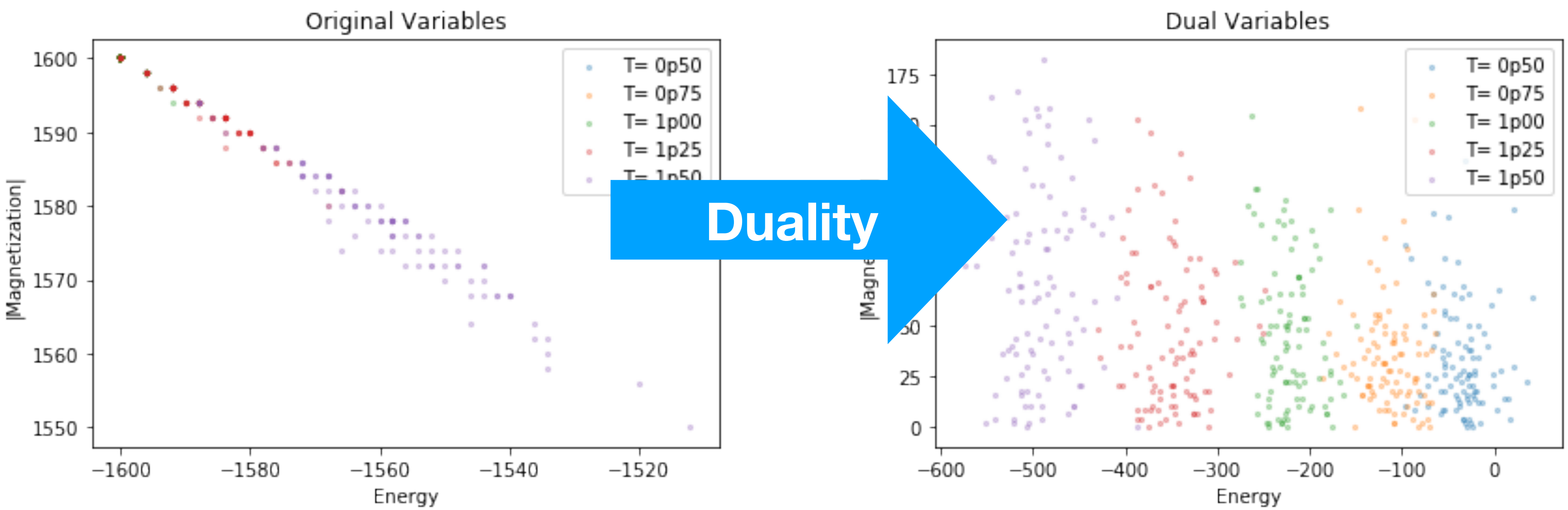
- Can we classify the temperature for low-temperature configurations? Which temperature is a sample drawn from (at low temperatures)?



They look rather similar. How about in the dual rep.?

Data question on Ising

- But at the dual temperatures, our data takes a different shape:

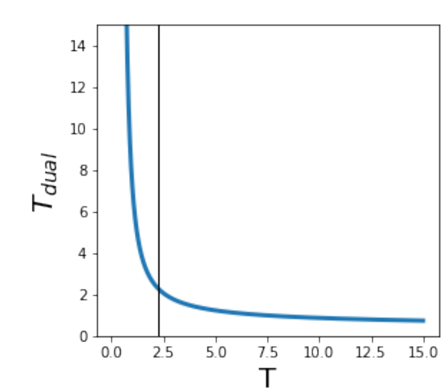


- It is easier to classify temperature of a low-temperature configuration in the dual representation …

- How come?

$$P(\mathbf{s}) = \frac{e^{E/T}}{Z}, \qquad P(\sigma) = \frac{e^{\tilde{E}/\tilde{T}}}{Z}$$

$$\langle \Delta E \rangle \ll \langle \Delta \tilde{E} \rangle \qquad \Delta T \ll \Delta \tilde{T}$$



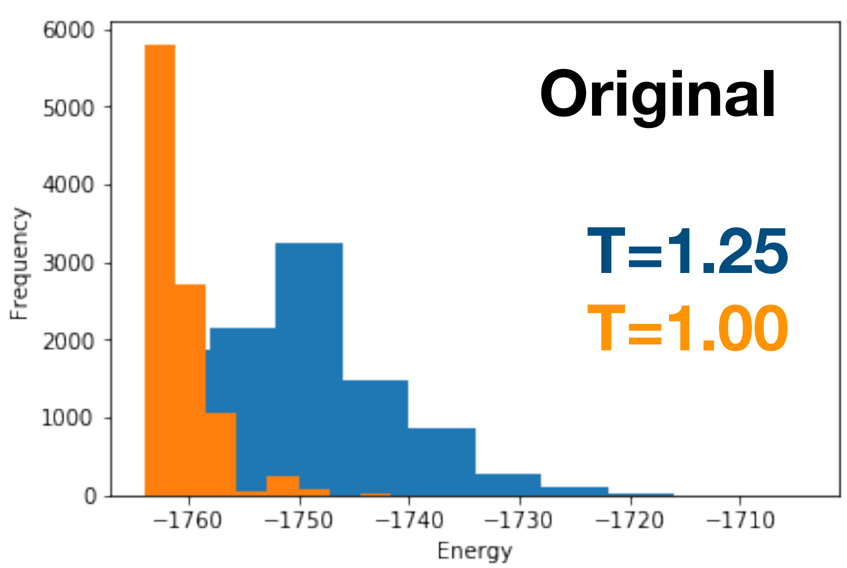
Ising: simple network

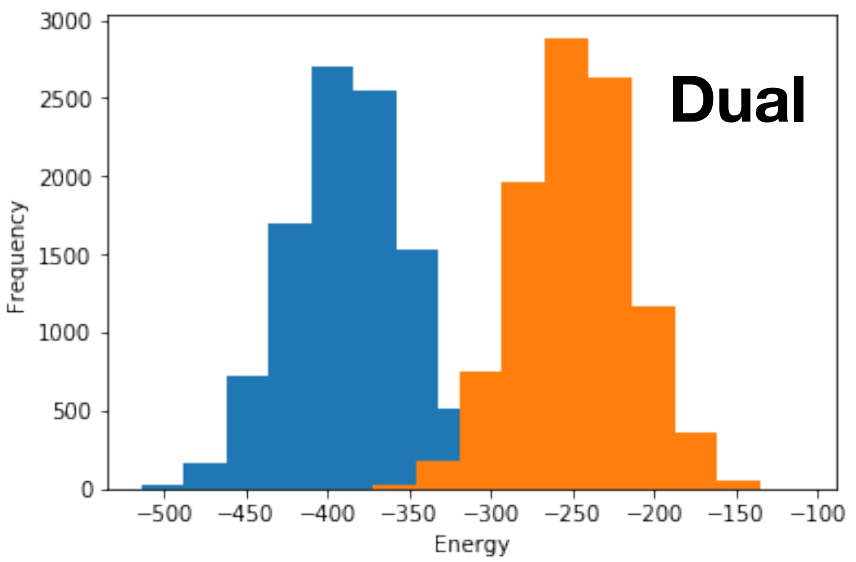
- Let's confirm this at simple networks:

```
model = Sequential()
model.add(Conv2D(4, kernel_size=3,
                 activation='relu',
                 input_shape=(42,42,1)))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))

opt = Nadam(lr=0.002)

model.compile(loss=binary_crossentropy,
              optimizer=opt, metrics=['accuracy'])
```

Original data: <83% val. acc.

Dual data: ~96% val. acc.





Side remark: way outperforming standard sklearn classifiers

How do we utilise this data representation?

Need duality transformation

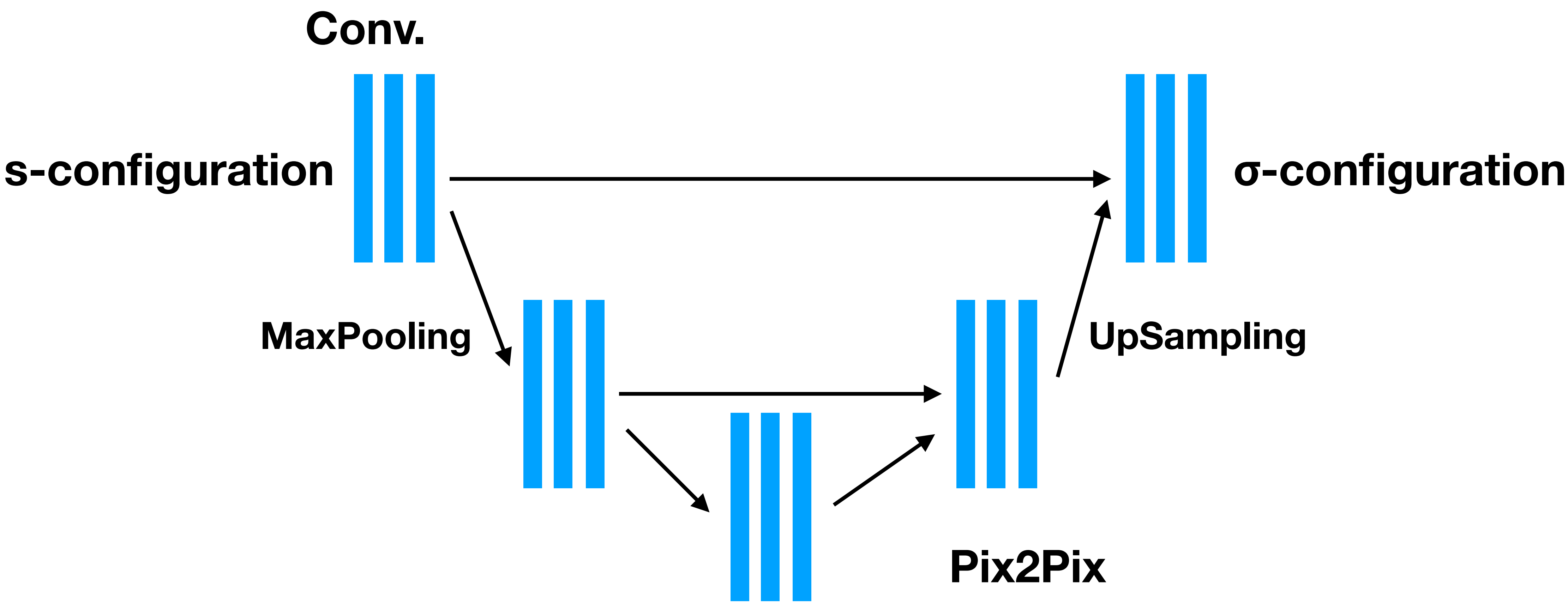
Ising: finding duality transformation

- In analogy to DFT, we want to go to dual representation.

- Which strategy?
- First route: Forcing feature separation by adding similar loss as before.
- Second route: KL-divergence (or categorical-crossentropy) between NN distribution and dual temperature distribution; based on p_{Ei} (sparse samples)

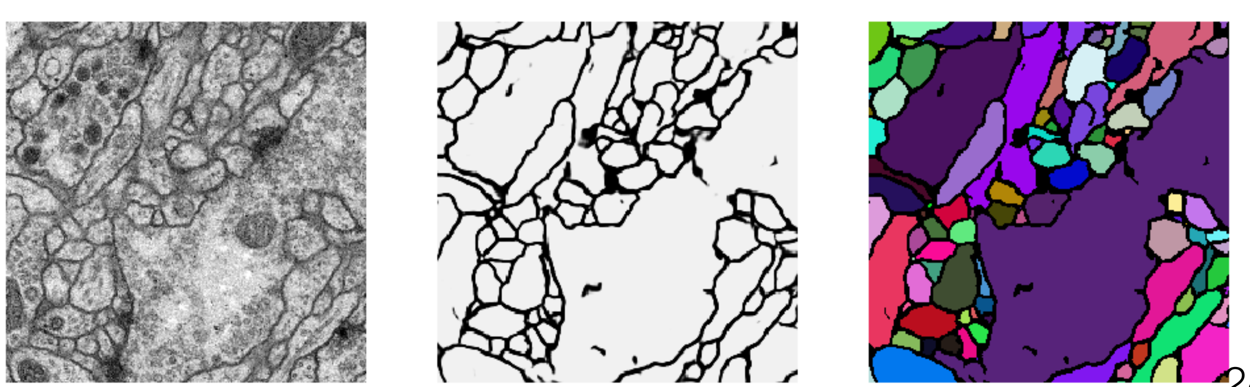
Feature separation in Ising

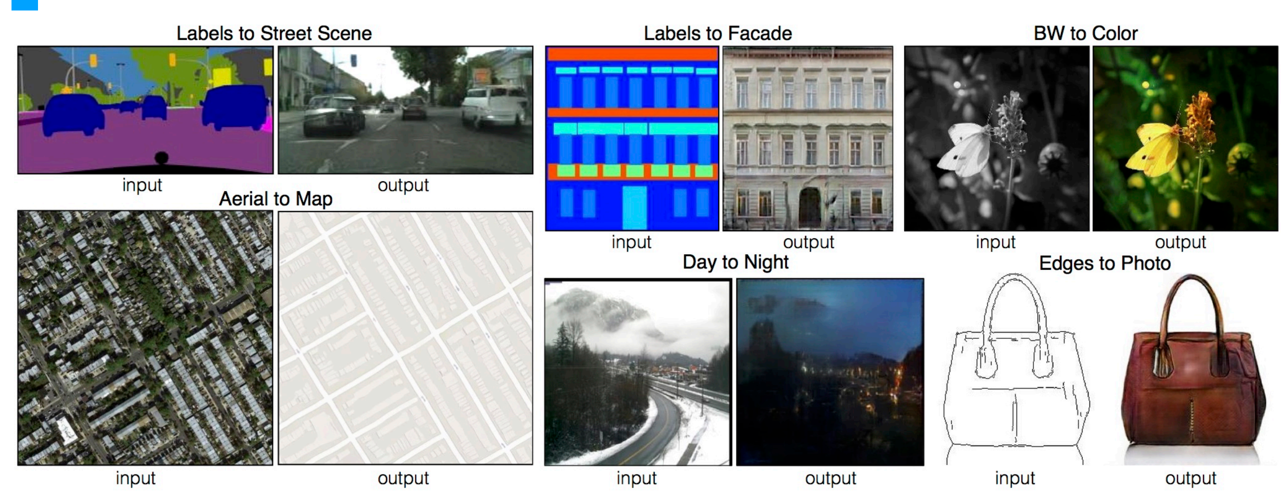
- Which neural network architecture? Several architectures, so far most promising: U-Net (1505.04597)



Pix2Pix

UNet





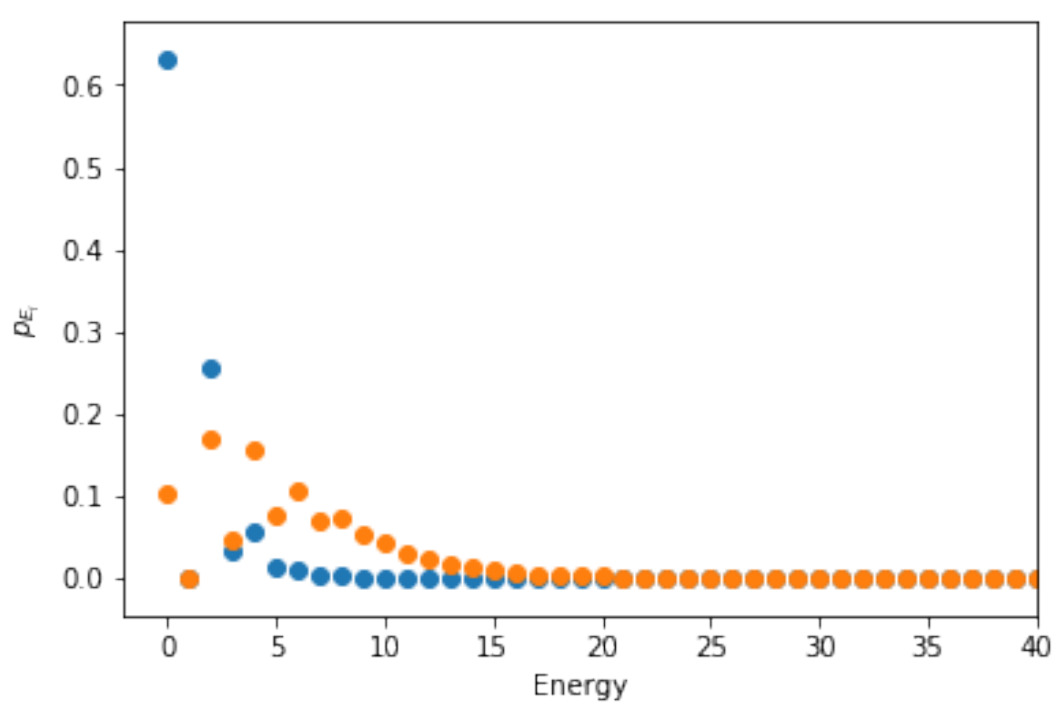
Matching p_{E_i}-distributions

- A “more sophisticated anchor” point…

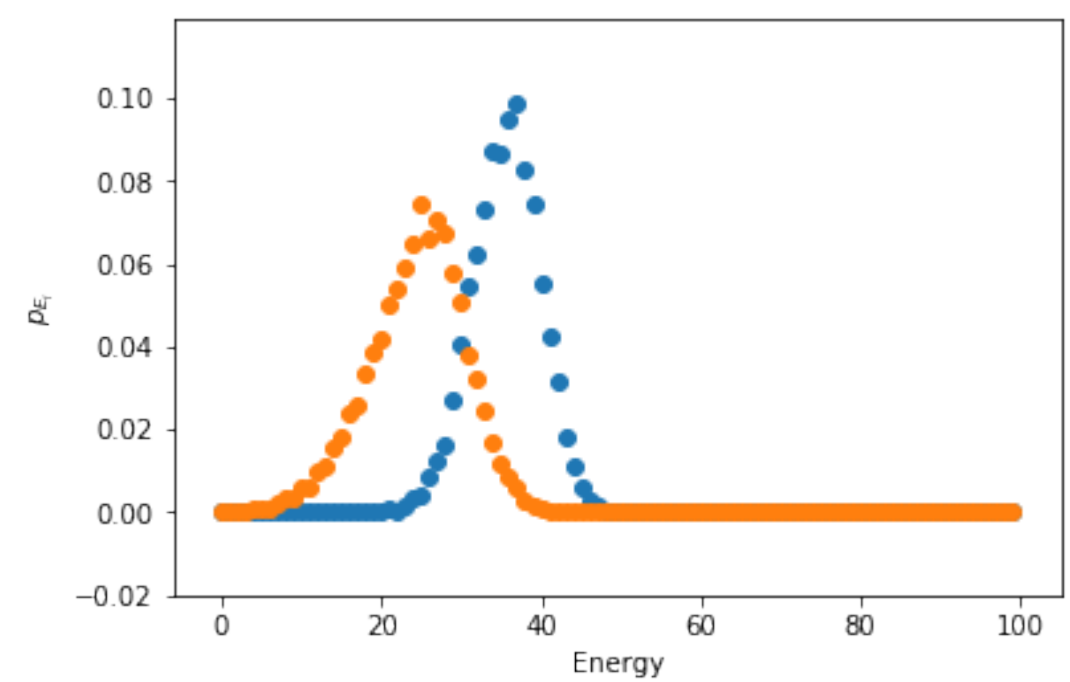
$$D_{KL}(P(E(f(s_i,\beta)))\,||\,P(E(\sigma_i,\beta))) = \sum_j P(E_j(f(s_i,\beta))\log\left(\frac{P(E_j(f(s_i,\beta)))}{P(E_j(\sigma_i,\tilde{\beta}))}\right)$$

- Network should transform from input to output distributions

Input Distributions



Dual Temperature distributions



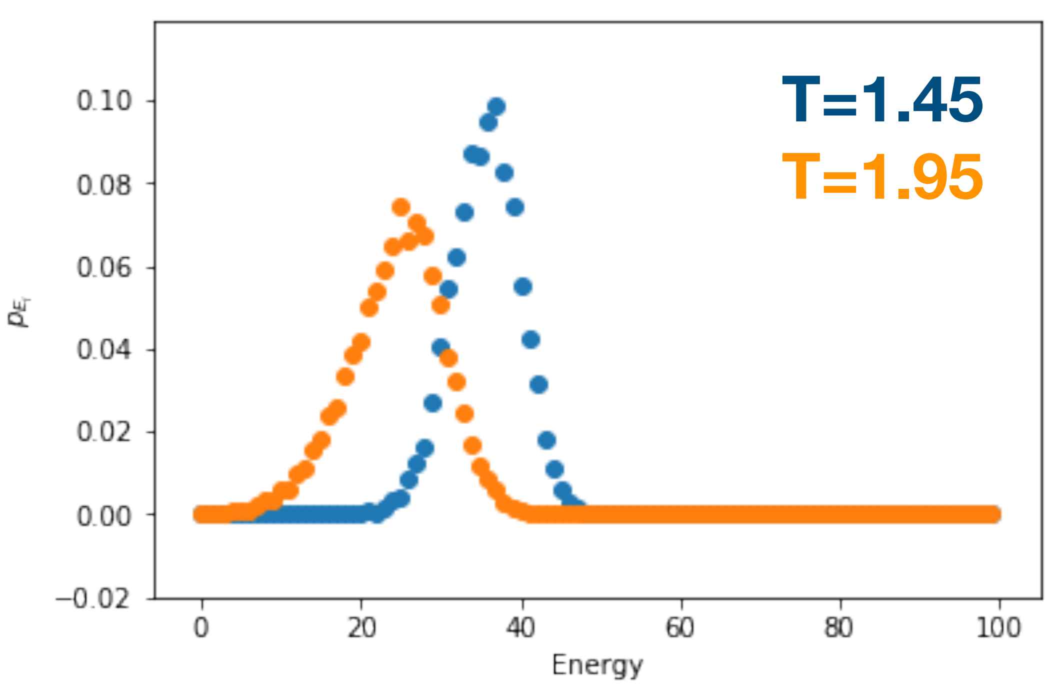
Matching p_{E_i}-distributions

- A "more sophisticated anchor" point…

$$D_{KL}(P(E(f(s_i,\beta)))\,||\,P(E(\sigma_i,\beta))) = \sum_j P(E_j(f(s_i,\beta))\log\left(\frac{P(E_j(f(s_i,\beta)))}{P(E_j(\sigma_i,\tilde{\beta}))}\right)$$

- Currently, we obtain the following dual distributions:

Dual Temperature distributions



Output distributions from network

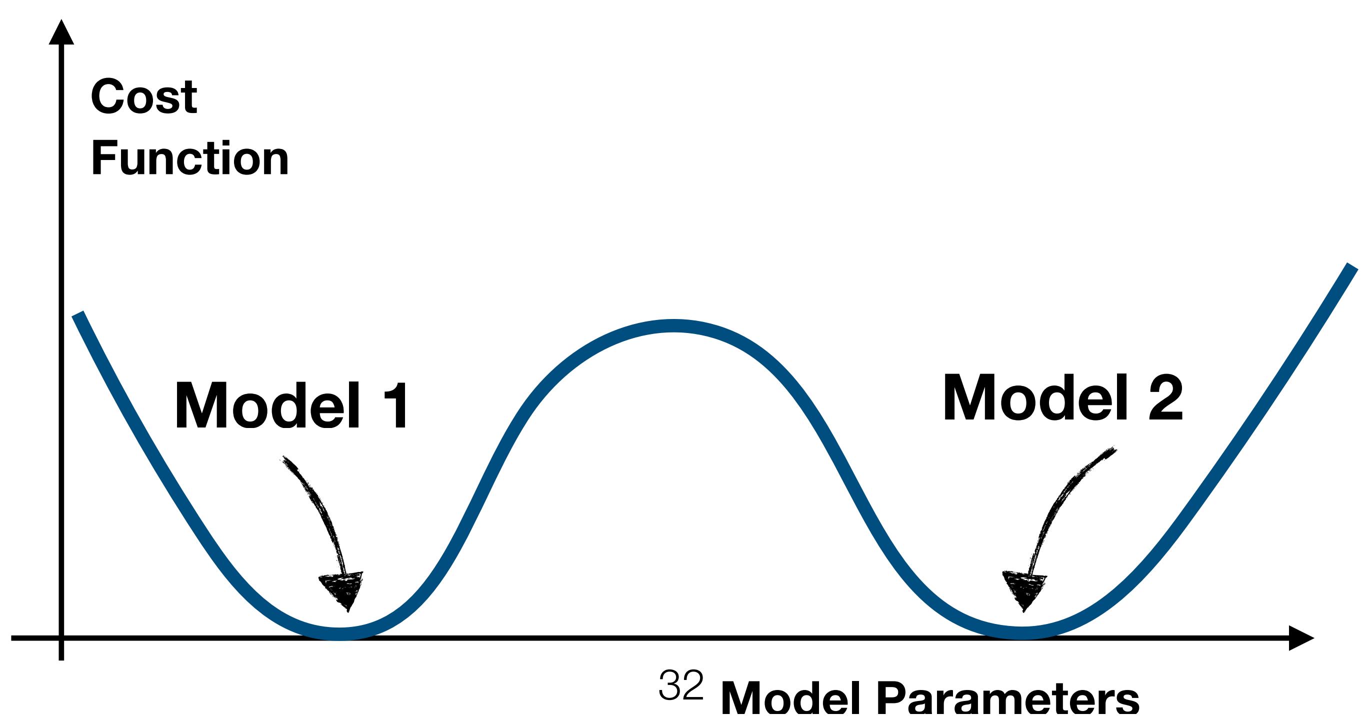
Preliminary, wait for publication

Further Dualities

Connection to String Phenomenology

Stringpheno $\leftrightarrow$ ML-Dualities

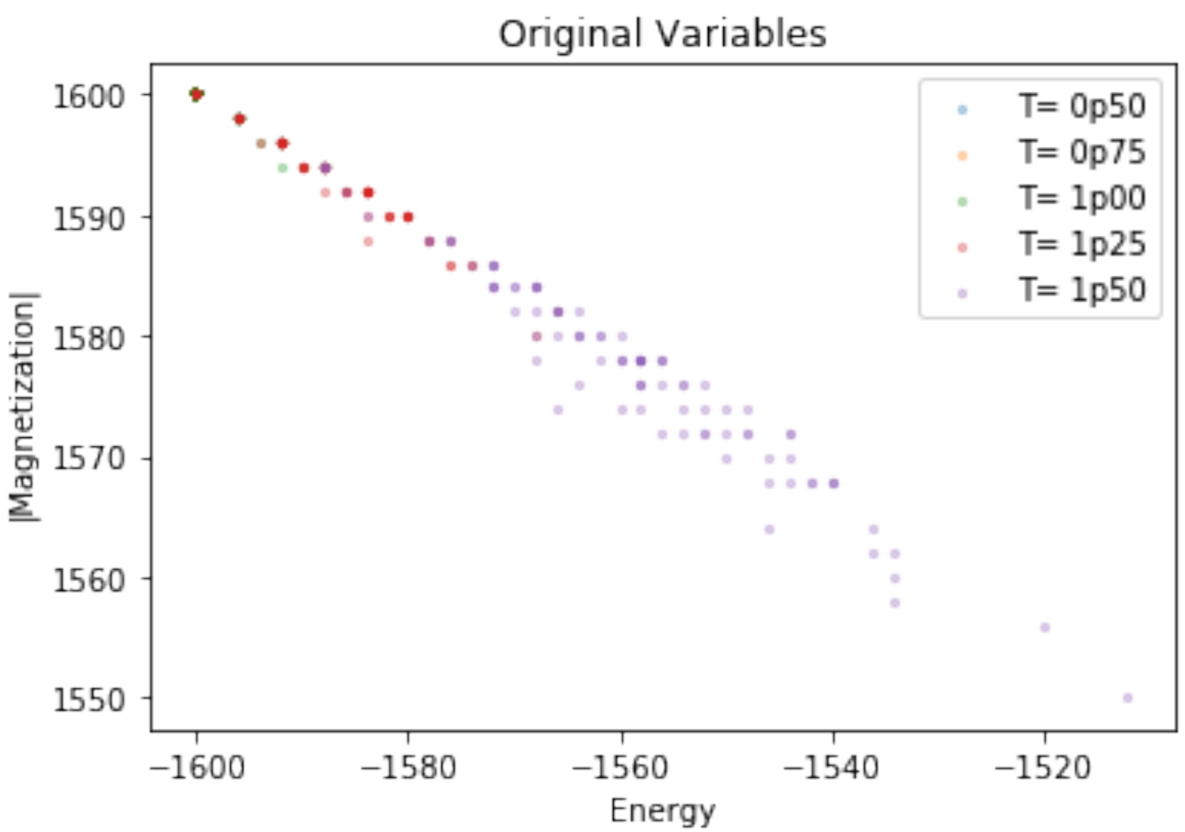
- Just some comments/thoughts:
- We need to improve on obtaining/recovering dualities: stringpheno perfect playground to obtain dualities and put dualities to use in data questions (couplings, higher order EFT operators)
- String landscape constrained by dualities, does this apply to neural networks as well?

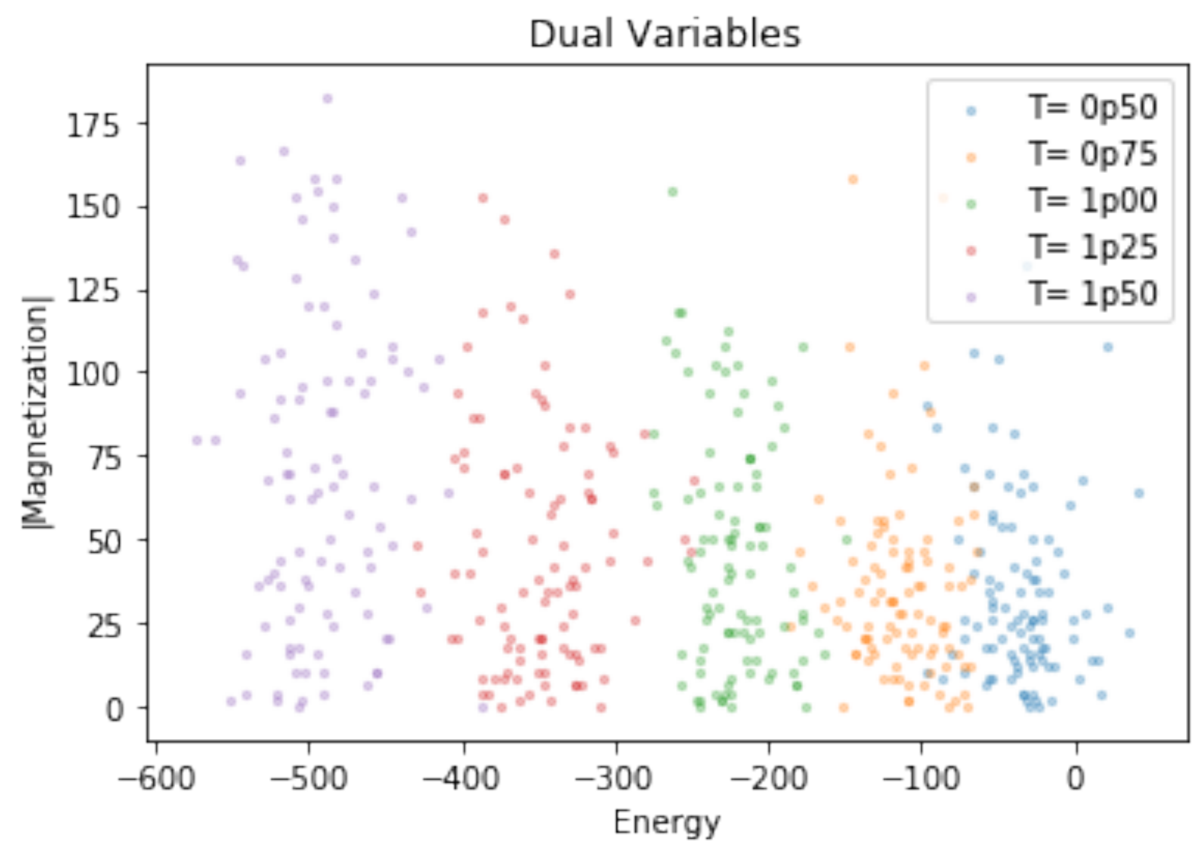


Conclusions

- Dualities tightly connected to data representations.
- Finding good data representations is at the heart of ML and often no interpretation is available.
- Favoured representation from asking data question (DFT, Ising)
- Dualities can emerge and identified from `feature separation’
- By using dualities/dual representations we can build more efficient networks (DFT, 2D Ising)
- Dualities in Physics motivate multiple minima in a different landscape, those of the cost functions of neural networks.

Thank you …


Original Variables
T= 0p50
T= 0p75
T= 1p00
T= 1p25
T= 1p50
|Magnetization|
Energy
1600
1590
1580
1570
1560
1550
−1600
−1580
−1560
−1540
−1520


Dual Variables
T= 0p50
T= 0p75
T= 1p00
T= 1p25
T= 1p50
|Magnetization|
Energy
175
150
125
100
75
50
25
0
−600
−500
−400
−300
−200
−100
0