Quantum field-theoretic machine learning

Dimitrios Bachtis

Joint work with Profs. Gert Aarts and Biagio Lucini.

Can we view machine learning as part of quantum field theory?

And why?

Probability distribution

A probability distribution is a product of strictly positive and appropriately normalized factors (or potential functions) ψ:

$$p(\phi) = \frac{\prod_{c \in C} \psi_c(\phi)}{\int_{\boldsymbol{\phi}} \prod_{c \in C} \psi_c(\boldsymbol{\phi}) d\boldsymbol{\phi}},$$

Probability distribution

A probability distribution is a product of strictly positive and appropriately normalized factors (or potential functions) ψ:

$$p(\phi) = \frac{\prod_{c \in C} \psi_c(\phi)}{\int_{\boldsymbol{\phi}} \prod_{c \in C} \psi_c(\boldsymbol{\phi}) d\boldsymbol{\phi}},$$

1. Factors are the fundamental building blocks of probability distributions.

Probability distribution

A probability distribution is a product of strictly positive and appropriately normalized factors (or potential functions) ψ:

$$p(\phi) = \frac{\prod_{c \in C} \psi_c(\phi)}{\int_{\boldsymbol{\phi}} \prod_{c \in C} \psi_c(\boldsymbol{\phi}) d\boldsymbol{\phi}},$$

1. Factors are the fundamental building blocks of probability distributions.

2. By controlling the factors we are able to control the form of the probability distribution.

Representation

We require some form of representation to construct the probability distribution. We are going to use a finite set Λ that we express as a graph $\mathcal{G}(\Lambda, e)$ where e is the set of edges in $\mathcal{G}$.

A clique c is a subset of Λ where the points are pairwise connected. A maximal clique is a clique where we cannot add another point that is pairwise connected with ***all*** the points in the subset.

Representation

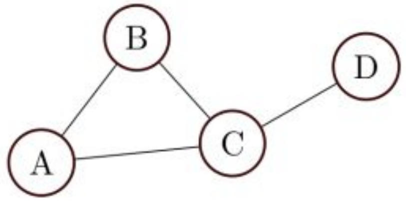
We require some form of representation to construct the probability distribution. We are going to use a finite set Λ that we express as a graph $\mathcal{G}(\Lambda, e)$ where e is the set of edges in $\mathcal{G}$.

A clique c is a subset of Λ where the points are pairwise connected. A maximal clique is a clique where we cannot add another point that is pairwise connected with ***all*** the points in the subset.

There are only two maximal cliques, the subsets {A,B,C} and {C,D}.

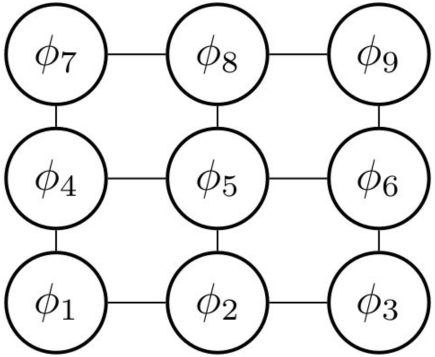
{A,B} is a clique but it is **not** maximal because another point (C) can be included that is pairwise connected with both {A,B}.

{A,B,C,D} is **not** a clique and **not** maximal because (D) is not pairwise connected with all other points (and vice versa).



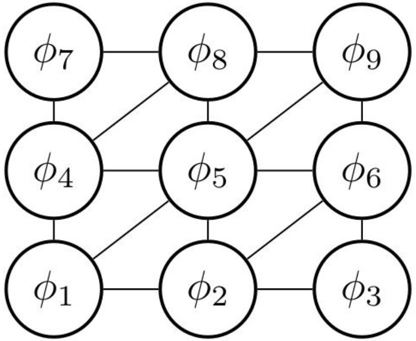
Representation

On the square lattice a maximal clique is a link.



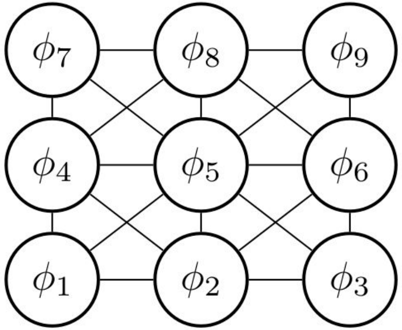
(a)

On a triangular lattice a maximal clique is a triangle.

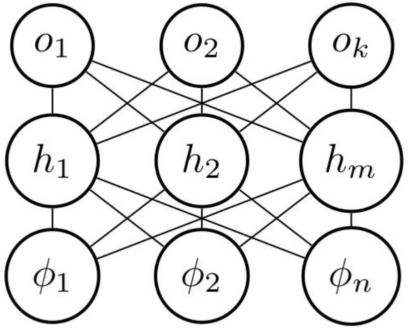


(b)

On the square lattice with both diagonals a maximal clique is a square.



(c)



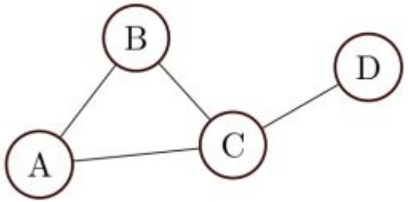
(d)

On the bipartite graph, which represents conventional neural network architectures a maximal clique is a link.

Representation

Given a graph $\mathcal{G}(\Lambda, e)$, the random variables φ_i at each point i define a Markov random field if they fulfill the local Markov property with respect to $\mathcal{G}$.

The local Markov property denotes that a random variable φ_i depends only on its neighbors and it is conditionally independent of all other random variables in the set:



$$p(\phi_i | (\phi_j)_{j \in \Lambda - \phi_i}) = p(\phi_i | (\phi_j)_{j \in n_i}).$$

Representation

Hammersley-Clifford theorem

A strictly positive distribution p satisfies the local Markov property of an undirected graph $\mathcal{G}$, if and only if p can be represented as a product of strictly positive potential functions ψ_c over $\mathcal{G}$, one per maximal clique c, i.e.

$$p(\phi) = \frac{1}{Z} \prod_{c \in C} \psi_c(\phi), \quad Z = \int_{\boldsymbol{\phi}} \prod_{c \in C} \psi_c(\boldsymbol{\phi}) d\boldsymbol{\phi}$$

where Z is the partition function and $\boldsymbol{\varphi}$ are all possible states of the system.

Representation

There are two different directions to pursue:

1. We can devise potential functions that satisfy the Hammersley-Clifford theorem to construct a Markov random field.

Representation

There are two different directions to pursue:

1. We can devise potential functions that satisfy the Hammersley-Clifford theorem to construct a Markov random field.
2. We can evaluate if known physical systems can be recast within this mathematical framework by verifying instead if they satisfy the theorem.

We will pursue the second direction.

Representation

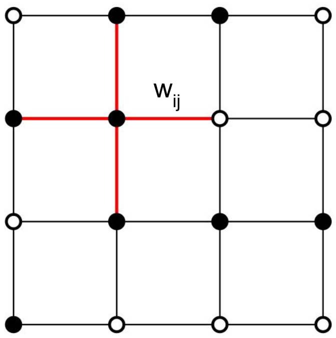
2d φ^4 theory:

$$\mathcal{L}_E = \frac{\kappa}{2}(\nabla\phi)^2 + \frac{\mu_0^2}{2}\phi^2 + \frac{\lambda}{4}\phi^4,$$

$$S_E = -\kappa_L \sum_{\langle ij\rangle} \phi_i\phi_j + \frac{(\mu_L^2 + 4\kappa_L)}{2}\sum_i \phi_i^2 + \frac{\lambda_L}{4}\sum_i \phi_i^4.$$

k_L, μ_L, λ_L dimensionless parameters

$$w = \kappa_L,\ a = (\mu_L^2 + 4\kappa_L)/2,\ b = \lambda_L/4$$

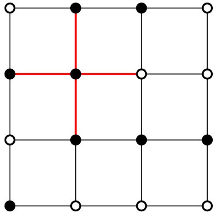


Inhomogeneous φ^4 theory:

$$S(\phi;\theta) = -\sum_{\langle ij\rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

Representation

The φ^4 theory is formulated on a square lattice which is equivalent to a graph $\mathcal{G}(\Lambda,e)$ where Λ is the set of lattice sites and e the links. A non-unique choice of potential function per each maximal clique is:



$$\psi_c = \exp\left[-w_{ij}\phi_i\phi_j + \frac{1}{4}(a_i\phi_i^2 + a_j\phi_j^2 + b_i\phi_i^4 + b_j\phi_j^4)\right],$$

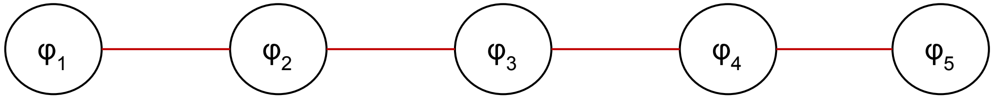
The probability distribution is expressed as a product of strictly positive potential functions ψ, over each maximal clique:

$$p(\phi;\theta) = \frac{\exp\left[\sum_{c\in C}\ln\psi_c(\phi)\right]}{\int_\phi \exp\left[\sum_{c\in C}\ln\psi_c(\phi)\right]d\phi} = \frac{1}{Z}\prod_{c\in C}\psi_c(\phi).$$

The φ^4 theory satisfies Markov properties and it is therefore a Markov random field.

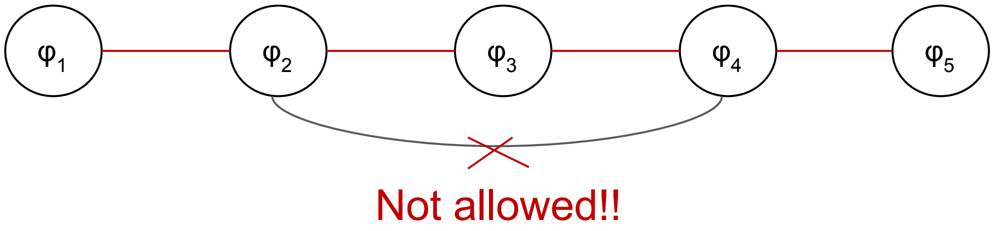
Representation

The Markov property in a Markov chain



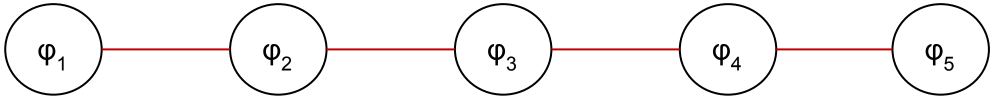
Representation

The Markov property in a Markov chain

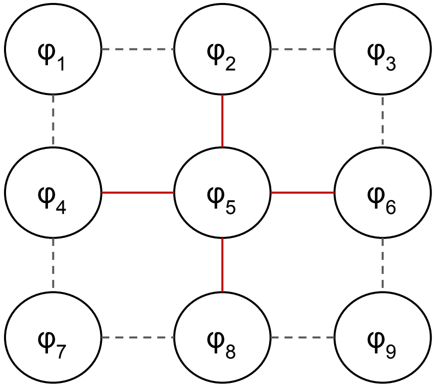


Representation

The Markov property in a Markov chain



A Markov random field satisfies the Markov property in high-dimensions



Learning

Having established that certain physical systems are Markov random fields, how do we use them for machine learning?

Learning

Having established that certain physical systems are Markov random fields, how do we use them for machine learning?

Exactly in the same way as any other machine learning algorithm...

Learning

The φ^4 theory has a probability distribution p(φ;θ) with action S(φ;θ):

$$p(\phi;\theta) = \frac{\exp\left[-S(\phi;\theta)\right]}{\int_{\boldsymbol{\phi}} \exp[-S(\boldsymbol{\phi},\theta)]d\boldsymbol{\phi}}.$$

We now consider a different statistical system or quantum field theory with action or Hamiltonian A and a target probability distribution q(φ):

$$q(\phi) = \exp[-\mathcal{A}]/Z_{\mathcal{A}}$$

Learning

We can then define an asymmetric distance between the probability distributions p(φ;θ) and q(φ), which is called the Kullback-Leibler divergence:

$$KL(p||q) = \int_{-\infty}^{\infty} p(\phi;\theta) \ln \frac{p(\phi;\theta)}{q(\phi)} d\phi \geq 0.$$

Learning

We can then define an asymmetric distance between the probability distributions p(φ;θ) and q(φ), which is called the Kullback-Leibler divergence:

$$KL(p||q) = \int_{-\infty}^{\infty} p(\phi;\theta) \ln \frac{p(\phi;\theta)}{q(\phi)} d\phi \geq 0.$$

We want to minimize the Kullback-Leibler divergence.

By minimizing it we would make the two probability distributions equal. We can then use the probability distribution p(φ;θ) to draw samples from the target distribution q(φ). In essence after the minimization we have created a mapping between the probability distributions of the two systems.

Learning

We now substitute the two probability distributions in the Kullback-Leibler divergence to obtain:

$$F_{\mathcal{A}} \leq \langle \mathcal{A} - S \rangle_{p(\phi;\theta)} + F \equiv \mathcal{F},$$

The expression above sets a rigorous upper bound to the calculation of the free energy of the system with action A and this calculation is conducted entirely on samples drawn from the distribution p(φ;θ) of the φ^4 theory.

Learning

To minimize the variational free energy we implement a gradient-based approach:

$$\frac{\partial \mathcal{F}}{\partial \theta_i} = \langle \mathcal{A} \rangle \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle - \left\langle \mathcal{A} \frac{\partial S}{\partial \theta_i} \right\rangle + \left\langle S \frac{\partial S}{\partial \theta_i} \right\rangle - \langle S \rangle \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle,$$

We then update the coupling constants θ at each step t until convergence.

$$\theta^{(t+1)} = \theta^{(t)} - \eta * \mathcal{L}, \qquad \mathcal{L} = \partial \mathcal{F} / \partial \theta^{(t)}$$

After training we expect that, practically:

$$\mathcal{F} \approx F_A \qquad p(\phi; \theta) \approx q(\phi).$$

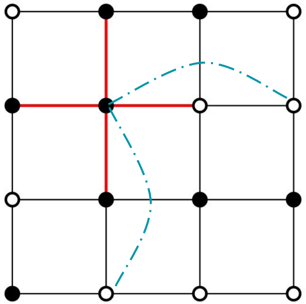
Learning

A proof-of-principle demonstration is to use the inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

to learn an action that includes longer-range interactions:

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij \rangle}\phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} \phi_i\phi_j$$



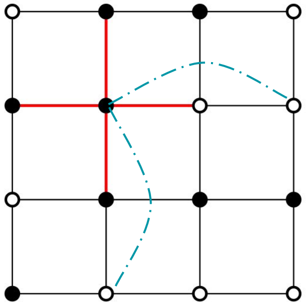
Learning

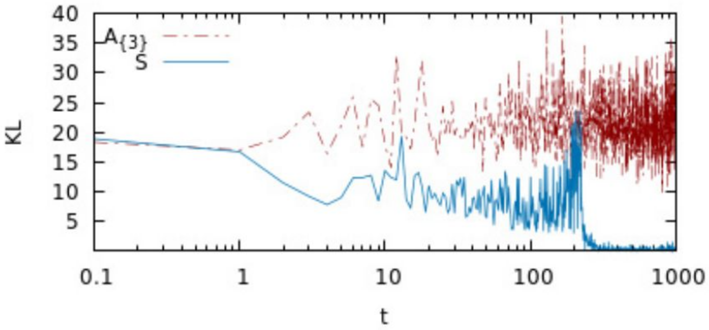
A proof-of-principle demonstration is to use the inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

to learn an action that includes longer-range interactions:

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij \rangle} \phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} \phi_i\phi_j$$





Learning

What does the approximate equality of two probability distributions imply?

$$p(\phi; \theta) \approx q(\phi),$$

We anticipate that observables calculated under the ensembles will also be approximately equal.

$$\langle O \rangle_{p(\phi;\theta)} \approx \langle O \rangle_{q(\phi)}.$$

Learning

What does the approximate equality of two probability distributions imply?

$$p(\phi;\theta) \approx q(\phi),$$

We anticipate that observables calculated under the ensembles will also be approximately equal.

$$\langle O \rangle_{p(\phi;\theta)} \approx \langle O \rangle_{q(\phi)}.$$

The numerical estimator/expectation value of an arbitrary observable O during a Monte Carlo simulation:

$$\langle O \rangle_{q(\phi)} = \frac{\sum_{l=1}^{N} \tilde{p}_l^{-1} O_l \exp[-\sum_{k=1}^{4} g_k \mathcal{A}_l^{(k)}]}{\sum_{l=1}^{N} \tilde{p}_l^{-1} \exp[-\sum_{k=1}^{4} g_k \mathcal{A}_l^{(k)}]},$$

Learning

Three reweighting (simultaneous) steps: Make the (already trained) inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

Equal to the target action A (acts as a correction step):

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij \rangle} \phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} \phi_i\phi_j$$

Learning

Three reweighting (simultaneous) steps: Make the (already trained) inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij\rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

Equal to the target action A (acts as a correction step):

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij\rangle}\phi_i\phi_j + 1.52425\sum_i\phi_i^2 + 0.175\sum_i\phi_i^4 - \sum_{\langle ij\rangle_{nnn}}\phi_i\phi_j$$

Extrapolate in the parameter space along the trajectory of a coupling constant g' of A

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij\rangle}\phi_i\phi_j + 1.52425\sum_i\phi_i^2 + 0.175\sum_i\phi_i^4 - \sum_{\langle ij\rangle_{nnn}} g'\,\phi_i\phi_j$$

Learning

Three reweighting (simultaneous) steps: Make the (already trained) inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

Equal to the target action A (acts as a correction step):

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij \rangle} \phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} \phi_i\phi_j$$

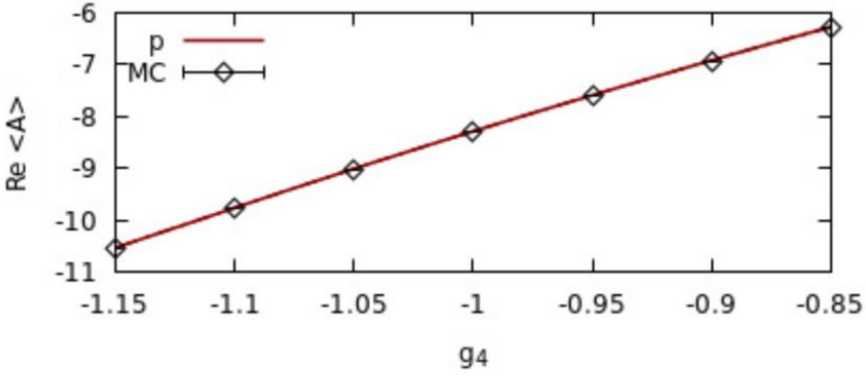
Extrapolate in the parameter space along the trajectory of a coupling constant g' of A

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij \rangle} \phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} g'\phi_i\phi_j$$

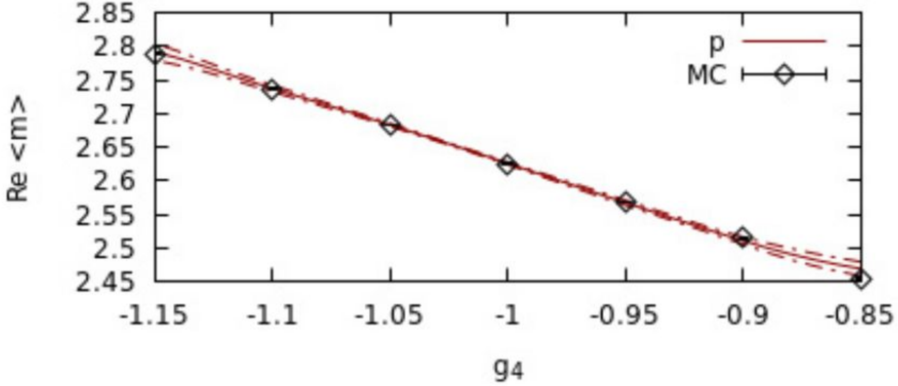
Extrapolate to an imaginary term

$$\mathcal{A}_{\{5\}}(\phi) = -\sum_{\langle ij \rangle} \phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} g'\phi_i\phi_j + i0.15\sum_i \phi_i^2$$

Learning



p
MC
Re <A>
g4
-6
-7
-8
-9
-10
-11
-1.15
-1.1
-1.05
-1
-0.95
-0.9
-0.85



p
MC
Re <m>
g4
2.85
2.8
2.75
2.7
2.65
2.6
2.55
2.5
2.45
-1.15
-1.1
-1.05
-1
-0.95
-0.9
-0.85

Conclusion:
Inhomogeneous actions give rise to probability distributions that have increased representational capacity compared to those of homogeneous actions.

Simply put, an inhomogeneous system can approximate target distributions better than what the same homogeneous system can.

Learning

Conclusion:

Inhomogeneous actions give rise to probability distributions that have increased representational capacity compared to those of homogeneous actions.

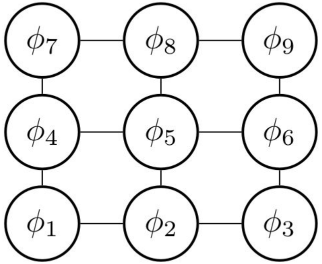
Simply put, an inhomogeneous system can approximate target distributions better than what the same homogeneous system can.

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

$$S(\phi;\theta) = -w\sum_{\langle ij \rangle} \phi_i\phi_j + a\sum_i \phi_i^2 + b\sum_i \phi_i^4$$

Neural Networks

φ^4 Markov random field

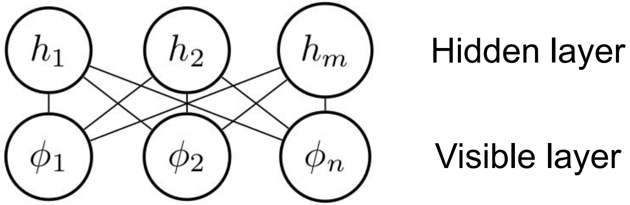


$$S(\phi;\theta) = -\sum_{\langle ij\rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

$$\theta = \{w_{ij}, a_i, b_i\}$$

$$p(\phi;\theta) = \frac{\exp\left[-S(\phi;\theta)\right]}{\int_{\boldsymbol{\phi}} \exp[-S(\boldsymbol{\phi},\theta)]d\boldsymbol{\phi}}.$$

φ^4 neural network



$$S(\phi,h;\theta) = -\sum_{i,j} w_{ij}\phi_i h_j + \sum_i r_i\phi_i + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4 + \sum_j s_j h_j + \sum_j m_j h_j^2 + \sum_j n_j h_j^4,$$

$$\theta = \{w_{ij}, r_i, a_i, b_i, s_j, m_j, n_j\}$$

$$p(\phi,h;\theta) = \frac{\exp[-S(\phi,h;\theta)]}{\int_{\boldsymbol{\phi},\boldsymbol{h}} \exp[-S(\boldsymbol{\phi},\boldsymbol{h};\theta)]d\boldsymbol{\phi}d\boldsymbol{h}}.$$

Neural Networks

The φ^4 neural network:

$$S(\phi, h; \theta) = -\sum_{i,j} w_{ij}\phi_i h_j + \sum_i r_i\phi_i + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4 + \sum_j s_j h_j + \sum_j m_j h_j^2 + \sum_j n_j h_j^4,$$

is a generalization of other neural network architectures:

Gaussian-Gaussian restricted Boltzmann machine:

$b_i = n_j = 0$

Gaussian-Bernoulli restricted Boltzmann machine:

$b_i = n_j = m_j = 0$
h_j binary

Bernoulli-Bernoulli restricted Boltzmann machine:

$b_i = n_j = m_j = a_i = 0$
$\varphi_{i,} h_j$ binary

φ^4-Bernoulli restricted Boltzmann machine:

$m_j = n_j = 0$
h_j binary

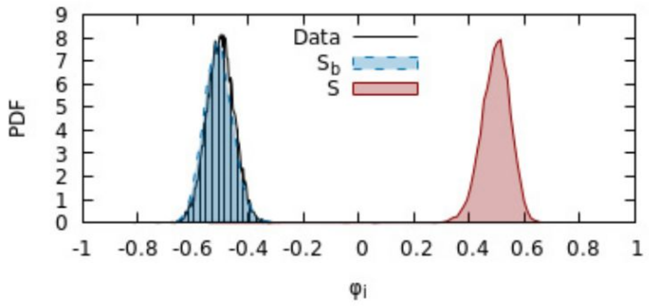
Has not been studied yet

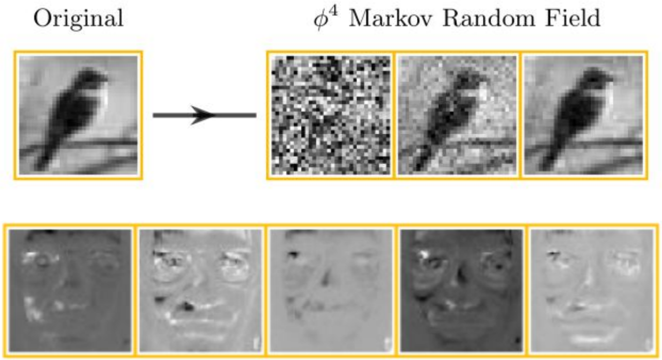
φ^4 equivalence with the Ising model (under an appropriate limit)

Learning

The same approach can be used even when the probability distribution is not known, but we have the probability distribution encoded in some available data. The data can be anything: images or experimental data or a set of Monte Carlo configurations.

$$KL(q||p) = \int_{-\infty}^{\infty} q(\phi) \ln \frac{q(\phi)}{p(\phi;\theta)} d\phi.$$



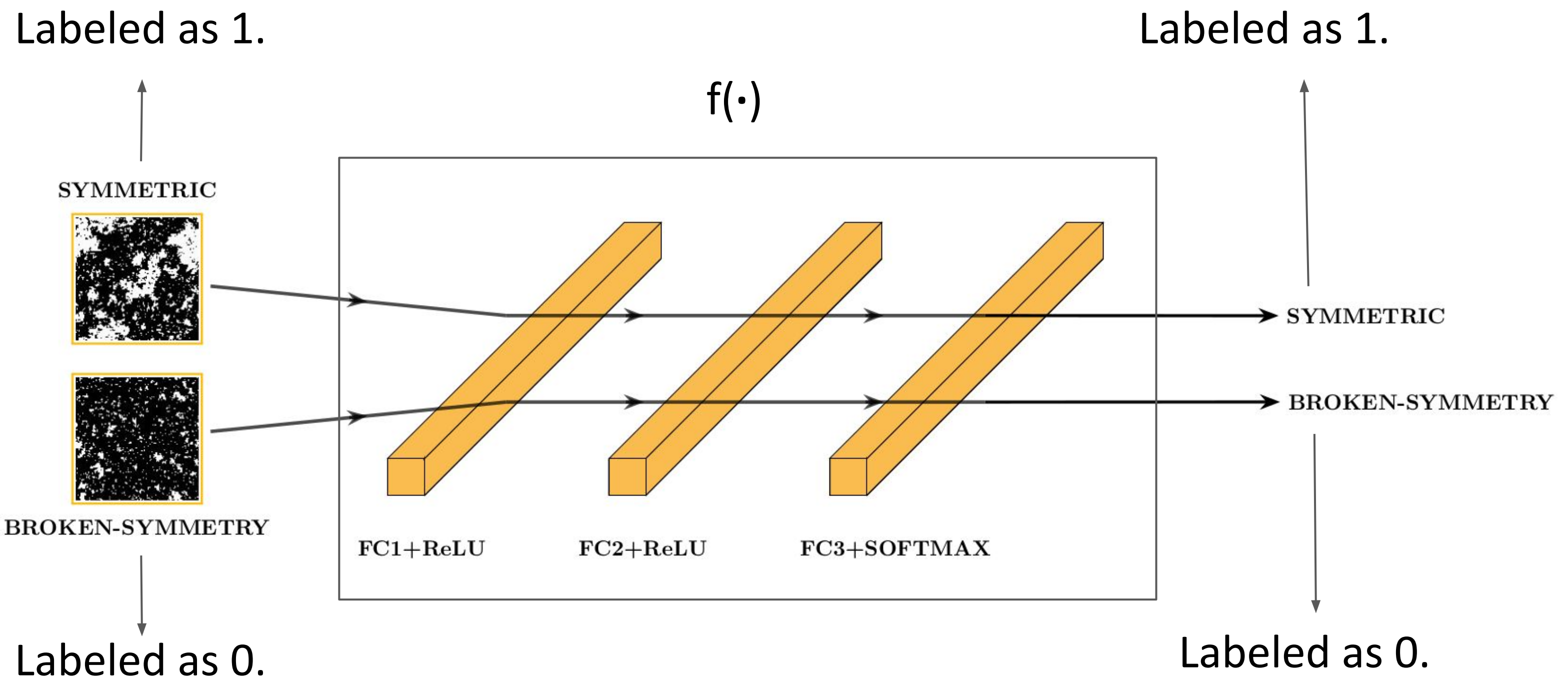


Quantum field-theoretic machine learning, D. Bachtis, G. Aarts and B. Lucini, (arXiv:2102.09449), Phys. Rev. D 103, 074510.

Can we view machine learning functions as statistical-mechanical observables (in a somewhat formal manner)?

Neural Networks as Physical Observables

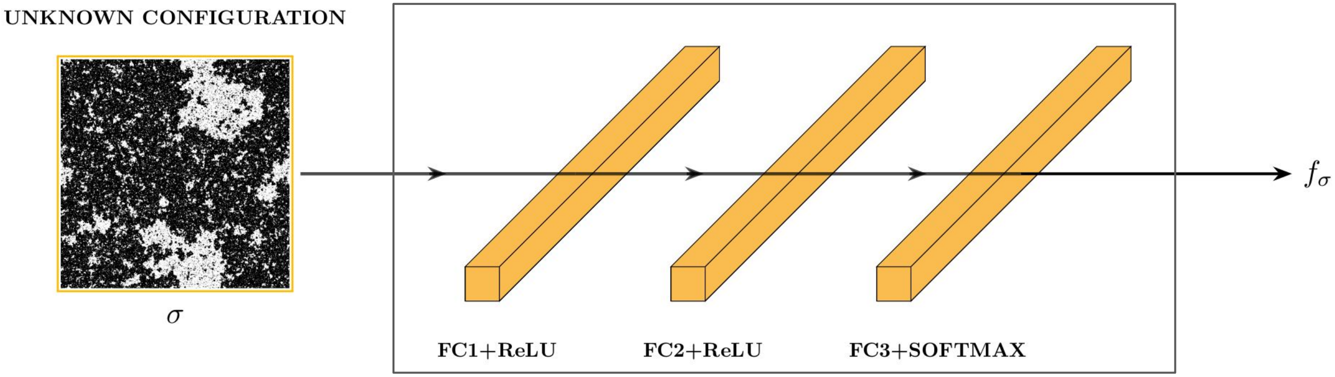
Training of a neural network on the Ising model:



Extending machine learning classification capabilities with histogram reweighting, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E **102** (2020).

Neural Networks as Physical Observables

f(·)

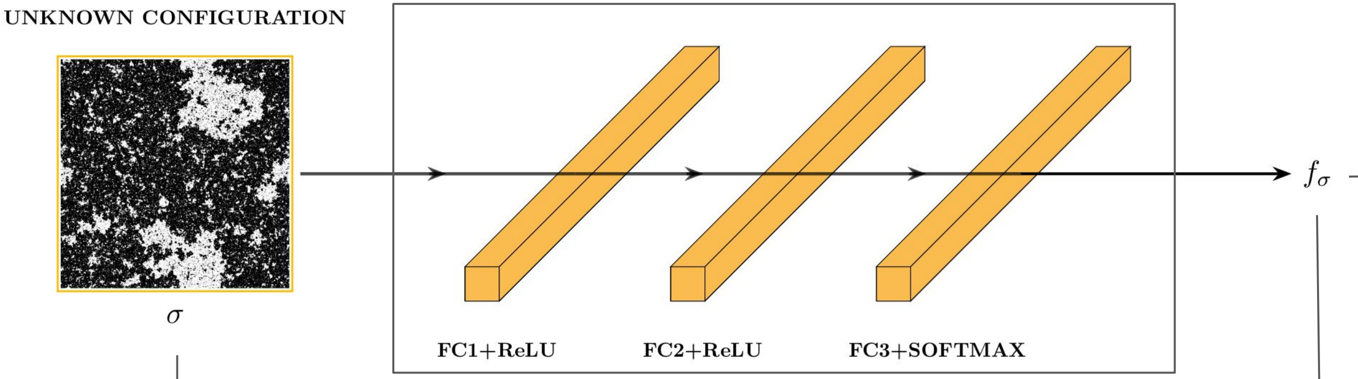


Probability that a configuration is in the broken-symmetry phase

Extending machine learning classification capabilities with histogram reweighting, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E **102** (2020).

Neural Networks as Physical Observables

f(·)



Probability that a configuration is in the broken-symmetry phase

The **configuration** is drawn from an equilibrium distribution and therefore has an **associated Boltzmann weight**.

The **output** is calculated on the configuration so it must have the **same Boltzmann weight**.

Extending machine learning classification capabilities with histogram reweighting, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E **102** (2020).

Neural Networks as Physical Observables

The neural network function is an observable in the system:

$$\langle f \rangle = \sum_{\sigma} f_\sigma p_\sigma = \frac{\sum_\sigma f_\sigma \exp[-\beta E_\sigma]}{\sum_\sigma \exp[-\beta E_\sigma]}$$

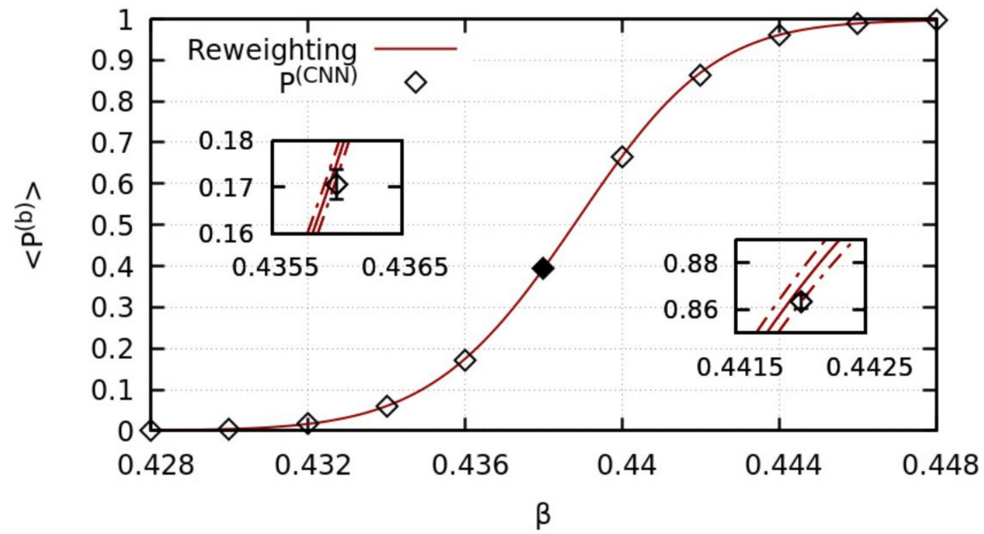
σ : configuration of the system

p_σ : Boltzmann probability distribution

β : inverse temperature

Extending machine learning classification capabilities with histogram reweighting, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E **102** (2020).

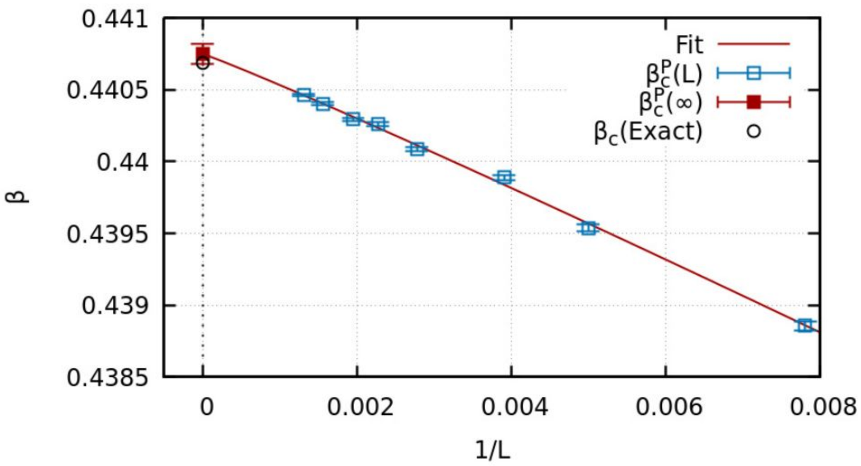
Neural Networks as Physical Observables

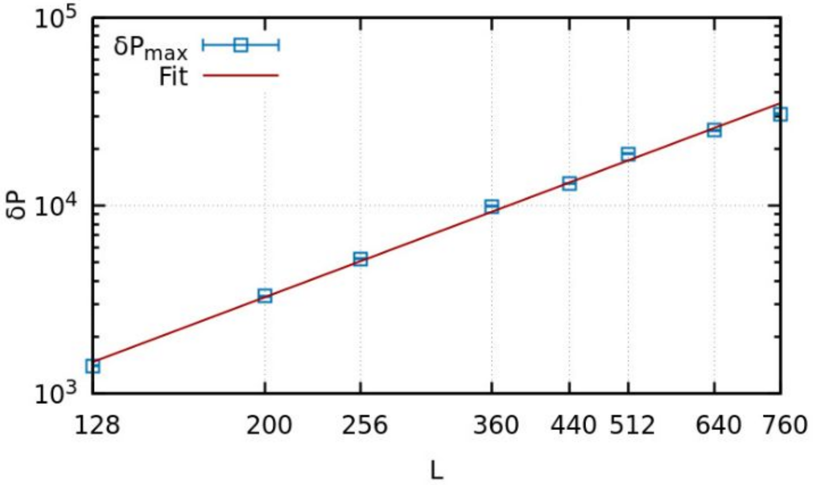


Extending machine learning classification capabilities with histogram reweighting, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E **102** (2020).

Neural Networks as Physical Observables

Results obtained by quantities derived entirely from the neural network





$$|t| = \left|\frac{\beta_c - \beta_c(L)}{\beta_c}\right| \sim \xi^{-\frac{1}{\nu}} \sim L^{-\frac{1}{\nu}}$$

$$\delta P \sim L^{\frac{\gamma}{\nu}}$$

	β_c	ν	γ/ν
CNN+Reweighting	0.440749(68)	0.95(9)	1.78(4)
Exact	$\ln(1+\sqrt{2})/2$	1	7/4
	≈ 0.440687		=1.75

Neural Networks as Physical Observables

The function f(·) was learned on configurations of the Ising model and f(x) can successfully predict the phase of Ising configurations x.

But what happens if we give configurations x’ of a different system as input to the Ising-learned function f(·)?

Can we accurately separate phases in different systems?

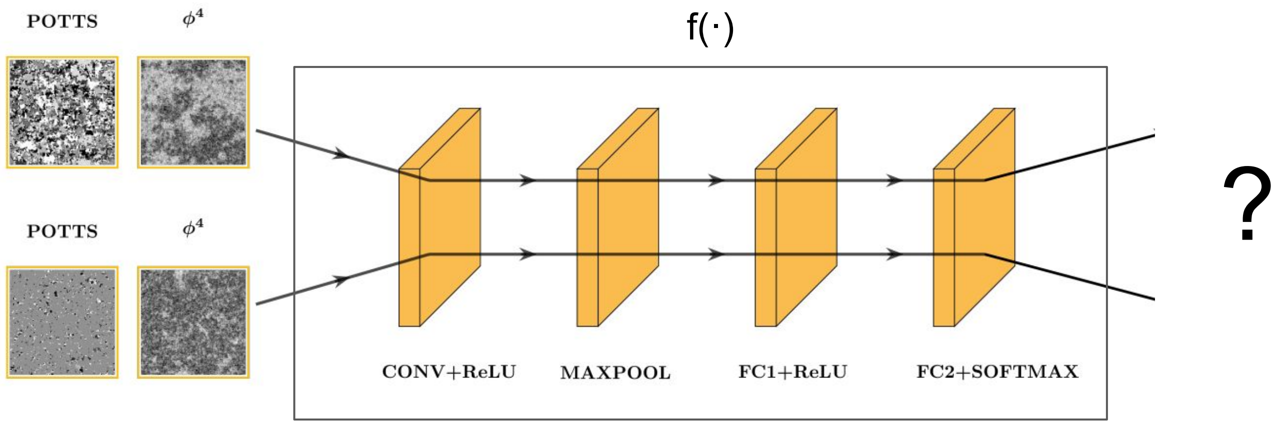
Can we discover a phase transition through f(x’)?

Mapping distinct phase transitions to a neural network, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E 102, 053306 (2020).

Neural Networks as Physical Observables

Equivalently:

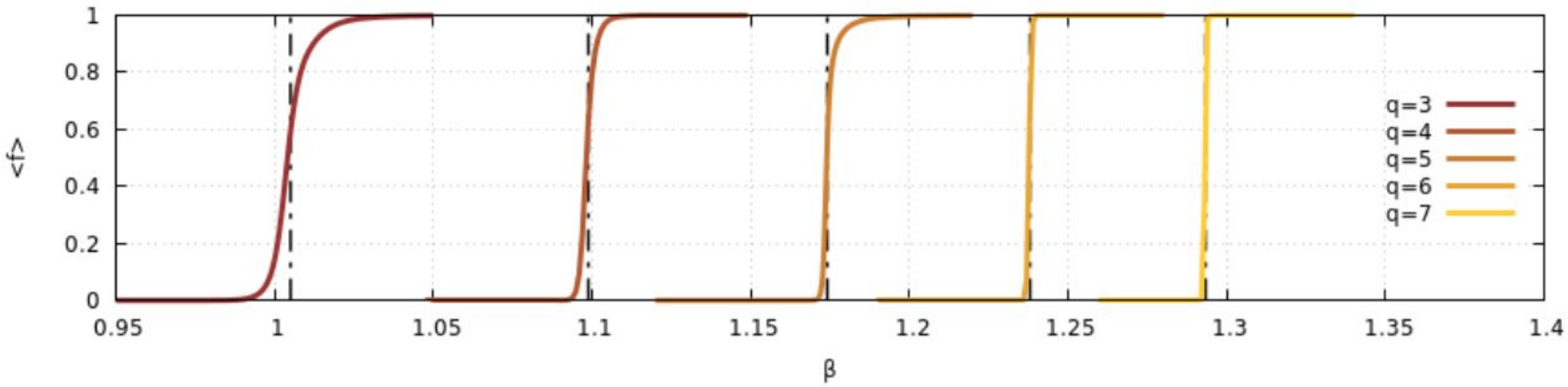
Learned on Ising configurations



Mapping distinct phase transitions to a neural network, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E 102, 053306 (2020).

Neural Networks as Physical Observables

Potts models:

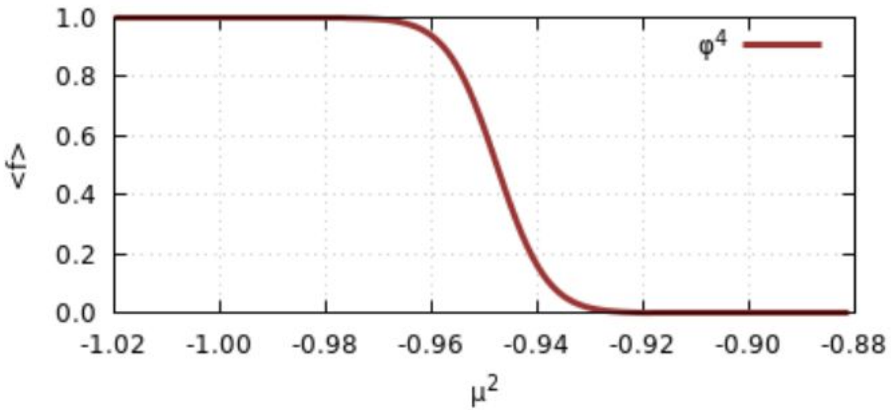


Results obtained through a function f learned exclusively on the Ising model.

Mapping distinct phase transitions to a neural network, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E 102, 053306 (2020).

Neural Networks as Physical Observables

φ^4 scalar field theory:

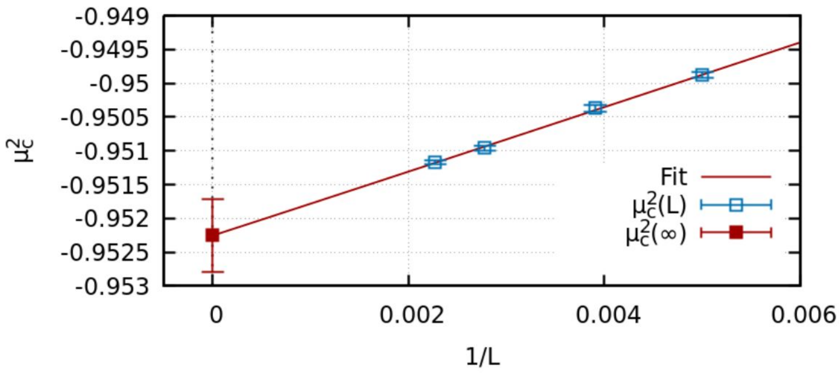


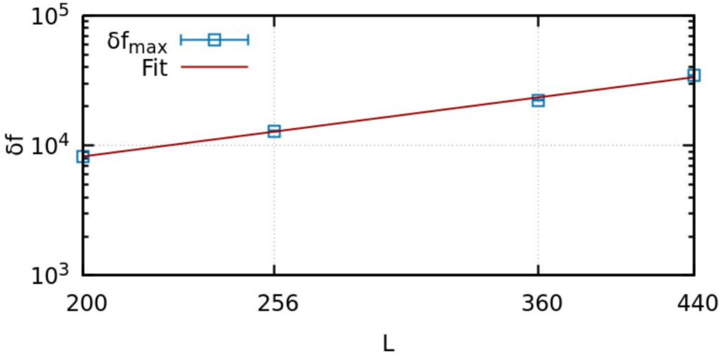
Fixed dimensionless λ=0.7 and varied the dimensionless mass μ^2

Results obtained through a function f learned exclusively on the Ising model.

Mapping distinct phase transitions to a neural network, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E 102, 053306 (2020).

Neural Networks as Physical Observables





	μ_c^2	ν	γ/ν
CNN+Reweighting	-0.95225(54)	0.99(34)	1.78(7)

TABLE II. Critical μ_c^2 for fixed $\lambda_L = 0.7$ and critical exponents of the ϕ^4 scalar field theory.

Adding Machine Learning within Hamiltonians

The neural network function f is an observable in the system.

But what does this mean?

Adding Machine Learning within Hamiltonians

Parameters, constraints or fields that interact with a system have conjugate variables which represent the response of the system to the perturbation of the corresponding parameter.

Can we make the same statement about the neural network function f?

Adding machine learning within Hamiltonians: Renormalization group transformations, symmetry breaking and restoration, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. Research 3, 013134 (arXiv:2010.00054).

Adding Machine Learning within Hamiltonians

We considered that Vf, where V is the volume of the system, couples to an arbitrary external field Y and defined a modified Hamiltonian for the Ising model:

$$E_Y = E - VfY.$$

Adding machine learning within Hamiltonians: Renormalization group transformations, symmetry breaking and restoration, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. Research 3, 013134 (arXiv:2010.00054).

Adding Machine Learning within Hamiltonians

We considered that Vf, where V is the volume of the system, couples to an arbitrary external field Y and defined a modified Hamiltonian for the Ising model:

$$E_Y = E - V f Y.$$

The question:

What happens if we allow a statistical system to interact with a neural network that has been trained to accurately separate its phases? The external field Y denotes the strength of the interaction.

Adding machine learning within Hamiltonians: Renormalization group transformations, symmetry breaking and restoration, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. Research 3, 013134 (arXiv:2010.00054).

Adding Machine Learning within Hamiltonians

Expectation value of an arbitrary observable <O> during a Monte Carlo simulation in the modified system:

$$\langle O \rangle = \frac{\sum_{i=1}^{N} O_{\sigma_i} \tilde{p}_{\sigma_i}^{-1} \exp[-\beta E_{\sigma_i} + \beta V f_{\sigma_i} Y]}{\sum_{i=1}^{N} \tilde{p}_{\sigma_i}^{-1} \exp[-\beta E_{\sigma_i} + \beta V f_{\sigma_i} Y]}$$

By choosing $\tilde{p}_{\sigma_i}$ equal to the probabilities of the original system:

$$\langle O \rangle = \frac{\sum_{i=1}^{N} O_{\sigma_i} \exp[\beta V f_{\sigma_i} Y]}{\sum_{i=1}^{N} \exp[\beta V f_{\sigma_i} Y]}$$

This form of reweighting is **Hamiltonian-agnostic**.

Adding machine learning within Hamiltonians: Renormalization group transformations, symmetry breaking and restoration, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. Research 3, 013134 (arXiv:2010.00054).

Adding Machine Learning within Hamiltonians

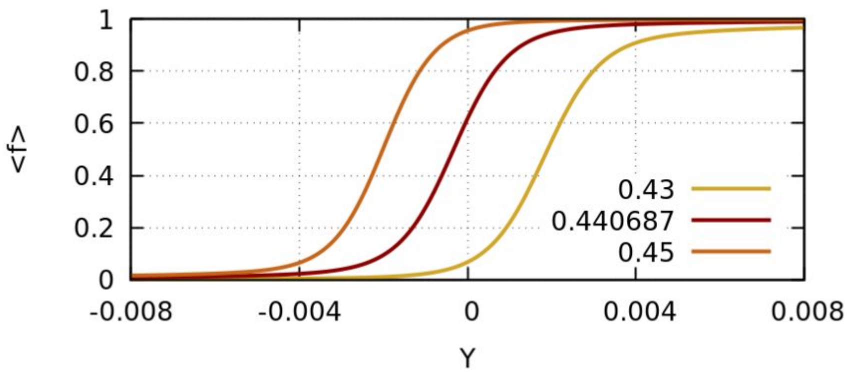


FIG. 2. Mean neural network function $\langle f \rangle$ versus external field Y at inverse temperature $\beta = 0.43, 0, 440687, 0.45$ (right to left). The statistical uncertainty is comparable with the width of the lines.

Recall that:

β=0.43-> symmetric phase

β_c≅0.440687 -> inverse critical temperature

β=0.45-> broken-symmetry phase

Adding machine learning within Hamiltonians: Renormalization group transformations, symmetry breaking and restoration, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. Research 3, 013134 (arXiv:2010.00054).

Adding Machine Learning within Hamiltonians

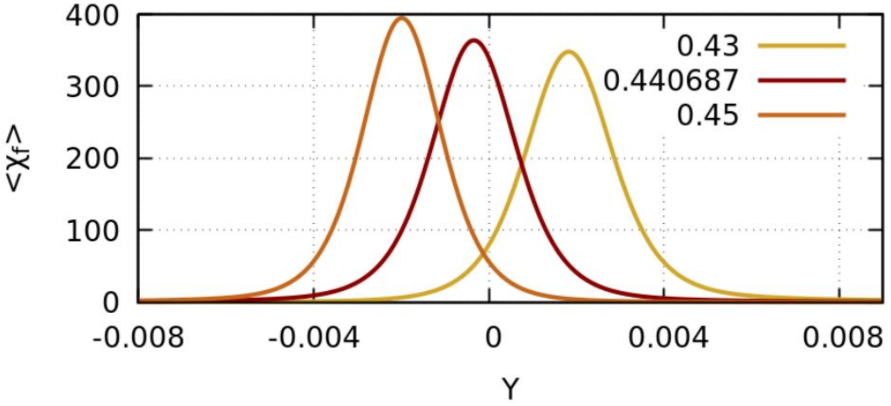
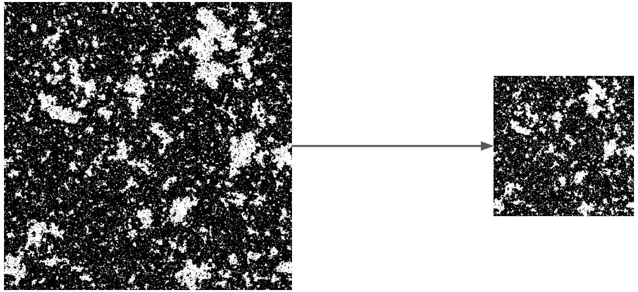


FIG. 3. Mean susceptibility of the neural network function $\langle \chi_f \rangle$ versus external field Y at inverse temperature $\beta = 0.43, 0, 440687, 0.45$ (right to left). The statistical uncertainty is comparable with the width of the lines.

Adding machine learning within Hamiltonians: Renormalization group transformations, symmetry breaking and restoration, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. Research 3, 013134 (arXiv:2010.00054).

Renormalization Group in Real Space

We then studied the neural network function f within the framework of the renormalization group and we were able to obtain the two relevant operators ν and θ that govern the divergence of the correlation length in the Ising model as well as its critical point β_c.



$$\beta' = f^{-1}(f'(\beta)) \qquad Y' = f^{-1}(f'(Y))$$

$$\beta_c = 0.44063(21) \quad \nu = 1.01(2) \quad \theta_Y = 0.534(3)$$

Adding machine learning within Hamiltonians: Renormalization group transformations, symmetry breaking and restoration, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. Research 3, 013134 (arXiv:2010.00054).

Conclusions

For machine learning in phase transitions:

1. No knowledge about the order parameter is required. Effective order parameters can be constructed for systems that undergo distinct phase transitions (universality class, order, etc.).
2. Machine learning observables can be extrapolated or interpolated in parameter space with reweighting. This can additionally be achieved with Hamiltonian-agnostic reweighting.
3. Neural networks can be included in Hamiltonians/lattice actions to break or restore symmetries.
4. We can obtain multiple critical exponents with machine learning functions.

Conclusions

For quantum field-theoretic machine learning:

All one needs to conduct machine learning is any probability distribution p(φ; θ):

$$KL(q||p) = \int_{-\infty}^{\infty} q(\phi) \ln \frac{q(\phi)}{p(\phi;\theta)} d\phi.$$

Conclusions

For quantum field-theoretic machine learning:

All one needs to conduct machine learning is any probability distribution p(φ; θ):

$$KL(q||p) = \int_{-\infty}^{\infty} q(\phi) \ln \frac{q(\phi)}{p(\phi;\theta)} d\phi.$$

Why do we ensure that the Markov property is satisfied?

Conclusions

If we view machine learning as a physical concept...

...is lattice field theory the appropriate tool to describe it?

Conclusions

The Lattice is…

1. Mathematical.

A (mathematically rigorous) bridge between Euclidean and Minkowski QFT:

Construction of quantum fields from Markoff fields, E. Nelson, J. Funct. Anal. 12, 97 (1973)

Conclusions

The Lattice is…

1. Mathematical.
2. Theoretical.

The physics of inhomogeneous quantum field theories:

Renormalization Group in Field Theories with Quantum Quenched Disorder, V. Narovlansky and O. Aharony, Phys. Rev. Lett. 121, 071601 (2018)
Renormalization group flow in field theories with quenched disorder, O. Aharony and V. Narovlansky, Phys. Rev. D 104398, 045012 (2018).
Disorder in large-*N* theories , O. Aharony, Z. Komargodski, & S. Yankielowicz, J. High Energ. Phys. **2016,** 13 (2016).

Conclusions

The Lattice is…

1. Mathematical.
2. Theoretical.
3. Experimental.

An interesting read:

The Hintons in your Neural Network: a Quantum Field Theory View of Deep Learning, Roberto Bondesan, Max Welling, arXiv:2103.04913.

Conclusions

The Lattice is…

1. Mathematical.
2. Theoretical.
3. Experimental.
4. Computational.

Conclusions

The Lattice is…

1. Mathematical.
2. Theoretical.
3. Experimental.
4. Computational.

Thank you for your attention!

Quantum field-theoretic machine learning, D. Bachtis, G. Aarts and B. Lucini, (arXiv:2102.09449), Phys. Rev. D 103, 074510.
Adding machine learning within Hamiltonians: Renormalization group transformations, symmetry breaking and restoration, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. Research 3, 013134 (2020).
Mapping distinct phase transitions to a neural network, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E 102, 053306 (2020).
Extending machine learning classification capabilities with histogram reweighting, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. E **102** (2020).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 813942

EuroPLEx
European network for Particle physics, Lattice field theory and Extreme computing