Classifying Phase Transitions with Machine Learning

Morten Hjorth-Jensen

May 21, 2025

Outline

Phase Transitions: Definitions

- A *phase transition* is a qualitative change in the state of a system when a control parameter (e.g. temperature) passes a critical point.
- **Order parameter**: quantity that distinguishes phases (e.g. magnetization M for magnetic systems).
- **Order vs disorder**: e.g. below T_c a ferromagnet has $|M| > 0$ (ordered), above T_c $M = 0$ (disordered).
- Phases can break symmetries; transitions can be *continuous* (second-order) or *first-order*.

Order Parameter & Symmetry Breaking

- Phase transitions often involve spontaneous symmetry breaking (e.g. Ising model Z_2 symmetry).
- The order parameter (e.g. magnetization $M = \frac{1}{N}\sum_i s_i$) changes behavior at T_c.
- In ferromagnets: $M = 0$ for $T > T_c$ (symmetric paramagnet), $M \neq 0$ for $T < T_c$ (broken symmetry).
- Example: in 2D Ising model, two symmetric ordered states (up/down) below T_c.

Critical Phenomena and Scaling

- Near a continuous transition, observables follow power laws: $M \sim |T - T_c|^{\beta}$, correlation length $\xi \sim |T - T_c|^{-\nu}$, etc.
- **Critical exponents** $(\alpha, \beta, \gamma, \nu, \dots)$ characterize singular behavior.
- Universality: systems with the same symmetry and dimension share exponents.
- E.g. 2D Ising exponents known analytically (Onsager).
- At $T \to T_c$, correlation length $\xi \to \infty$, large-scale fluctuations appear.

2D Ising Model: Definition

- Spins $s_i = \pm 1$ on a 2D square lattice, nearest-neighbor ferromagnetic coupling.
- Hamiltonian:

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

  with $J > 0$ favoring alignment.
- Exhibits a second-order phase transition at critical temperature T_c.
- Order parameter: magnetization $M = \frac{1}{N} \sum_i s_i$.
- Below T_c, $M \neq 0$ (ferromagnetic order); above T_c, $M = 0$ (paramagnet).

2D Ising Model: Critical Temperature

- Exact result (Onsager): critical point T_c satisfies

$$\frac{2J}{k_B T_c} = \ln(1+\sqrt{2}) \quad \Rightarrow \quad T_c \approx \frac{2J}{\ln(1+\sqrt{2})} \approx 2.269J.$$

  (For simplicity set $k_B = 1$) [oai$_c$*itation* : 0*sites.pitt.edu*](*https* : //*sites.pitt.edu*/ *jdnorton*/*teaching*/*philphys*/*Ising*$_s$*im*/*index.html* : : *text* =, *isAt*T¿T$_c$: spins are mostly disordered, no net magnetization.

- At $T < T_c$: long-range order develops (nonzero M).
- Correlation length ξ diverges at T_c [oai$_c$*itation* : 1*sites.pitt.edu*](*https* : //*sites.pitt.edu*/ *jdnorton*/*teaching*/*philphys*/*Ising*$_s$*im*/*index.html* : : *text* = 4, *divergesExample* : *at*T=T$_c$ large clusters of aligned spins appear.

q-State Potts Model: Definition

- Generalization of Ising: each spin $s_i \in \{1, 2, \ldots, q\}$.
- Ferromagnetic Potts Hamiltonian:

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

  where $\delta_{a,b} = 1$ if $a = b$, else 0.
- If $q = 2$, reduces to the Ising model. Higher q allows richer symmetry breaking ($\mathbb{Z}_q$).
- Widely used to study phase transitions with multiple equivalent ordered states.

- In 2D, the ferromagnetic Potts model has a phase transition for all $q \geq 1$ [oai$_c$*itation* : 2*en.wikipedia.org*](*https* : //*en.wikipedia.org*/*wiki*/*Potts*$_m$*odel* : : $text = for Exactcriticalpoint$: $\frac{J}{k_B T_c} = \ln(1 + \sqrt{q})$.

- The nature of the transition depends on q [oai$_c$*itation* : 3*en.wikipedia.org*](*https* : //*en.wikipedia.org*/*wiki*/*Potts*$_m$*odel* : : $text = for$

  $1 \leq q \leq 4$: continuous (second-order) transition.

  $q > 4$: discontinuous (first-order) transition (latent heat appears).

  Example: $q = 3, 4$ have continuous transitions; $q = 5$ and higher show first-order behavior.

Monte Carlo Sampling of Spin Models

- Use Monte Carlo (MC) to generate spin configurations at given T: sample from Boltzmann distribution $P \propto e^{-H/T}$.
- Metropolis algorithm: attempt random single-spin flips to equilibrate the system.
- Provides training data: spin configurations $\{s_i\}$ labeled by temperature or phase.
- Ensures statistical equilibrium and detailed balance [oai$_c$$itation : 4rajeshrinet.github.io$]($https : //rajeshrinet.github.io/blog/2014/ising - model/ : \quad : text = 1, 4).Efficientsampling(especiallynear$T$_c$ cluster algorithms help, e.g. Wolff or Swendsen-Wang).

Metropolis Algorithm (Concept)

- Initialize spins randomly or in a fixed state.
- Repeat for many steps:
  1. Pick a random lattice site i.
  2. Propose flipping $s_i \to -s_i$ (Ising) or change state (Potts).
  3. Compute energy change ΔE.
  4. If $\Delta E \leq 0$, accept the flip (lower energy).

5. Else accept with probability $\exp(-\Delta E/T)$ (Boltzmann factor) [oai$_c$$itation : 5rajeshrinet.github.io](https : //rajeshrinet.github.io/blog/2014/ising - model/ : \;\; : text = 1, 4).Otherwise, rejectandkeeptheoldstate$.

6. After equilibration, record configurations as samples.

Metropolis Algorithm (Pseudo-code)

```
for T in temperature_list:
    # Initialize lattice (e.g., random spins)
    config = random_configuration(Lx, Ly)
    for step in range(num_steps):
        i,j = random_site()
        dE = compute_deltaE(config, i, j)  # energy change if
             spin flipped
        if dE <= 0 or rand() < exp(-dE/T):
            flip_spin(config, i, j)
    record_configuration(config, T)
```

Monte Carlo Data for ML

- Generate many spin configurations across a range of temperatures T.
- Label each configuration by its temperature or by phase (ordered/disordered).
- This labeled dataset is used for *supervised* methods (e.g. CNN).
- For *unsupervised* methods (PCA, VAE), labels are not used in training.
- Data augmentation: one can use symmetries (e.g. spin flip) to enlarge dataset.

Principal Component Analysis (PCA) Basics

- PCA is an unsupervised method for dimensionality reduction.
- Finds orthogonal directions (principal components) of maximum variance in data.
- Project data onto the first few PCs to visualize structure.
- Advantages: linear, fast, and interpretable (PCs are linear combinations of features).
- Disadvantage: only captures linear correlations (may miss complex features).

PCA for Phase Identification

- Apply PCA to the ensemble of spin configurations (flattened to vectors).
- The first principal component (PC1) often correlates with the order parameter (e.g. magnetization).
- Hu et al. (2017) found PCA distinguishes different phases and can locate critical points [oai$_c$itation : 6link.aps.org](https : //link.aps.org/doi/10.1103/PhysRevE.95.062122 : : text = WeByplottingdatainthesubspaceofPCs, oneseesseparationoflow−T(ordere

- No labels needed: phase transitions are revealed by clustering in PC space [oai$_c$itation : 7link.aps.org](https : //link.aps.org/doi/10.1103/PhysRevE.95.062122 : : text = We

PCA Workflow for Spin Data

- Collect data matrix X of shape (num_samples) $\times$ (num_features), e.g. $N \times (L \times L)$.
- Subtract the mean from each column (feature) of X.
- Compute covariance matrix $C = X^T X$ (or use SVD on X directly).
- Obtain eigenvalues/vectors of C: $C = U \Lambda U^T$. Columns of U are principal directions.
- Sort by eigenvalues (variance). Project X onto top k PCs: $X_{\mathrm{red}} = X\, U[:, 1:k]$.
- Analyze X_{red}: e.g. scatter plot PC1 vs PC2.

PCA Example: Ising Model

- In the 2D Ising model, PC1 is essentially proportional to the overall magnetization.
- At $T < T_c$, configurations cluster with large positive or negative PC1 (ordered states).
- At $T > T_c$, configurations cluster near PC1 0 (disordered).
- The variance captured by PC1 drops sharply at T_c, signaling the transition.

- PCA automatically finds these features, without knowing the physics a priori [oai$_c$$itation : 8link.aps.org](https : //link.aps.org/doi/10.1103/PhysRevE.95.062122 : \;\; : text = We$

PCA Limitations

  - PCA is linear: complex nonlinear features (e.g. vortex order) may not be captured.
  - Example: In a frustrated 2D spin model, PCA failed to detect certain correlations (vorticity) [oai$_c itation : 9 link.aps.org](https : //link.aps.org/doi/10.1103/PhysRevE.95.062122 : \;\; : text = well PCA does not directly classify; it provides features for clustering or visualizat$
- Sensitive to scaling: data should be normalized appropriately.
- Still useful as a first-pass: identifies the most significant variations [oai$_c itation : 10 link.aps.org](https : //link.aps.org/doi/10.1103/PhysRevE.95.062122 : \;\; : text = We$

PCA with PyTorch (Example Code)

```
import torch

# X: tensor of shape (N, L*L) containing spin configurations
     as floats (e.g. +1/-1)
# Center the data
X = X - X.mean(dim=0, keepdim=True)

# Compute covariance (or use torch.pca_lowrank)
cov = torch.mm(X.t(), X) / (X.size(0)-1)

# Eigen-decomposition (SVD) of covariance
U, S, V = torch.svd(cov)

# Select first k principal components
k = 2
PCs = U[:, :k]  # shape (L*L, k)

# Project data onto principal components
X_reduced = torch.mm(X, PCs)  # shape (N, k)
```

Convolutional Neural Networks (CNNs)

- CNNs are deep neural networks designed for spatial data (e.g. images).
- Architecture: convolutional layers (feature detectors) + pooling, followed by fully connected layers.
- In physics: treat spin lattice as an image with multiple channels (e.g. one channel of spins).
- CNNs can learn complex nonlinear features automatically from data.
- They require labeled examples for training (supervised learning).

CNN for Phase Classification

- Prepare training data: spin configurations labeled by phase or temperature.
- CNN learns to map configuration $\rightarrow$ phase label (ordered/disordered) or predict T.
- As shown by Carrasquilla and Melko (2017), CNNs can identify phases from raw states [oai$_c$$itation : 11nature.com](https : //www.nature.com/articles/nphys4035 : \ : text =$ $abilitiesAchieveshighaccuracyonIsingandothermodelswhentraininglabelsar$
- CNNs exploit locality: can detect clusters or domains of aligned spins via convolution filters.

Example CNN Architecture

- **Input**: single-channel $L \times L$ lattice (values -1 or $+1$).
- **Conv layer 1**: e.g. 8 filters of size 3×3, ReLU activation, stride=1, padding=1.
- **Conv layer 2**: 16 filters of size 3×3, ReLU, followed by a 2×2 max-pooling.
- **Fully Connected**: flatten feature maps to vector; FC layer to 64 units (ReLU); final FC to 2 outputs (softmax for binary phase).
- **Training**: minimize cross-entropy loss between predicted and true labels.
- **Note**: architecture and hyperparameters can be tuned for best performance.

CNN: Training and Results

- Train on many labeled samples (e.g. temperatures T and whether $T < T_c$ or $T > T_c$).
- The network learns features such as magnetization domains, energy patterns, etc.
- CNN accuracy can be very high (often $\sim$100% on clean data) for distinguishing phases.

- Fukushima & Sakai (2021): a CNN trained on 2D Ising can detect transition in q-state Potts [oai$_c$$itation : 12arxiv.org](https : //arxiv.org/abs/2104.03632 : \; : text = generated, However\,CNNbehavior : athigh\mathrm{T}iteffectivelyusesaverageenergy; atlow\mathrm{T}itcorrelateswithmagnetization[o$ $13arxiv.org](https : //arxiv.org/abs/2104.03632 : \; : text = accuracy$

CNN Interpretability

- CNNs are often seen as "black boxes", but their learned filters can sometimes be interpreted.
- Outputs correlate with known physics:
    - At low T: classification heavily influenced by magnetization (order).
- At high T: classification influenced by internal energy (disorder) $[\mathrm{oai}_c itation : 14arxiv.org](https : //arxiv.org/abs/2104.03632 : \; : text = accuracy$

- CNNs can generalize: e.g. Ising-trained CNN finds Potts T_c $[\mathrm{oai}_c itation : 15arxiv.org](https : //arxiv.org/abs/2104.03632 : \; : text = generated, However Visualizationmethods(e.g. saliencymaps)canhighlightwhat$

CNN (PyTorch) Code Example

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class PhaseCNN(nn.Module):
    def __init__(self, L):
        super(PhaseCNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 8, kernel_size=3, padding
            =1)  # 1 channel -> 8
        self.conv2 = nn.Conv2d(8, 16, kernel_size=3, padding
            =1) # 8 -> 16
        self.pool = nn.MaxPool2d(2)  # downsample by 2
        self.fc1 = nn.Linear(16 * (L//2) * (L//2), 64)
        self.fc2 = nn.Linear(64, 2)  # 2 output classes

    def forward(self, x):
        x = F.relu(self.conv1(x))       # (B,8,L,L)
        x = self.pool(F.relu(self.conv2(x)))  # (B,16,L/2,L
            /2)
        x = x.view(x.size(0), -1)       # flatten
        x = F.relu(self.fc1(x))
```

Variational Autoencoders (VAE) Overview

- A VAE is an *unsupervised* generative model that learns a latent representation of data.
- Components:
  - **Encoder**: maps input X to parameters $(\mu, \log \sigma^2)$ of a latent Gaussian.
  - **Latent** z: sampled via $z = \mu + \sigma\epsilon$ $(\epsilon \sim N(0, I))$.
  - **Decoder**: reconstructs input $\hat{X}$ from z.
- Loss: reconstruction error + KL divergence to enforce latent prior $\mathcal{N}(0, I)$.
- VAEs can both encode data compactly and generate new samples by sampling z.

VAE for Spin Configurations

- Train VAE on spin configurations (no labels).
- Latent space (usually low-dimensional) captures key features (like order parameter).
- Walker et al. (2020): latent variables provide metrics to track order vs disorder in Ising [oai$_c$*itation* : 16*nature*.*com*](*https* : //*www*.*nature*.*com*/*articles*/*s*41598 − 020 − 69848 − 5 : : *text* = *TheTheyfoundthelatentrepresentationcloselycorrespondstophysicalorder*(
17*nature*.*com*](*https* : //*www*.*nature*.*com*/*articles*/*s*41598 − 020 − 69848 − 5 : : *text* = *By*

- After training, one can:
  - Inspect latent space (e.g. scatter plot of (μ_1, μ_2)) to distinguish phases.
  - Sample $z \sim N(0, 1)$ and decode to generate synthetic configurations.

VAE Architecture Details

- Typically use convolutional encoder/decoder for 2D structure.
- Example:
  - Encoder: conv layers downsampling to a flat vector $\rightarrow$ linear layers $\rightarrow (\mu, \log \sigma^2)$ (size of latent space, e.g. 2–10 dims).
  - Decoder: linear layer from z to feature map size, followed by transposed-conv layers to reconstruct $L \times L$ lattice.
- Activation: ReLU (or LeakyReLU); final output often sigmoid to model spin distribution.
- Training with minibatch gradient descent optimizing

$$\mathcal{L} = \mathbb{E}[\|X - \hat{X}\|^2] + \mathrm{KL}(\mathcal{N}(\mu, \sigma) \,\|\, \mathcal{N}(0, 1)).$$

VAE Results on Ising Model

- The first latent dimension (ν_0) learned by the VAE correlated strongly with magnetization [oai$_c$*itation* : 18*nature*.*com*](*https* : //*www*.*nature*.*com*/*articles*/*s*41598 − 020 − 69848 − 5 : : *text* = *ByPlotting*ν_0 vs temperature shows clear change around T_c (order–disorder).

- This means VAE "discovered" the order parameter without supervision.

- The VAE predicted the critical region and crossover consistently with theory [oai$_c$*itation* : 19*nature*.*com*](*https* : //*www*.*nature*.*com*/*articles*/*s*41598 − 020 − 69848 − 5 : : *text* = *TheLatentspaceclustering* : *ordered* − *phasepointsseparatefromdisordered*.

VAE: Generation and Interpretation

- After training, sample random z from Gaussian prior and decode to generate configurations.
- The VAE latent space is continuous: can interpolate between phases.
- The learned representation is smooth and disentangled: one latent coordinate tracks magnetization, others track disorder.
- VAEs can also be used for anomaly detection: points with unusual z indicate atypical states.
- Overall, VAEs provide both a dimensionally-reduced view of phase structure and a generative model.

VAE (PyTorch) Code Example

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class VAE(nn.Module):
    def __init__(self, L, latent_dim=2):
        super(VAE, self).__init__()
        # Encoder: conv -> conv -> flatten -> fc_mu/fc_logvar
        self.encoder = nn.Sequential(
            nn.Conv2d(1, 8, 3, stride=2, padding=1),   # ->
                (8, L/2, L/2)
            nn.ReLU(),
            nn.Conv2d(8, 16, 3, stride=2, padding=1),  # ->
                (16, L/4, L/4)
            nn.ReLU(),
            nn.Flatten()
        )
        self.fc_mu = nn.Linear(16*(L//4)*(L//4), latent_dim)
        self.fc_logvar = nn.Linear(16*(L//4)*(L//4),
            latent_dim)
```

Supervised vs Unsupervised Methods

- **Supervised (CNN)**: Requires labeled data (phase labels or temperatures). Learns a direct mapping $\{\text{config}\} \rightarrow \{\text{phase}\}$.
- **Unsupervised (PCA, VAE)**: Uses only the raw configurations without labels. Learns features or representations of the data.
- PCA reduces dimensionality; requires no training labels $[\mathrm{oai}_c itation : 20 link.aps.org](https : //link.aps.org/doi/10.1103/PhysRevE.95.062122 : \; : text = We VAE learns a latent generative model; also label - free[oai_c itation : 21 nature.com](https : //www.nature.com/articles/s41598 - 020 - 69848 - 5 : \; : text = The$

- CNN typically achieves higher accuracy in classifying known phases, but needs supervised labels.

Method Interpretability and Features

- **PCA**: Principal components often have clear physical meaning (e.g. PC1 magnetization) [oai$_c$$itation : 22link.aps.org](https : //link.aps.org/doi/10.1103/PhysRevE.95.062122 : \ : text = We$**CNN** $: Filtersarelessdirectlyinterpretable; featuresarelearned.However, somecorre$ $23arxiv.org](https : //arxiv.org/abs/2104.03632 : \ : text = accuracy$
- **VAE**: Latent variables can often be interpreted as order/disorder features (e.g. one latent magnetization) [oai$_c$$itation : 24nature.com](https : //www.nature.com/articles/s41598 - 020 - 69848 - 5 : \ : text = By CNNisa'' blackbox'' classifier; PCA/VAEprovideinsightintodatastructure.$
- In terms of visualization: PCA and VAE produce low-dim plots of data (semi-transparent), whereas CNN only outputs a decision boundary.

Performance and Use Cases

- **PCA**: Fast to compute; good for preliminary analysis of large datasets. Best for linearizable transitions.
- **CNN**: High classification accuracy; powerful for large and complex datasets. Can predict critical T or classify multiple phases [oai$_c$*itation* : 25*arxiv*.*org*](*https* : //*arxiv*.*org*/*abs*/2104.03632 : : *text* = *generated*, *However* **VAE** : *Usefulwhennolabelsareavailable*; *providesagenerativemodel*.*Effectiveindete*
26*nature*.*com*](*https* :
//*www*.*nature*.*com*/*articles*/*s*41598 − 020 − 69848 − 5 : : *text* = *The*

- Computational cost: PCA very cheap, CNN and VAE require training time (GPU recommended for large data).
- Choosing a method: depends on data availability and goal (classification vs insight vs generation).

Example Case Studies

- **Ising/CNN**: CNNs trained on Ising achieve $\approx 100\%$ phase classification accuracy.
- **Ising $\rightarrow$ Potts**: Fukushima & Sakai show an Ising-trained CNN still identifies T_c in Potts models $[\mathrm{oai}_c itation : 27 arxiv.org](https : //arxiv.org/abs/2104.03632 : \quad : text = generated, However$ **PCA on Potts** $: PCAcanalsolocate \mathrm{T}_c$ for Potts by looking at variance of PCs.

- **VAE on Ising**: Walker et al. VAE found latent coordinate correlates with magnetization; predicted crossover region $[\mathrm{oai}_c itation : 28 nature.com](https : //www.nature.com/articles/s41598 - 020 - 69848 - 5 : \quad : text = By$ **Limitation example** $: PCAfailedtofindtransitioninafrustratedspinmodelduetonon - linearorder[oai_c itation : 30 link.aps.org](https : //link.aps.org/doi/10.1103/PhysRevE.95.062122 : \quad : text = well$

Summary of Methods

- **PCA**: Unsupervised, linear, interpretable. Good for dimensionality reduction and initial exploration [oai$_c$$itation : 31link.aps.org](https : //link.aps.org/doi/10.1103/PhysRevE.95.062122 : \ : text = We\textbf{CNN} : Supervised, non - linear, highaccuracy.Requireslabels, butlearnscomplexfeatures(worksacros$ $32arxiv.org](https : //arxiv.org/abs/2104.03632 : \ : text = generated, However$
- **VAE**: Unsupervised, generative. Learns latent representation reflecting order/disorder [oai$_c$$itation : 33nature.com](https : //www.nature.com/articles/s41598 - 020 - 69848 - 5 : \ : text = By Eachmethodhastrade - offsinaccuracy, interpretability, anddatarequirements.$
- Combining methods (e.g. using PCA or VAE features as input to another classifier) can also be fruitful.

Conclusions

- Machine learning provides powerful tools for studying phase transitions in statistical models.
- *Unsupervised* methods (PCA, VAE) can discover phase structure without labels [oai$_c itation : 34 link.aps.org](https : //link.aps.org/doi/10.1103/PhysRevE.95.062122 : \; : text = WeSupervised methods(CNNs) achievehighclassificationperformancegiven$ $36 nature.com](https : //www.nature.com/articles/nphys4035 : \; : text = abilities$

- Interpretability: PCA/VAE offer more insight into physics (latent/PC represent order parameters), while CNNs focus on prediction accuracy.
- Choice of method depends on the problem: data availability, need for generative modeling, and interpretability.
- Future directions: deeper architectures (e.g. ResNets), unsupervised generative flows, transfer learning across models, real experimental data.

References

- Carrasquilla, J. & Melko, R. G. (2017). Machine learning phases of matter. *Nature Physics*, 13, 431–434 [oai$_c$*itation* : 37*nature.com*](*https* : //*www.nature.com*/*articles*/*nphys*4035 : : *text* = *abilitiesHu*, *W.et al.*(2017).*Discoveringphases*, *phasetransitionsthroughun* 38*link.aps.org*](*https* : //*link.aps.org*/*doi*/10.1103/*PhysRevE*.95.062122 : : *text* = *We*
- Fukushima, K. & Sakai, K. (2021). Can a CNN trained on Ising detect Potts? *Prog. Theor. Exp. Phys.* 2021, 061A01 [oai$_c$*itation* : 39*arxiv.org*](*https* : //*arxiv.org*/*abs*/2104.03632 : : *text* = *generated*, *HoweverWalker*, *N.et al.*(2020).2*DIsingmodelcrossoverviaVAE.Sci* 40*nature.com*](*https* : //*www.nature.com*/*articles*/*s*41598 − 020 − 69848 − 5 : : *text* = *The*
- .