Learning Phase Transitions

with/without Confusion

Ye-Hua Liu
ETH Zurich

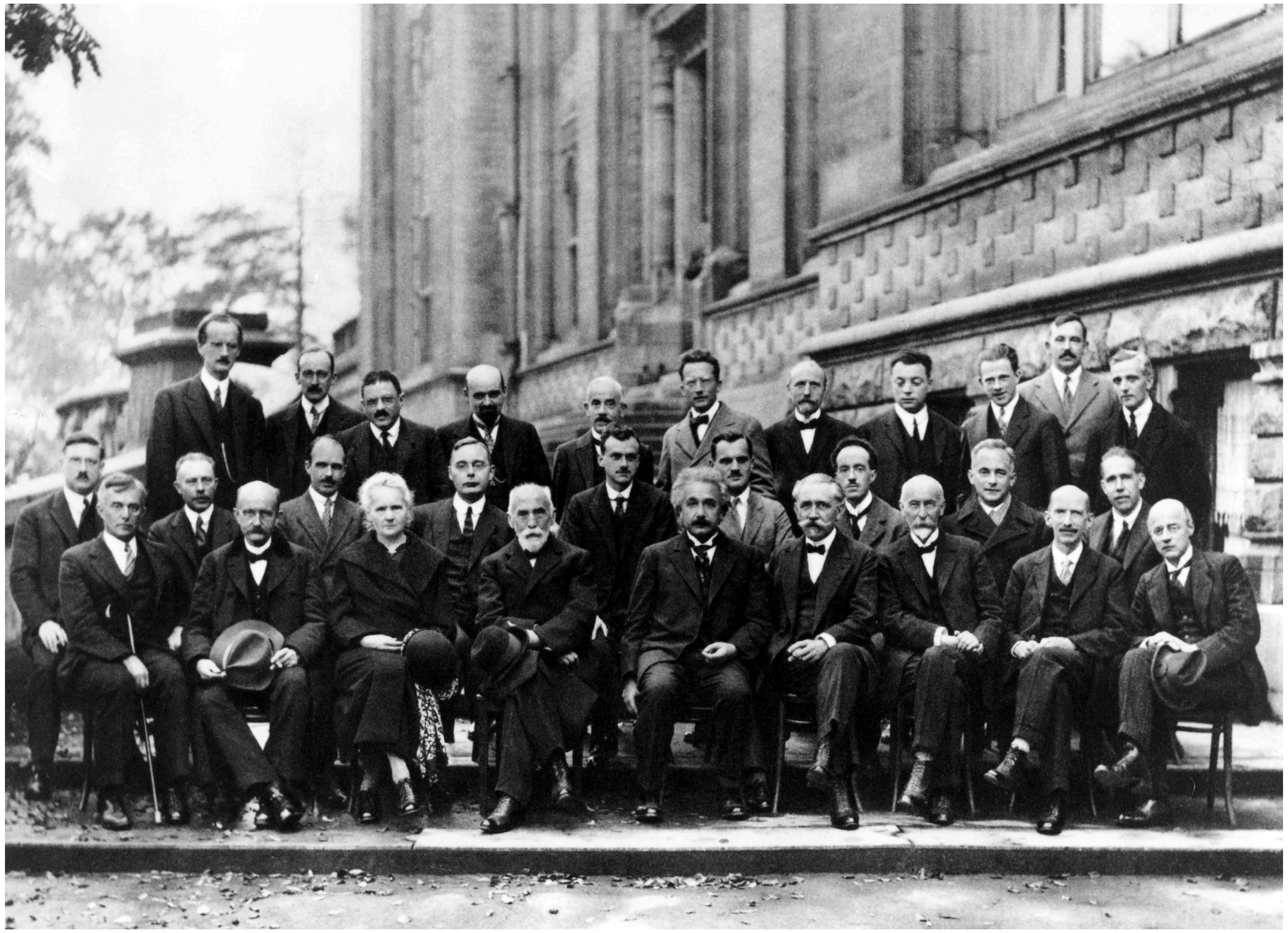
Collaborators:
E. P. L. van Nieuwenburg (Caltech)
S. D. Huber (ETHZ)

July 3rd, 2017
KITS, Beijing

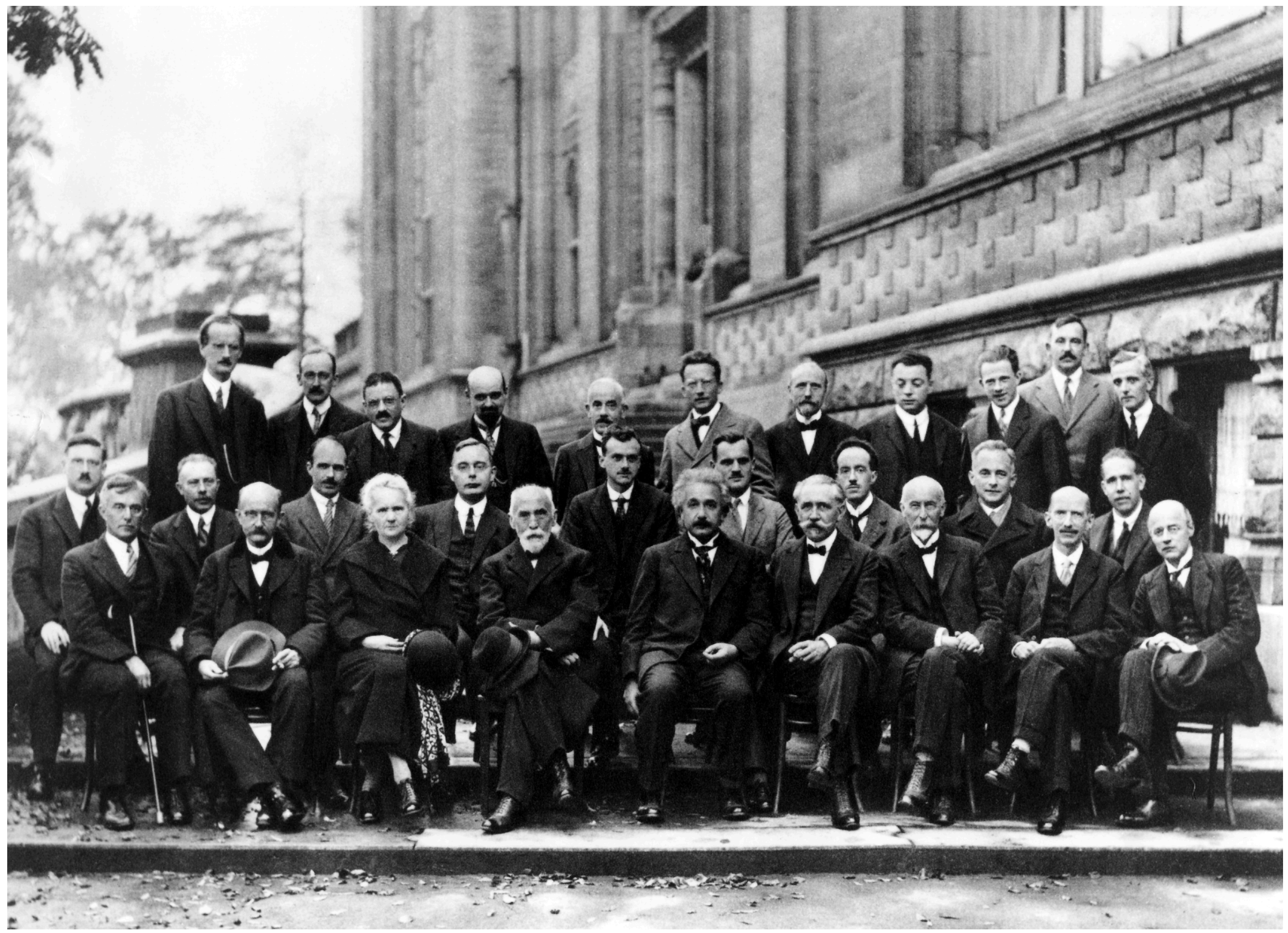
Outline

- Supervised/Unsupervised learning of phase transitions
- A hybrid: the confusion scheme (with)
- The self-learning scheme (without)
- The self-learning snake for 2D parameter spaces (without)

Motivation



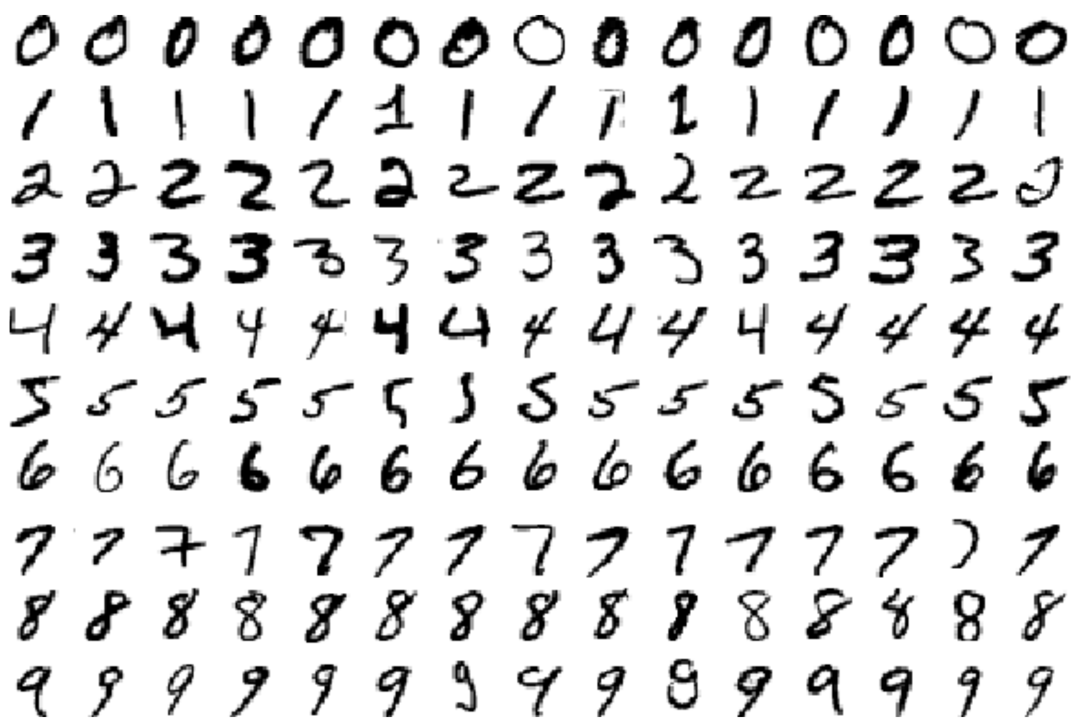
Motivation

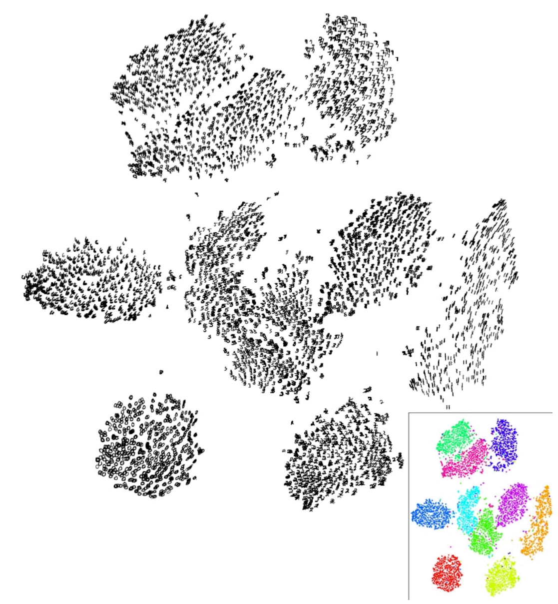


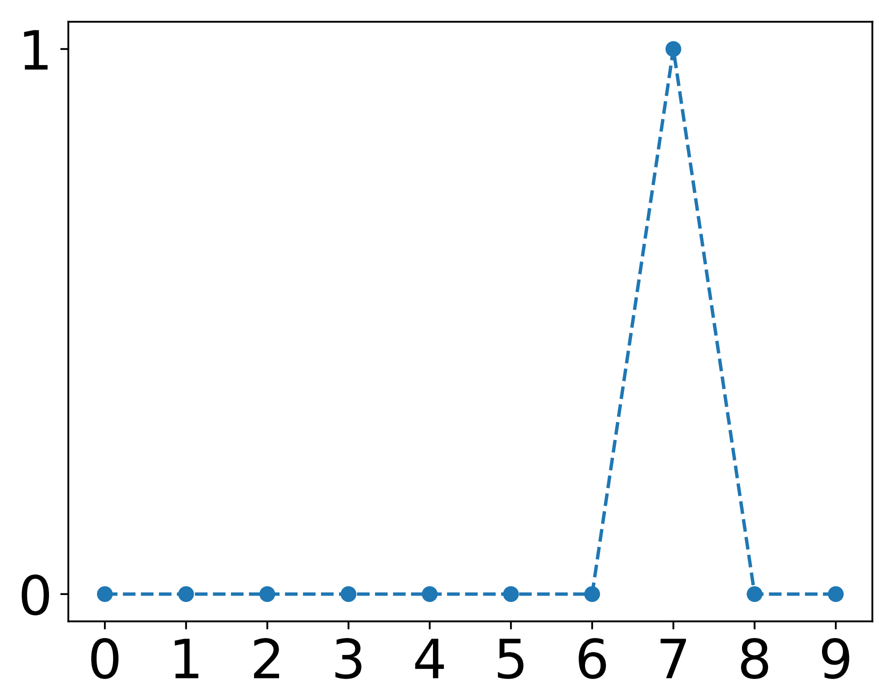
What can I help you with?

Machine Learning Classification

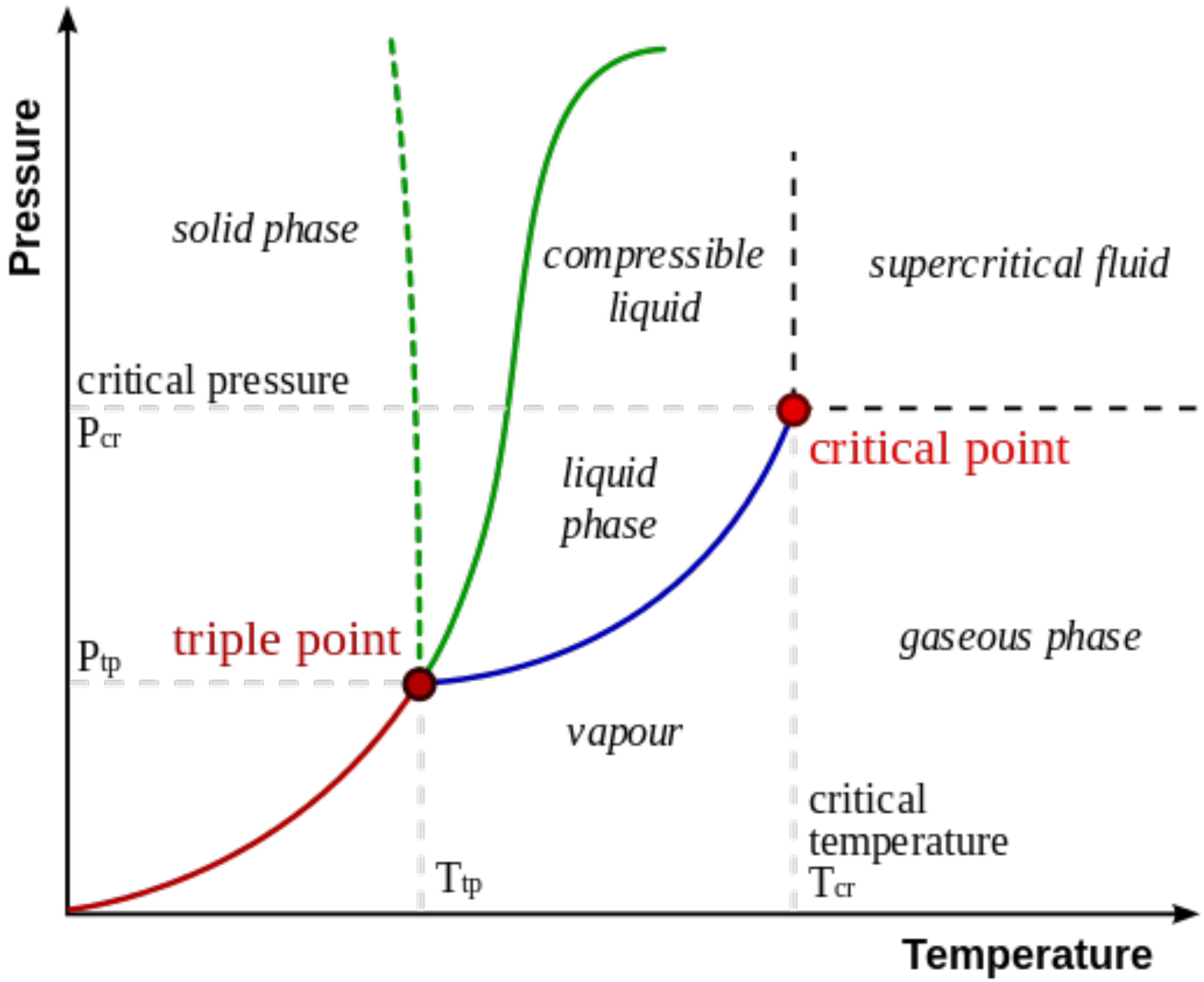
Paradigm: given data, assign a class.



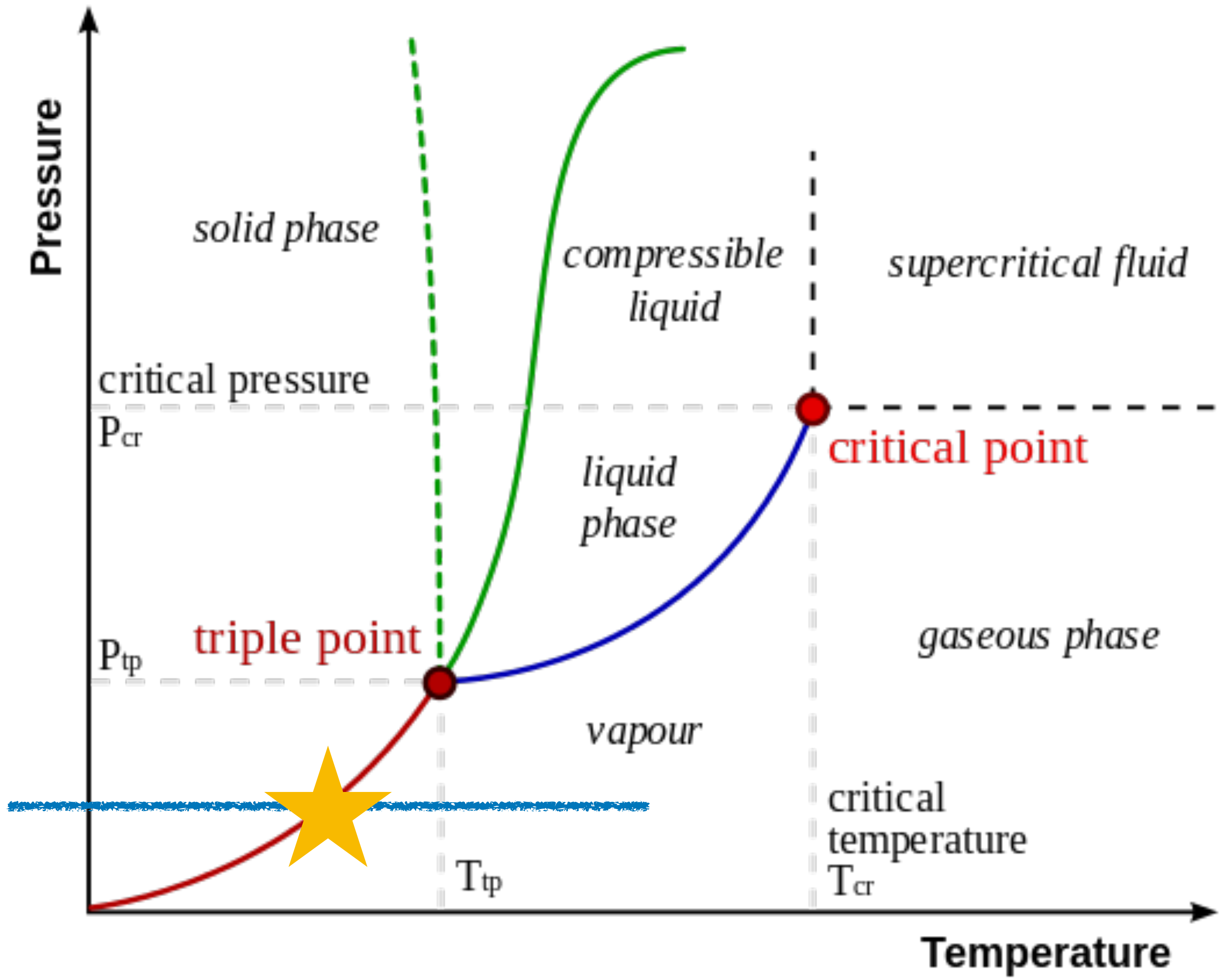




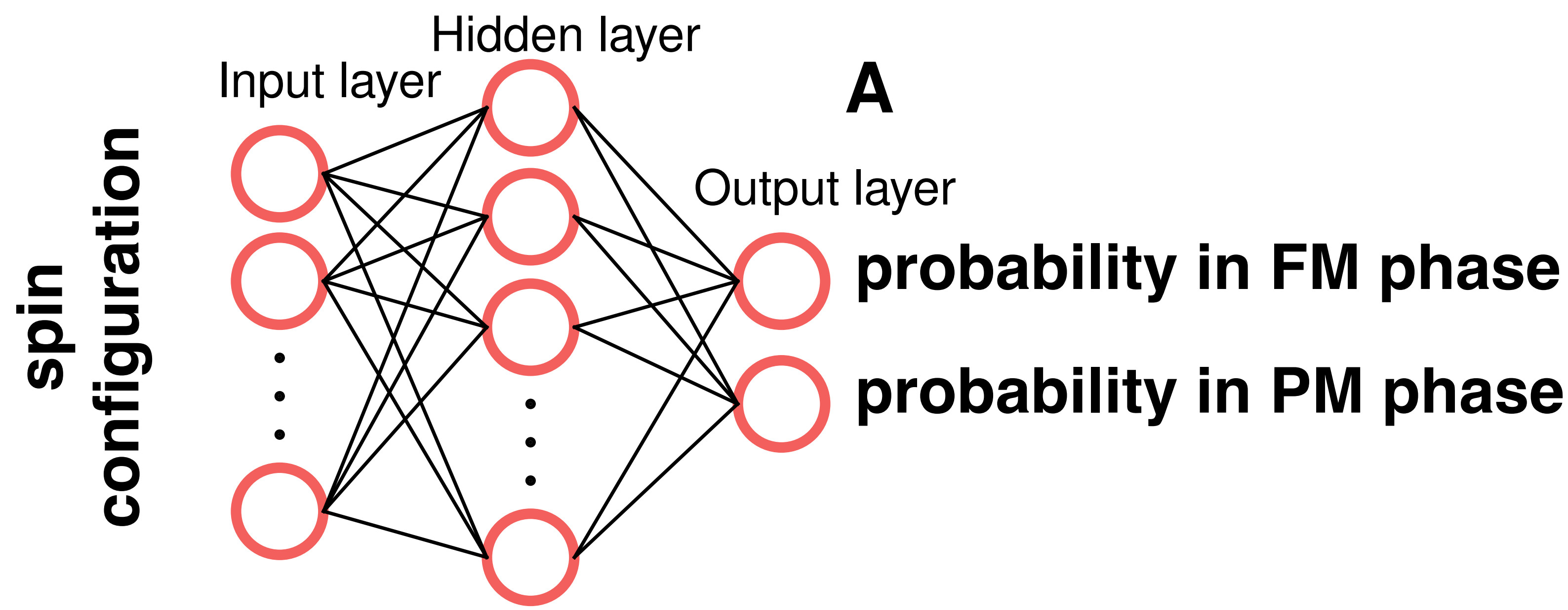
"Class" in Physics — Phase of Matter



"Class" in Physics — Phase of Matter



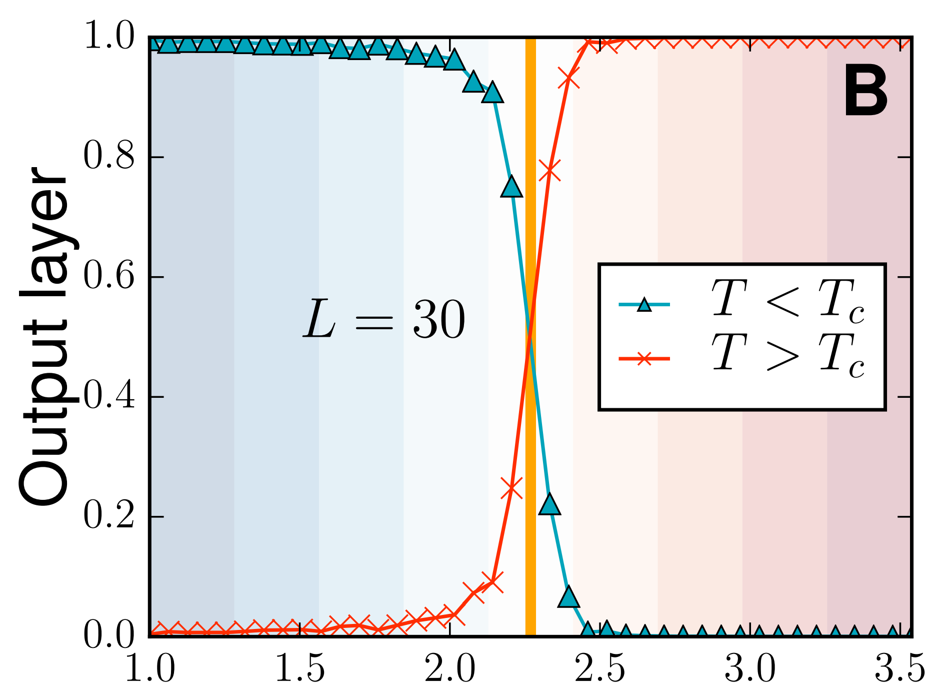
Learning Phase Transitions



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

FM T_c PM

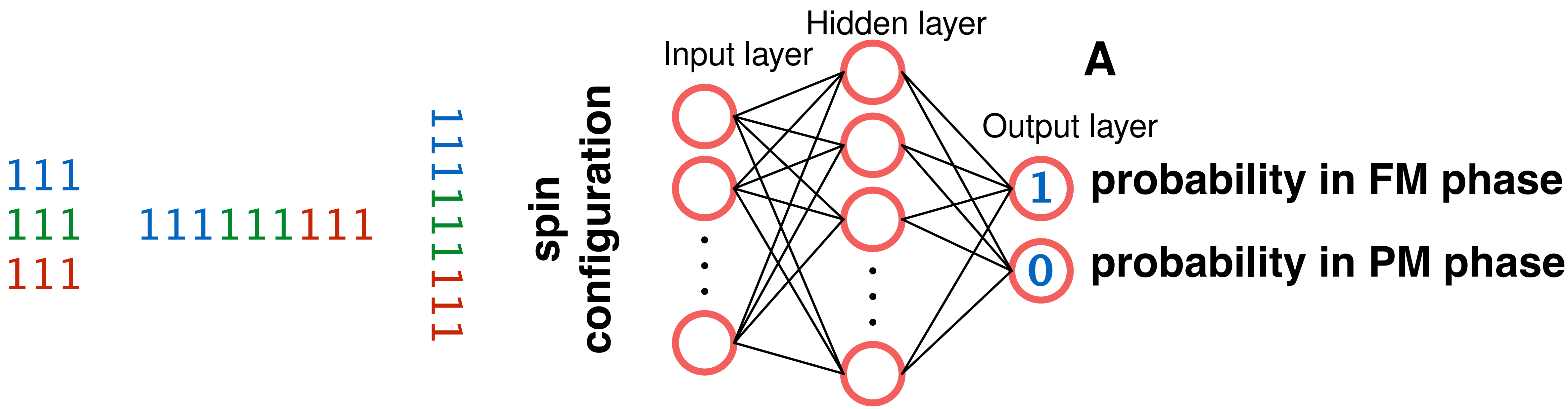
T



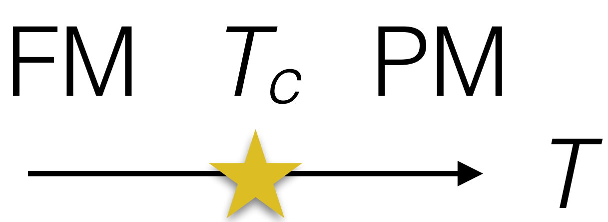
- Model: Neural Network
- Training: Supervised

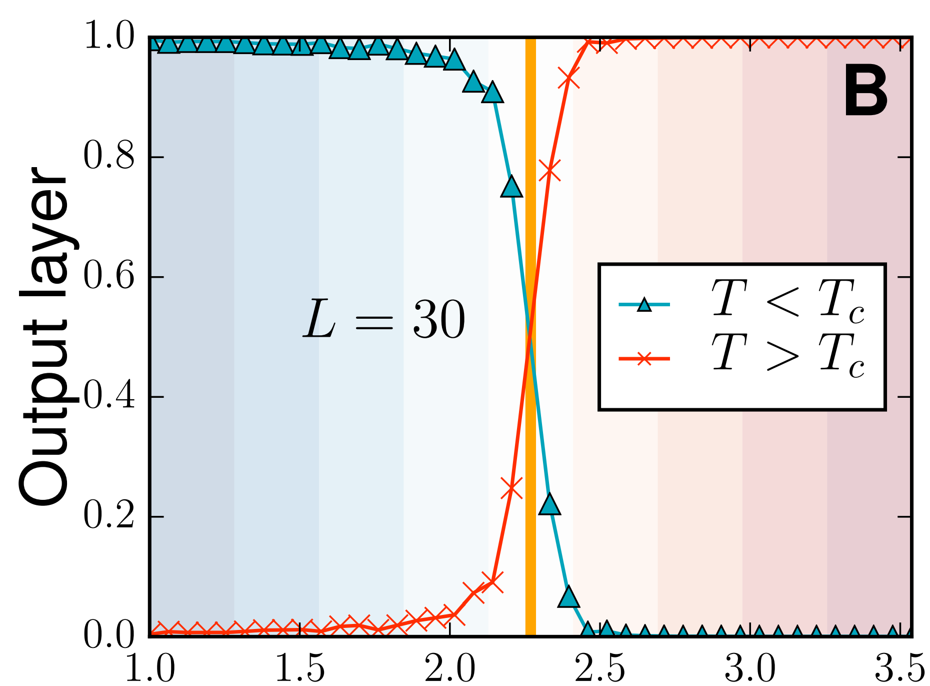
J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017)

Learning Phase Transitions



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

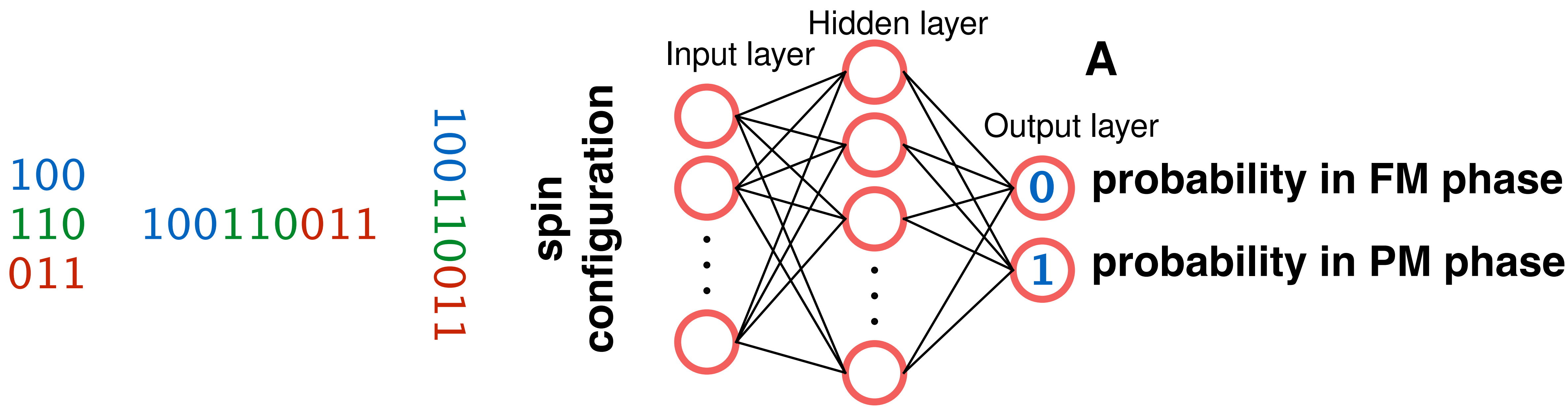




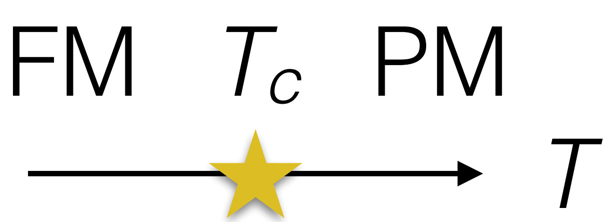
- Model: Neural Network
- Training: Supervised

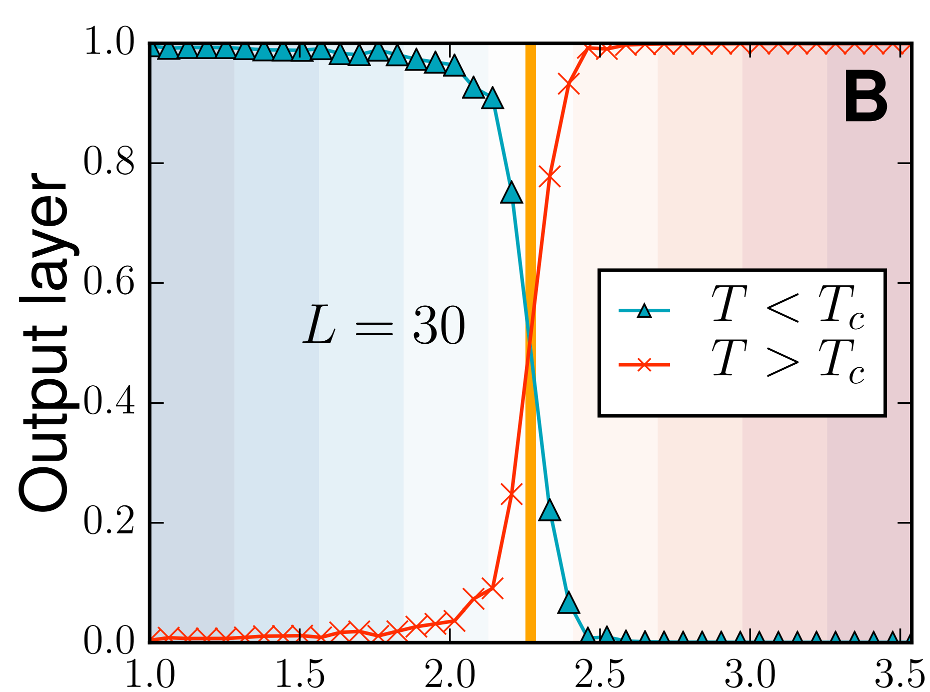
J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017)

Learning Phase Transitions



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

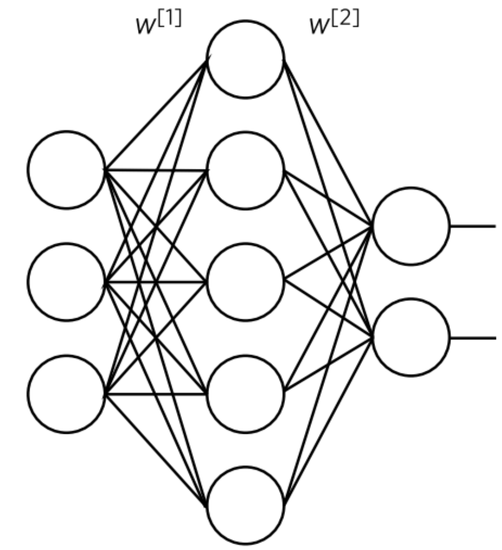




- Model: Neural Network
- Training: Supervised

J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017)

Training — Supervised



back-propagation

$$\Delta W = -\alpha\, \partial C / \partial W$$

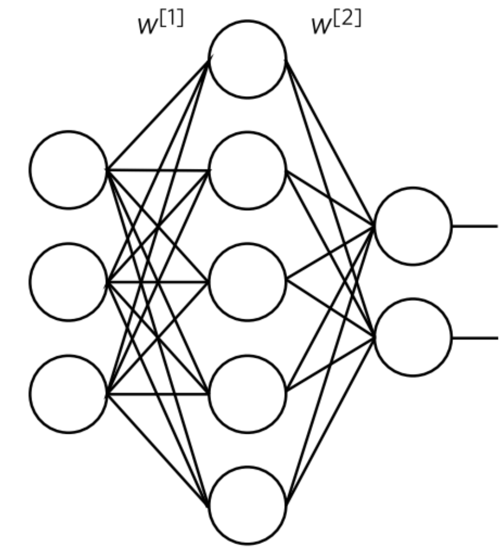
- Aim: Tune the weights, such that the NN reproduces **known config.-phase pairs.**
- Method: Reduce the cost function.

Cross entropy loss:

$$C(\vec{y}, \vec{y}') = -\log \vec{y} \cdot \vec{y}' - \log(1 - \vec{y}) \cdot (1 - \vec{y}')$$

- y **:** NN output for the input.
- y’**:** known result for the same input.
- Minimized when y=y’.

Training — Supervised



back-propagation

$$\Delta W = -\alpha\, \partial C / \partial W$$

- Aim: Tune the weights, such that the NN reproduces **known config.-phase pairs.**
- Method: Reduce the cost function.

Cross entropy loss:

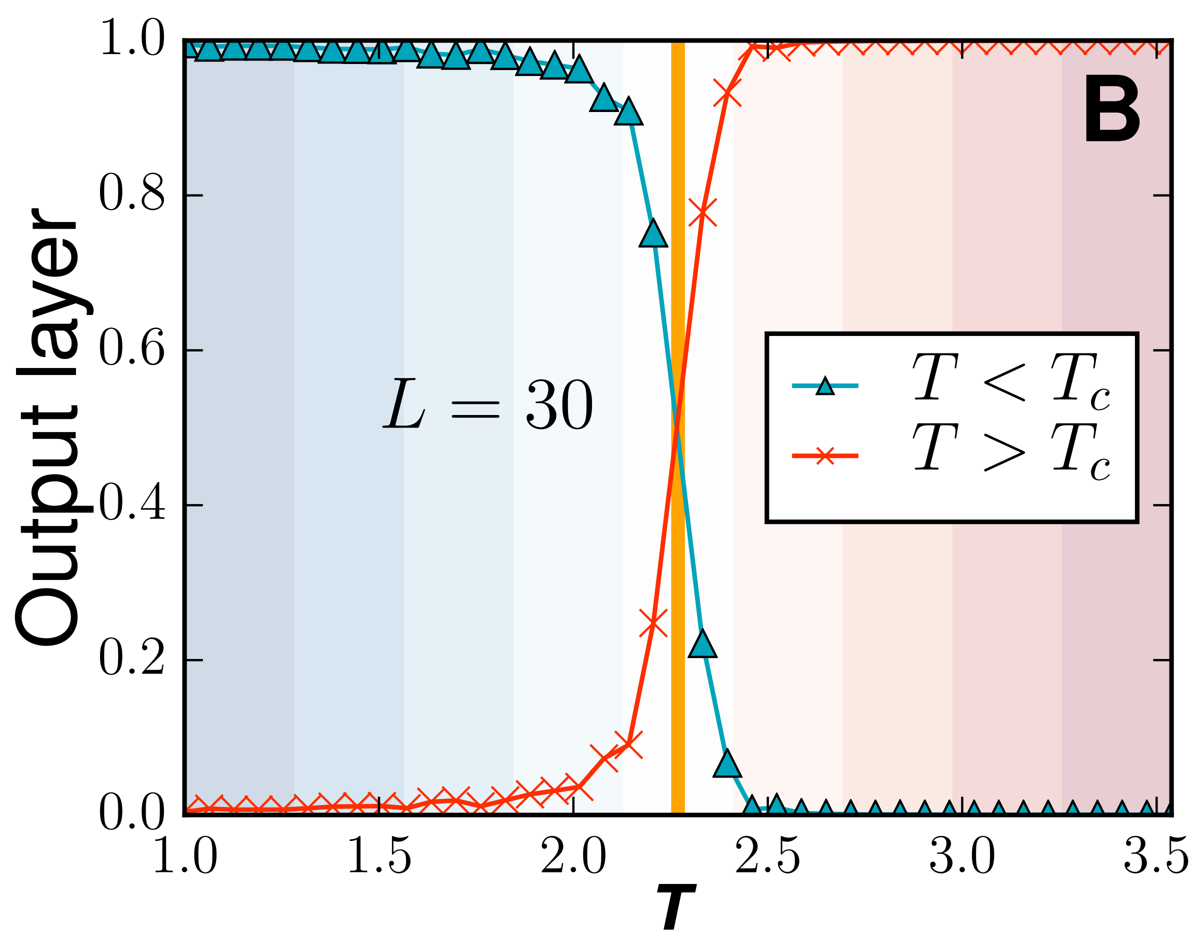
$$C(\vec{y}, \vec{y}') = -\log \vec{y} \cdot \vec{y}' - \log(1 - \vec{y}) \cdot (1 - \vec{y}')$$

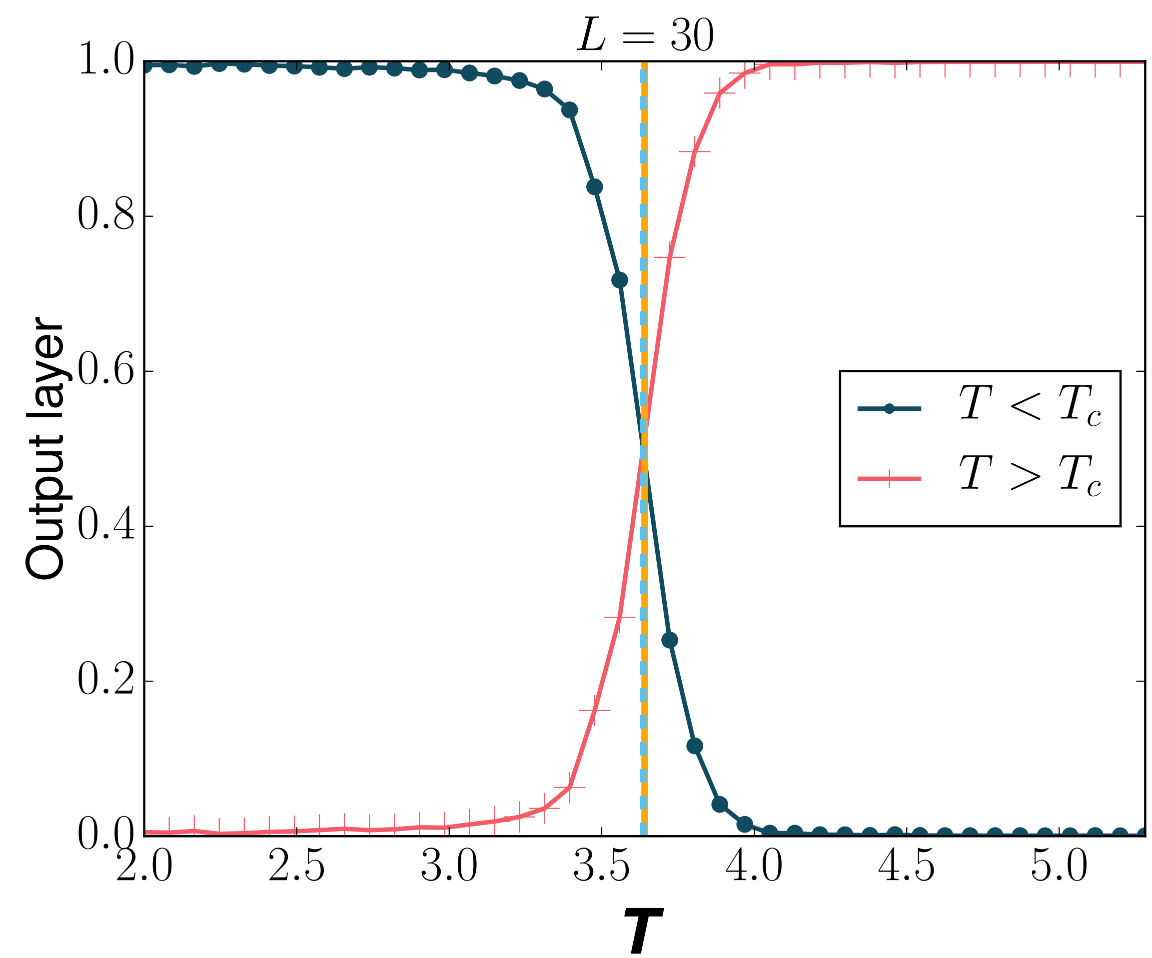
- y **:** NN output for the input.
- y'**:** known result for the same input.
- Minimized when y=y'.

Only reproduces (and not perfectly) known facts?

Generalization（举一反三）

J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017)





Trained on square lattice.

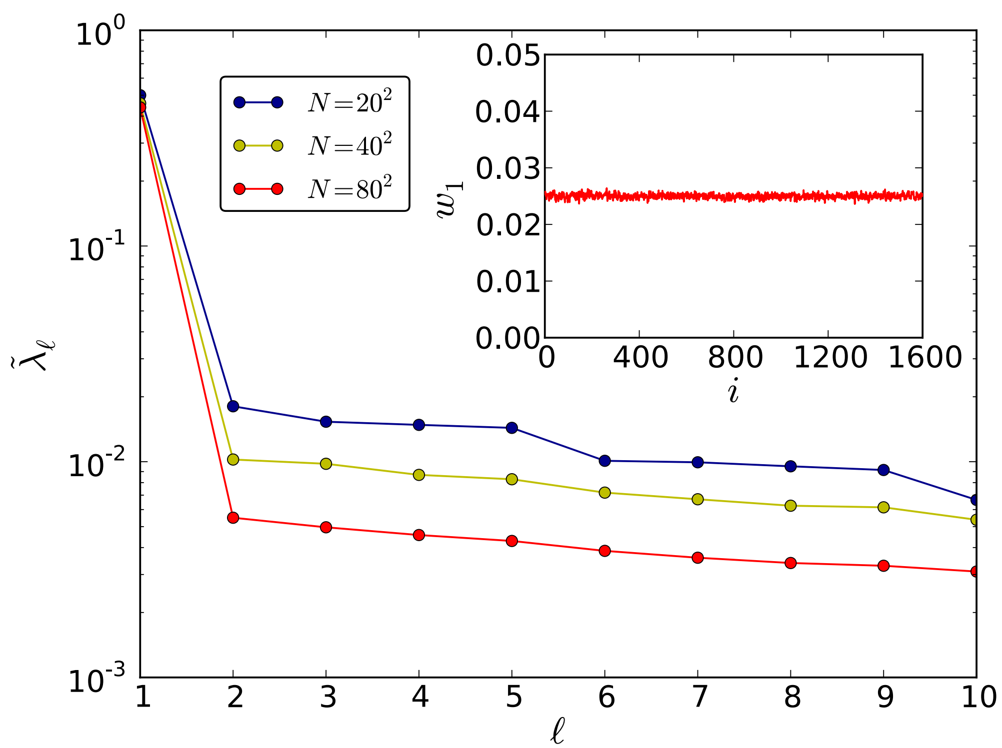
Generalized to triangular lattice.

Learning Phase Transitions — Unsupervised

Principal component analysis:

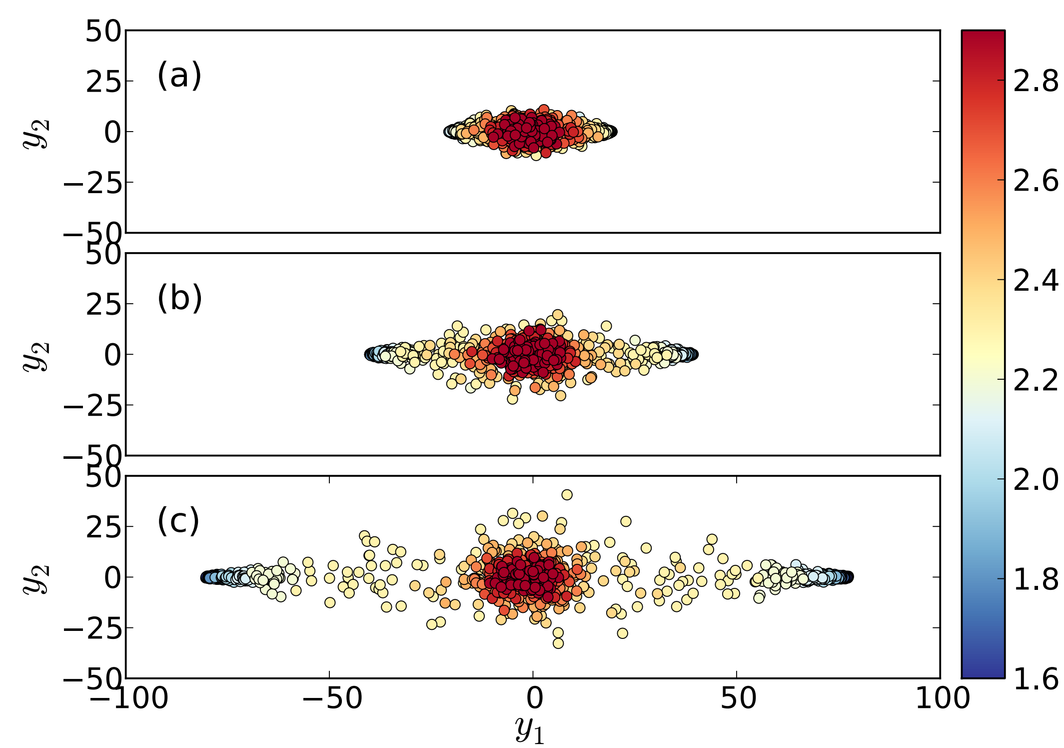
Find the important directions in the high-dimensional dataset space.

Lei Wang, Phys. Rev. B **94**, 195105 (2016)



$$\mathrm{diag}(X^T X)$$

X: row vectors of data



project data to the first two principal directions

For Ising data the first principal direction is (1,1,1…).

Comparison of Two Training Methods

	supervised	unsupervised (PCA)
pros	expressive power	no knowledge needed
cons	knowledge needed	linear OP

Comparison of Two Training Methods

	supervised	unsupervised (PCA)
pros	expressive power	no knowledge needed
cons	knowledge needed	linear OP

Confusion Scheme

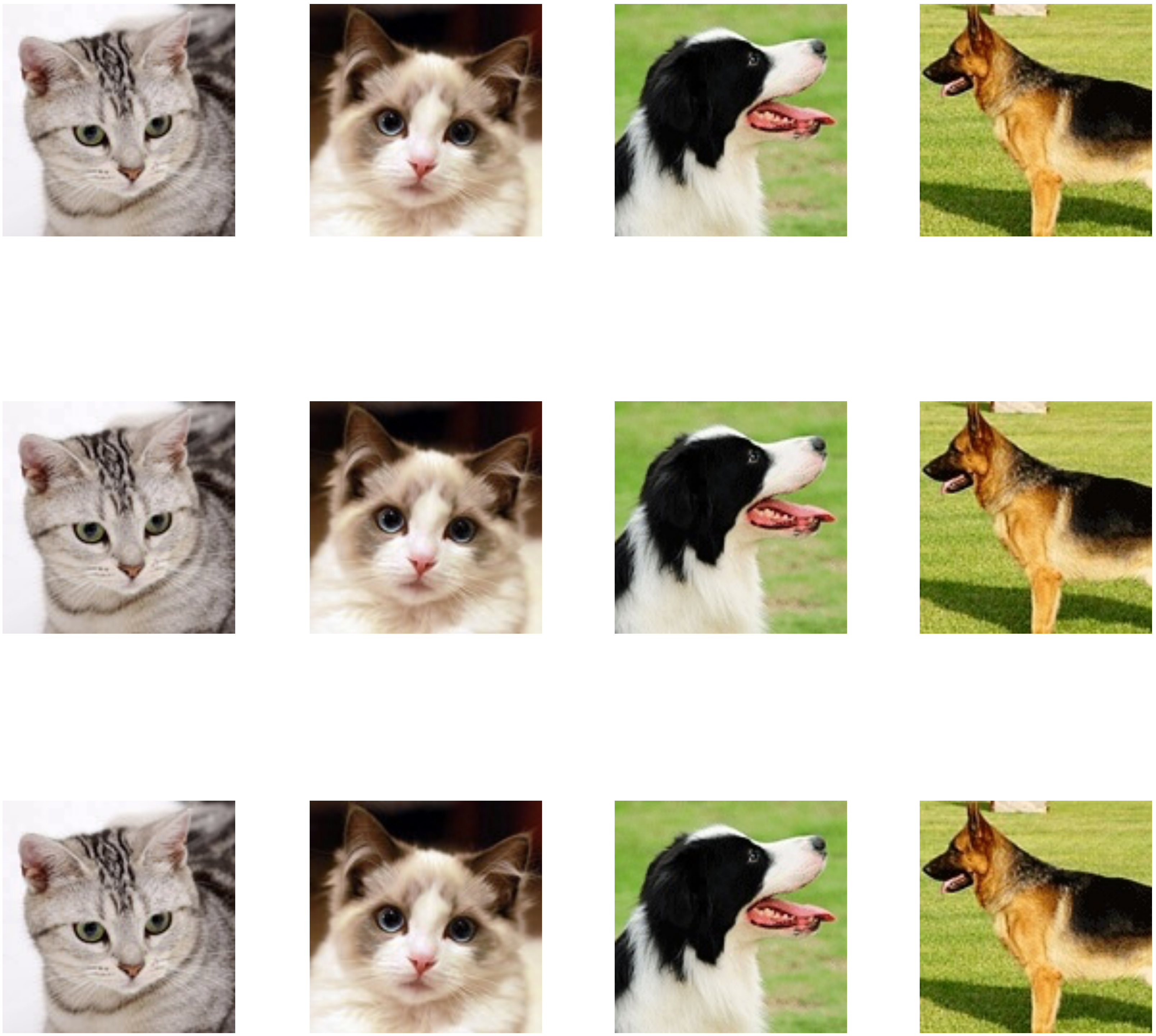
In Between, the Confusion Scheme

Confusion scheme:

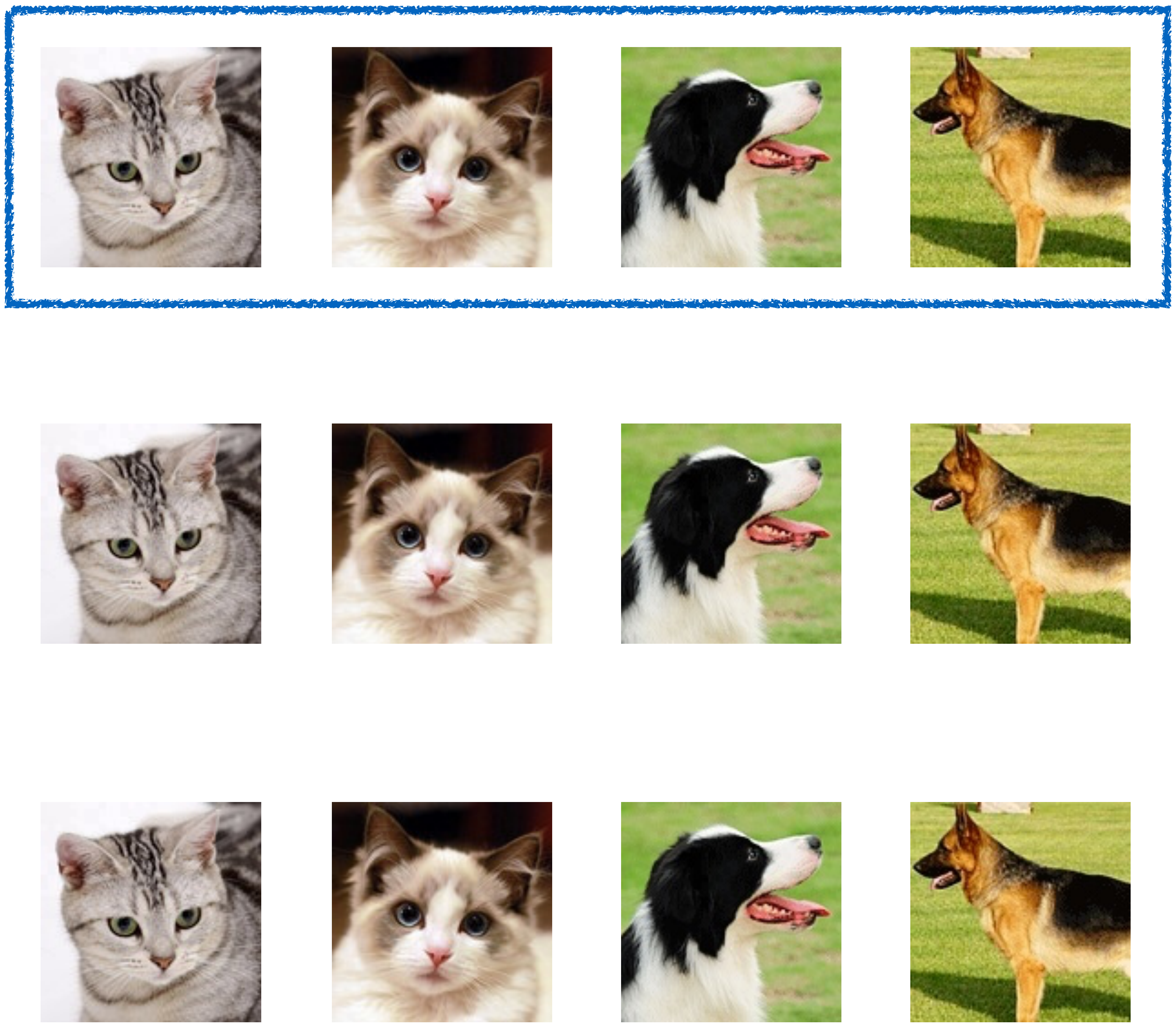
- Randomly **guess** the transition point.
- Train neutral network **for this guess,** get optimized performance.
- The right transition point: **the guess with the least "confusion".**

E. P. L. van Nieuwenburg, YHL, and S. D. Huber, Nat. Phys. **13**, 435 (2017)

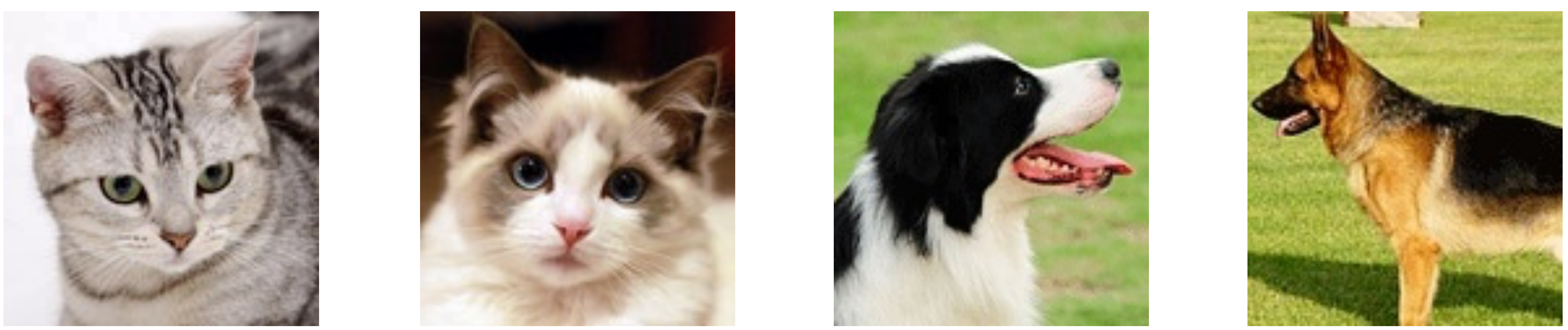
"Animal Transition" on a Line

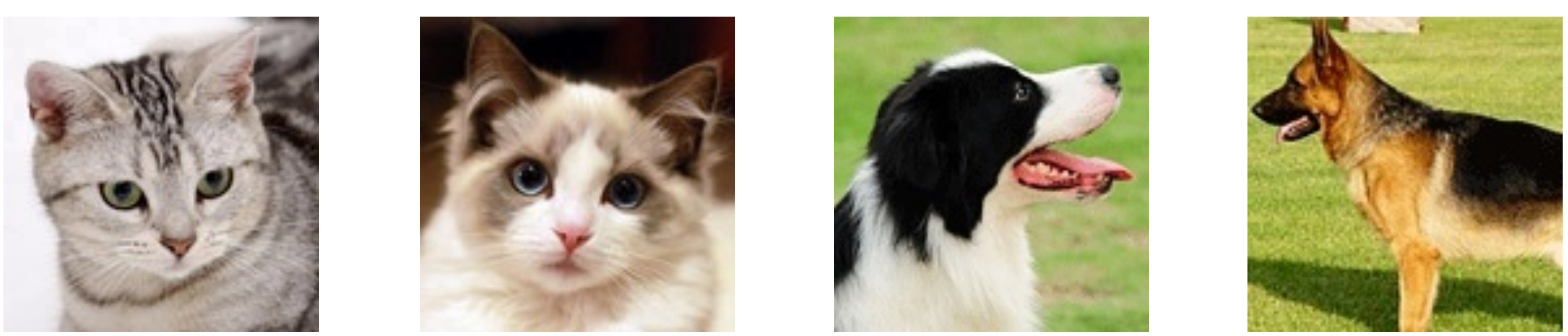


"Animal Transition" on a Line

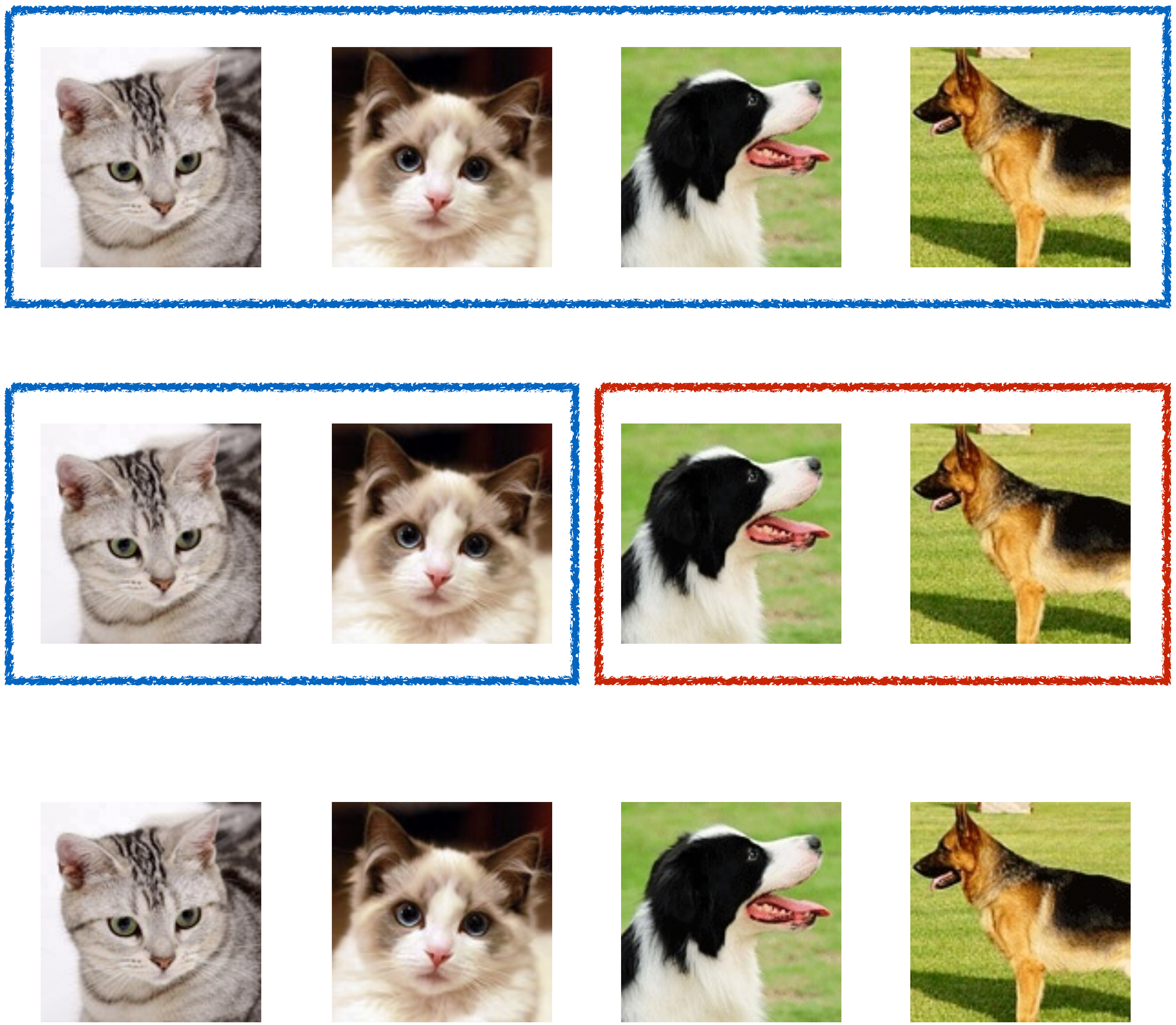


"Animal Transition" on a Line

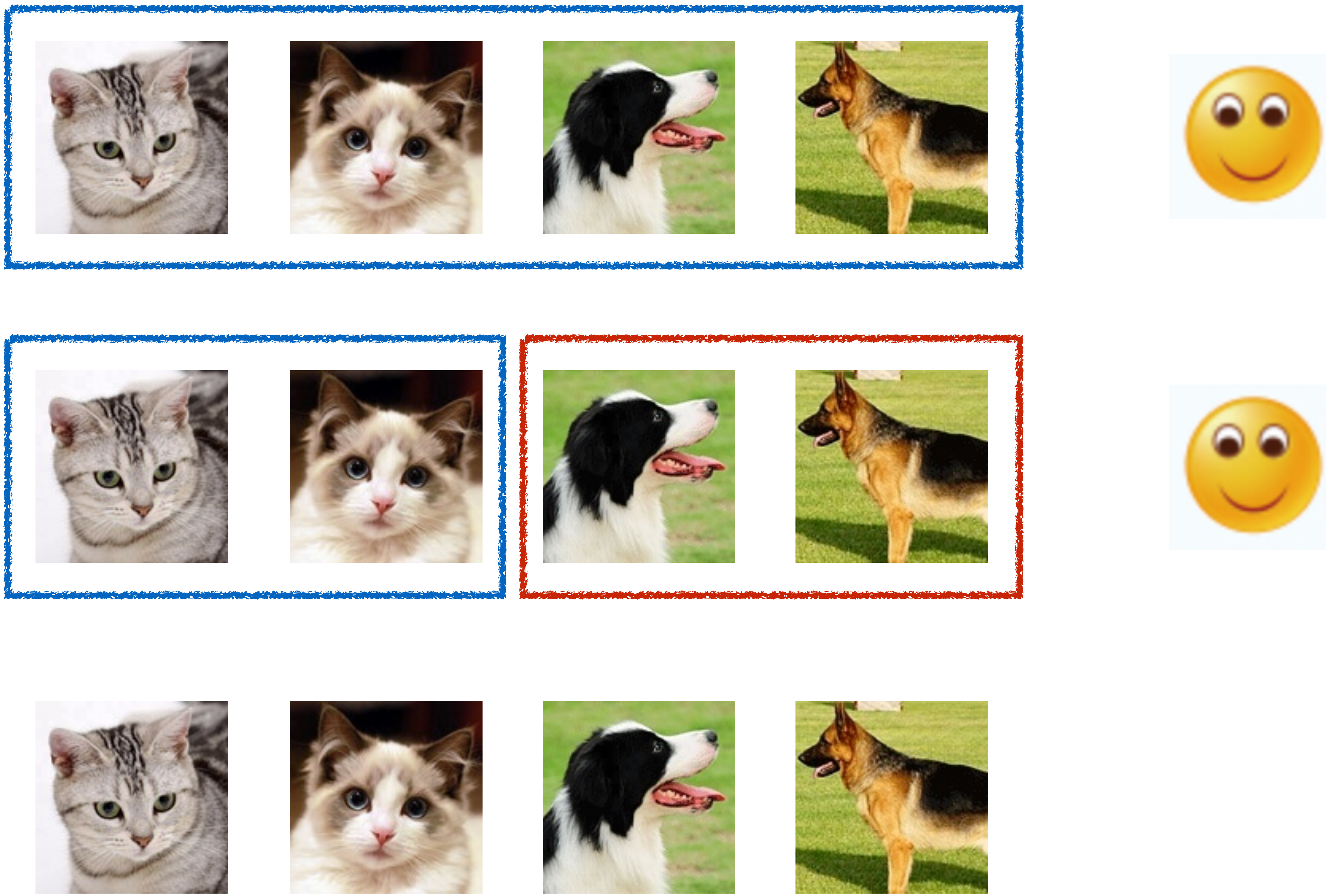




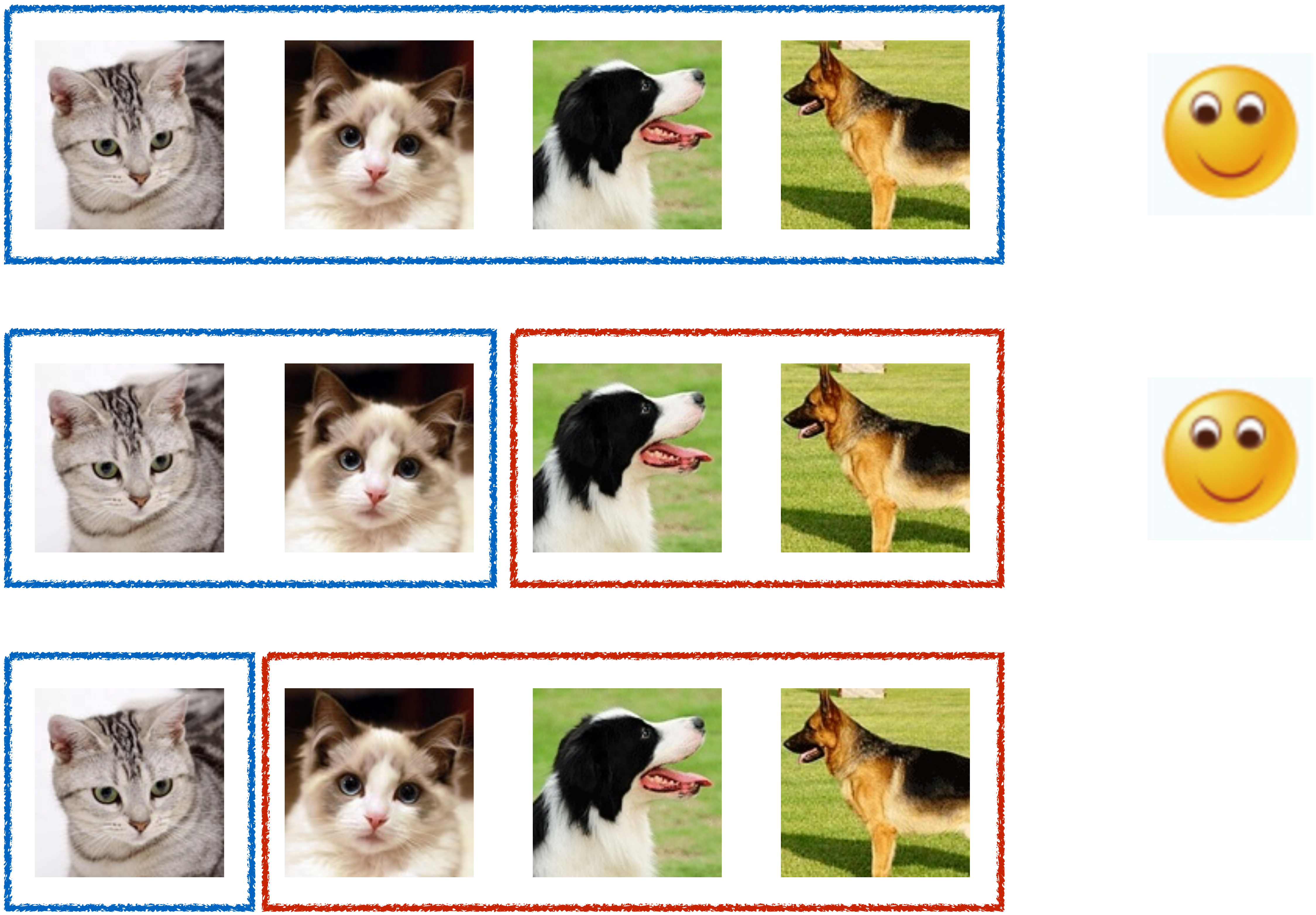
"Animal Transition" on a Line



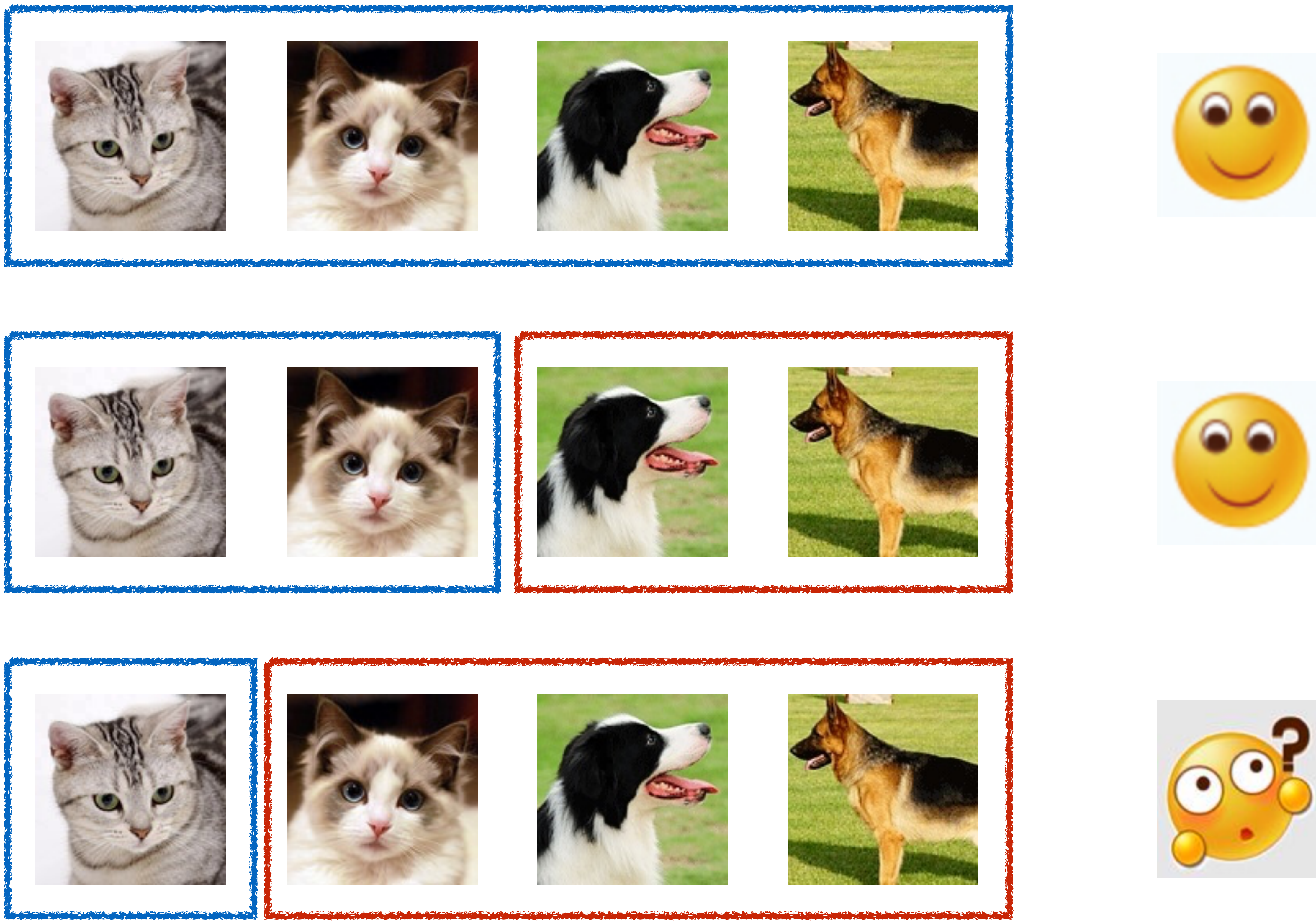
"Animal Transition" on a Line



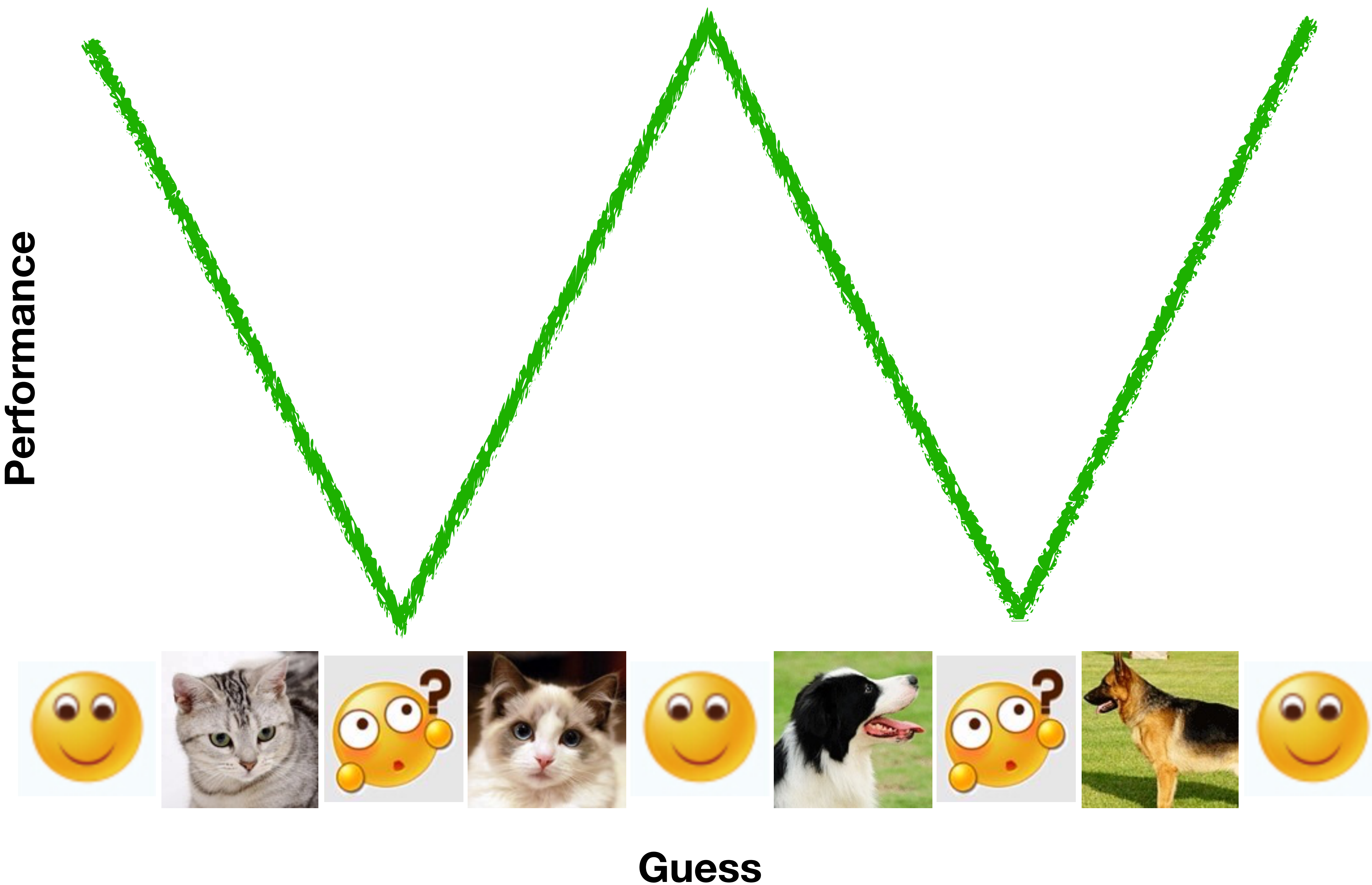
"Animal Transition" on a Line



"Animal Transition" on a Line



The Characteristic W-Shape

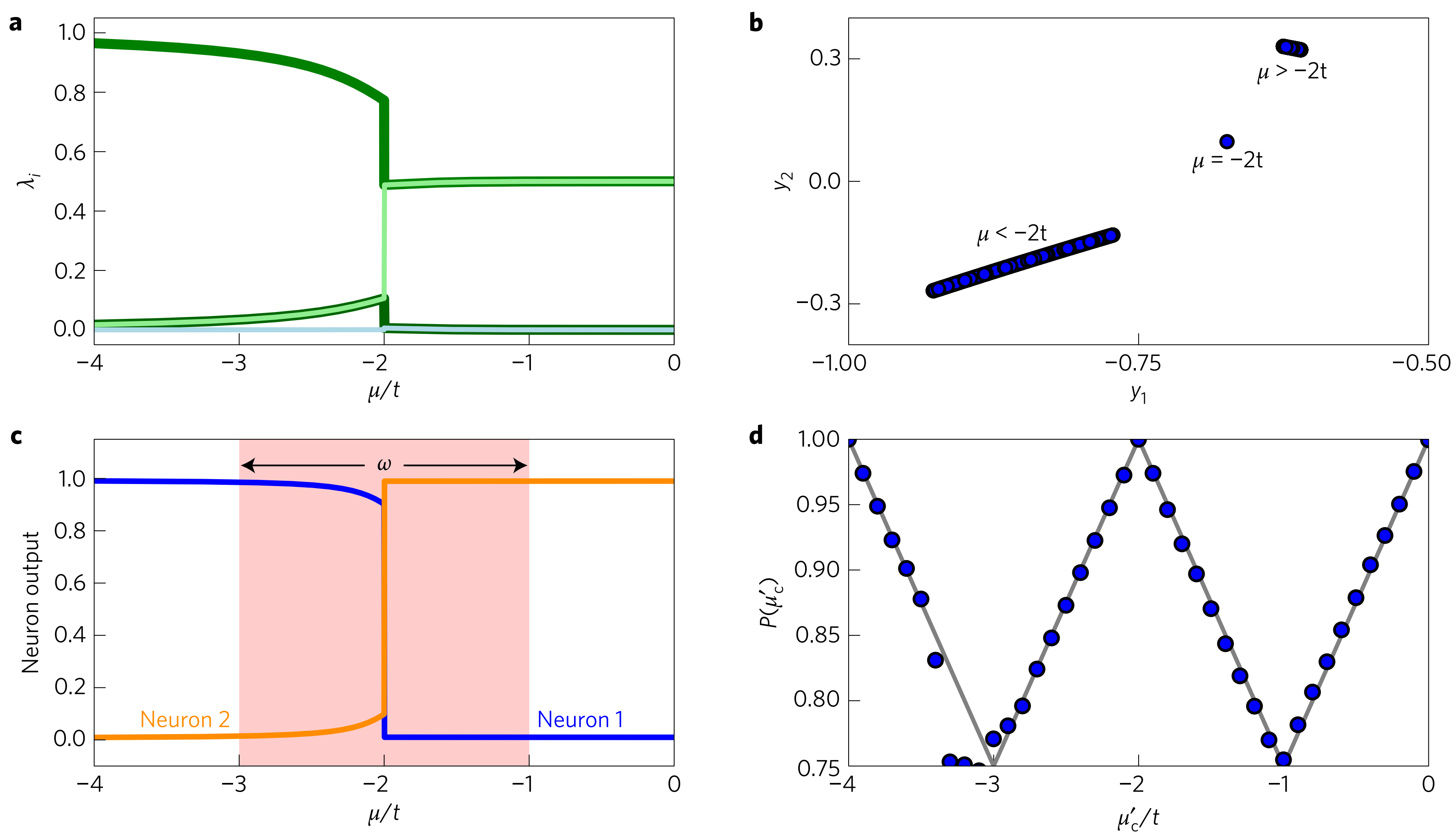


The 1D Kitaev Chain

$$\rho_A = \mathrm{tr}_B(\rho_{AB}), \mathcal{H} = -\log \rho_A$$

Topological phase transition (degeneracy structure of entanglement Hamiltonian)

$$\hat{H} = -t \sum_{i=1}^{L} (\hat{c}_{i+1}^{\dagger}\hat{c}_i + \hat{c}_{i+1}\hat{c}_i + h.c.) - \mu \sum_{i=1}^{L} \hat{c}_i^{\dagger}\hat{c}_i$$



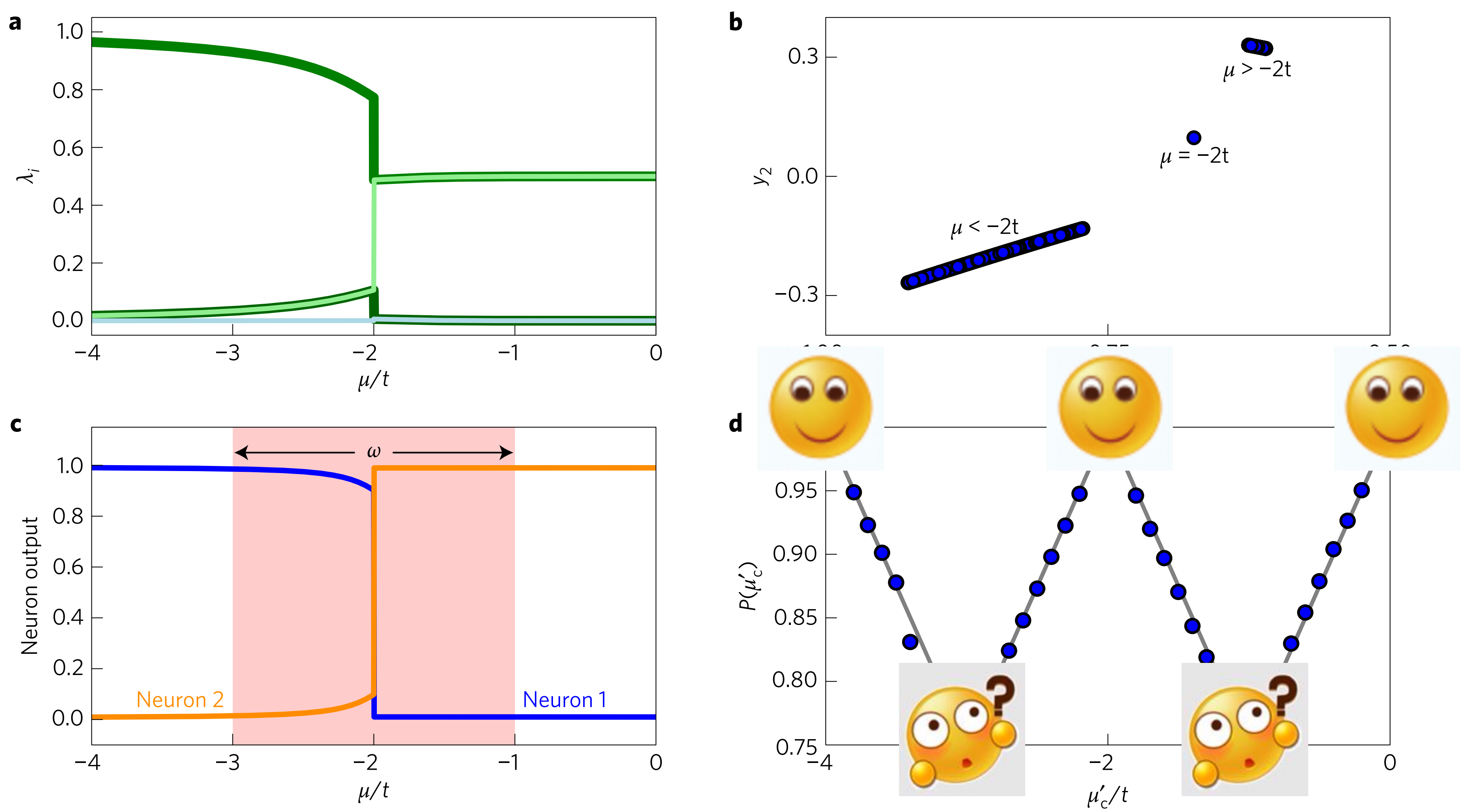
Input data: entanglement spectrum

The 1D Kitaev Chain

$$\rho_A = \mathrm{tr}_B(\rho_{AB}), \mathcal{H} = -\log \rho_A$$

Topological phase transition (degeneracy structure of entanglement Hamiltonian)

$$\hat{H} = -t \sum_{i=1}^{L} (\hat{c}_{i+1}^{\dagger} \hat{c}_i + \hat{c}_{i+1} \hat{c}_i + h.c.) - \mu \sum_{i=1}^{L} \hat{c}_i^{\dagger} \hat{c}_i$$

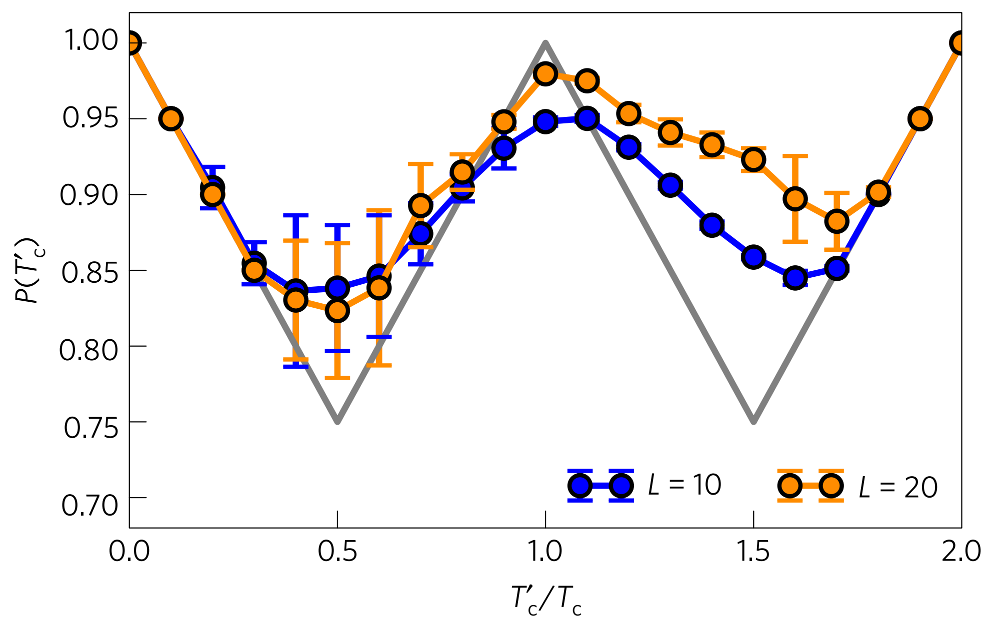


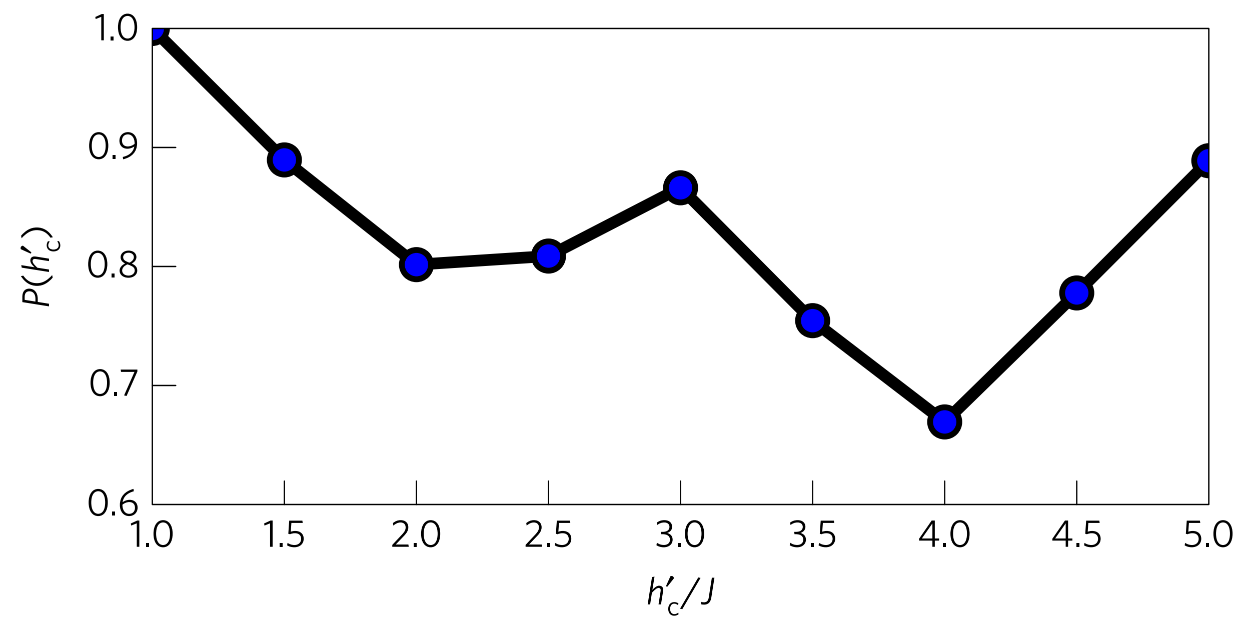
Input data: entanglement spectrum

Other Kinds of Transitions

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

$$H = J \sum_{i=1}^{L} S_i \cdot S_{i+1} + \sum_{\alpha=x,y,z} \sum_{i=1}^{L} h_i^\alpha S_i^\alpha$$





On Monte Carlo configuration for 2D Ising model

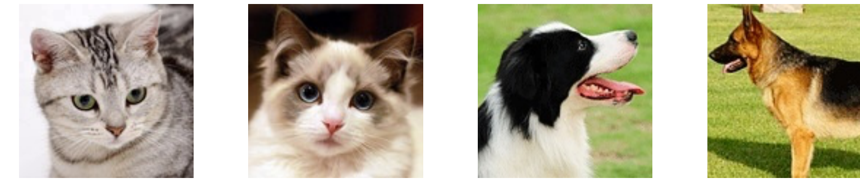
On entanglement spectrum for random-field Heisenberg chain

Symmetry breaking (local order parameter)

Many-body localization (level statistics of entanglement Hamiltonian)

The brute-force sweep for the guess is expensive.

Recap



This is strange, no?

Principle of continuity:

Features are ordered by tuning parameters, and most of the time, the label does not change.

Phase-transition classification is very special, because the number of possible ways to assign phases is small.

Unordered data in feature space:	**Ordered data in parameter space:**
$\vec{d}_1$ ... $\vec{d}_N$ $\vec{d}_2$ $\vec{d}_3$	$\vec{d}_1$ $\vec{d}_2$ $\vec{d}_3$ ... $\vec{d}_N$ → p
2^N	$N+1$

Self-Learning Scheme

ML Search for Optimal Guess

Promote the human guess to a guesser agent.

YHL and E. P. L. van Nieuwenburg, arXiv:1706.08111

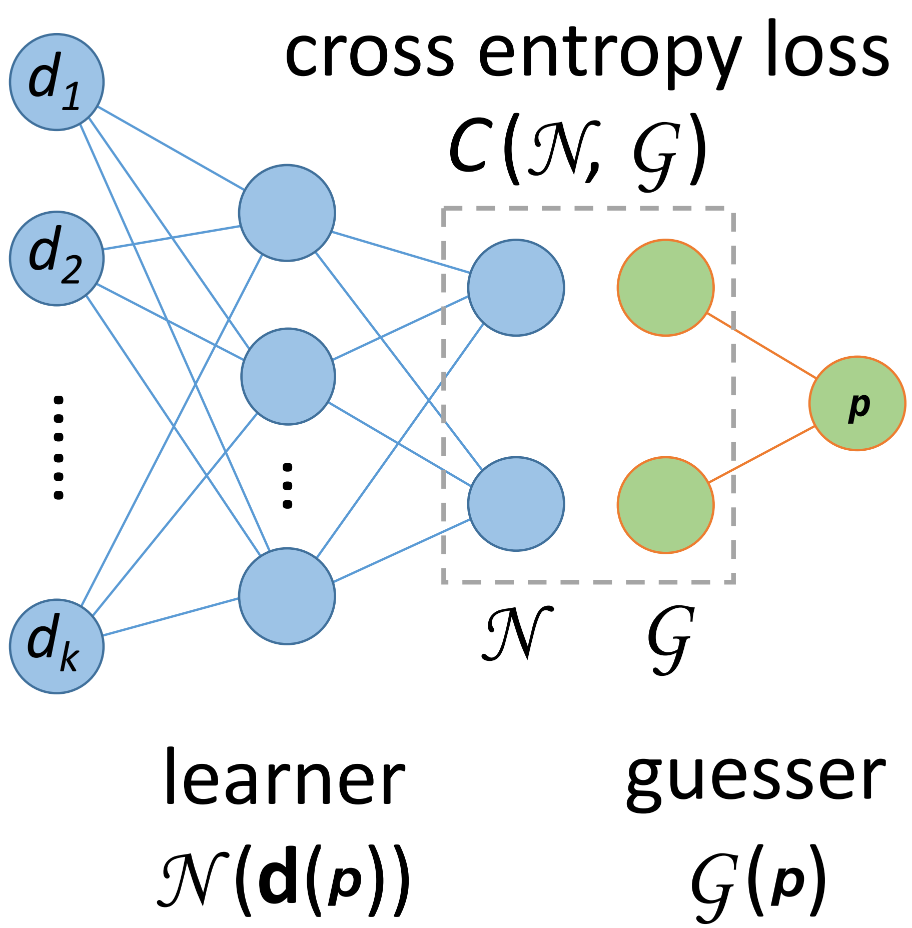
Cooperative two-nets scheme:

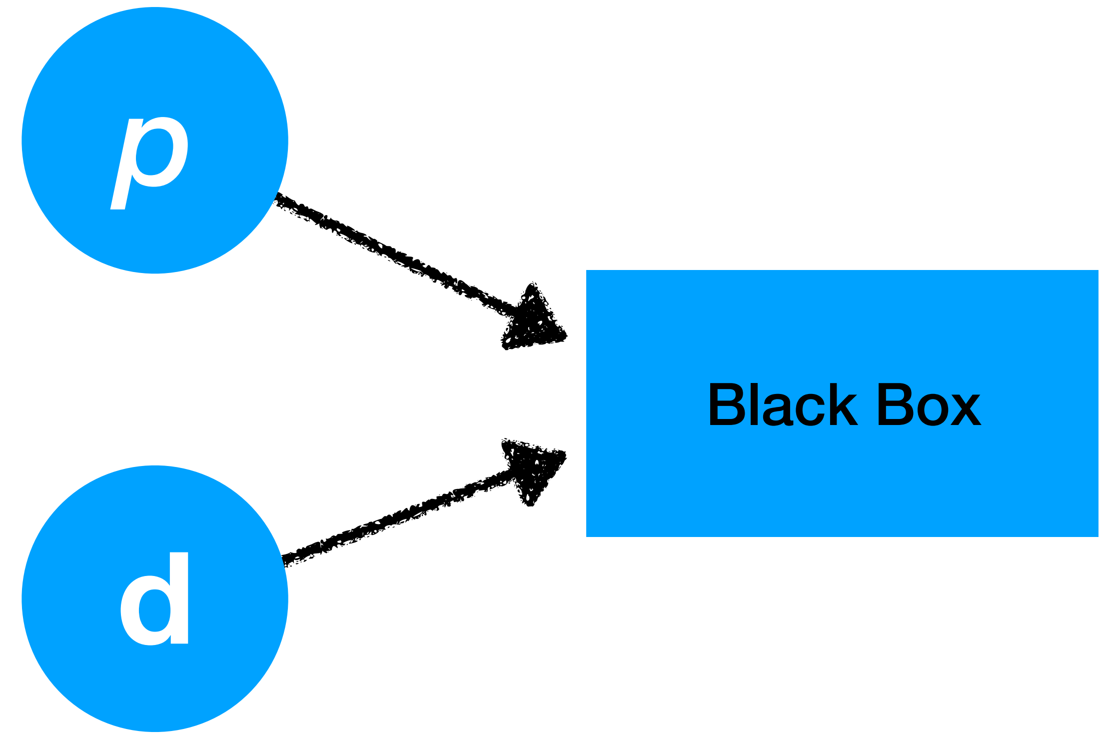
Proposes the guess
Wants to be a better teacher

Guesser **Learner**

Responds with learning performance
Wants to be a better student

Self-Learning Scheme

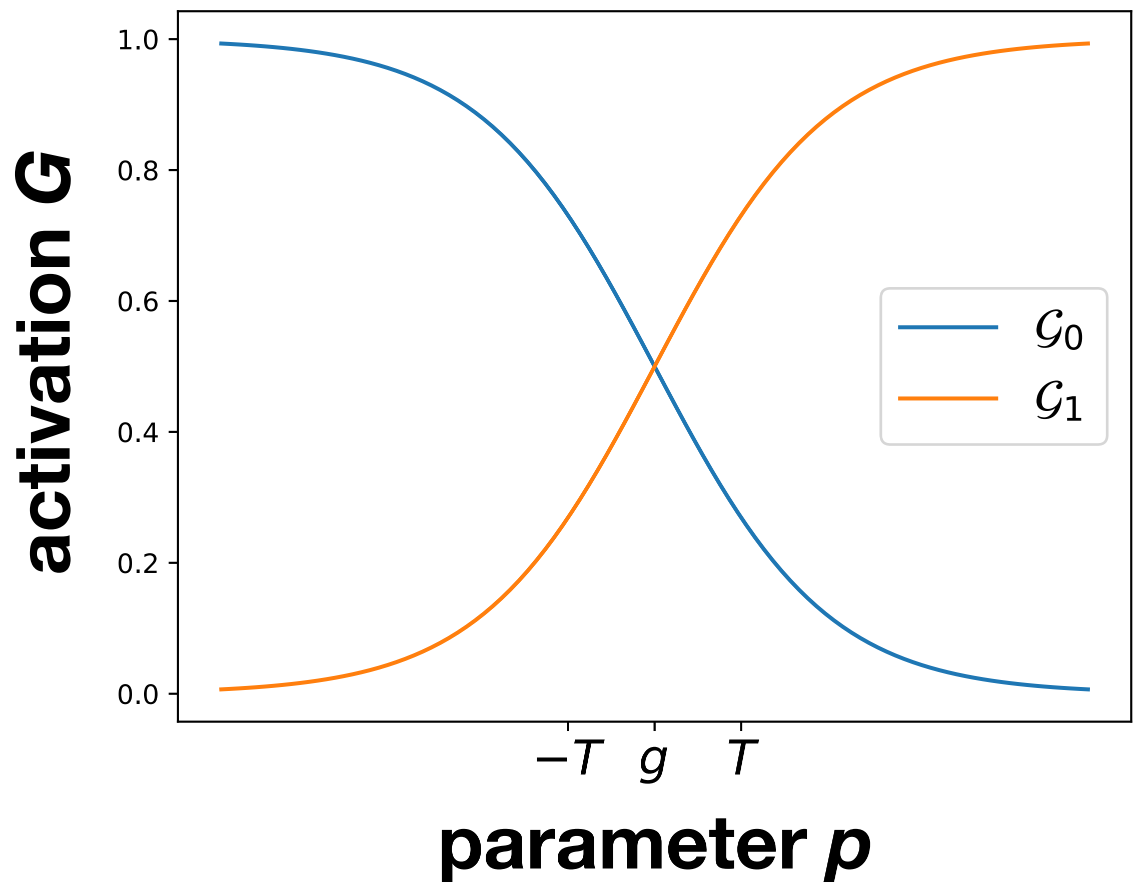




- **Better teacher**
- **Better student**
- **Better academy**

- **Unsupervised learning**
- **Self-supervised learning**
- **Self-learning**

The Guesser



$$\begin{aligned}\mathcal{G}_0(p) &= \sigma(-(p-g)/T),\\ \mathcal{G}_1(p) &= \sigma(+(p-g)/T).\end{aligned}$$

A logistic regression parametrized by:

- **a guess *g***
- **a sharpness *T***

The Learning Equations

Gradient descent:

$$\Delta g = -\alpha_g \frac{\partial C}{\partial g}, \Delta T = -\alpha_T \frac{\partial C}{\partial T}, \Delta W_{\mathcal{N}} = -\alpha_{W_{\mathcal{N}}} \frac{\partial C}{\partial W_{\mathcal{N}}}.$$

Cross-entropy cost/loss function:

$$C(\mathcal{N}, \mathcal{G}) = -\log \mathcal{N} \cdot \mathcal{G} - \log(1 - \mathcal{N}) \cdot (1 - \mathcal{G})$$

Chain rule:

$$\frac{\partial C}{\partial \mathcal{G}} = -\log \mathcal{N} + \log(1 - \mathcal{N}),$$

$$\frac{\partial \mathcal{G}_{0,1}}{\partial g} = -\frac{s_{0,1}}{4T \cosh^2 \left[(p - g)/2T\right]},$$

$$\frac{\partial \mathcal{G}}{\partial T} = \frac{p - g}{T} \frac{\partial \mathcal{G}}{\partial g},$$

The Learning Equations

Gradient descent:

$$\Delta g = -\alpha_g \frac{\partial C}{\partial g}, \Delta T = -\alpha_T \frac{\partial C}{\partial T}, \Delta W_{\mathcal{N}} = -\alpha_{W_{\mathcal{N}}} \frac{\partial C}{\partial W_{\mathcal{N}}}.$$

back-propagation

Cross-entropy cost/loss function:

$$C(\mathcal{N}, \mathcal{G}) = -\log \mathcal{N} \cdot \mathcal{G} - \log(1 - \mathcal{N}) \cdot (1 - \mathcal{G})$$

Chain rule:

$$\frac{\partial C}{\partial \mathcal{G}} = -\log \mathcal{N} + \log(1 - \mathcal{N}),$$

$$\frac{\partial \mathcal{G}_{0,1}}{\partial g} = -\frac{s_{0,1}}{4T \cosh^2 \left[(p-g)/2T\right]},$$

$$\frac{\partial \mathcal{G}}{\partial T} = \frac{p-g}{T} \frac{\partial \mathcal{G}}{\partial g},$$

The Learning Equations

Gradient descent:

$$\Delta g = -\alpha_g \frac{\partial C}{\partial g}, \Delta T = -\alpha_T \frac{\partial C}{\partial T}, \Delta W_{\mathcal{N}} = -\alpha_{W_{\mathcal{N}}} \frac{\partial C}{\partial W_{\mathcal{N}}}.$$

back-propagation

Cross-entropy cost/loss function:

$$C(\mathcal{N}, \mathcal{G}) = -\log \mathcal{N} \cdot \mathcal{G} - \log(1 - \mathcal{N}) \cdot (1 - \mathcal{G})$$

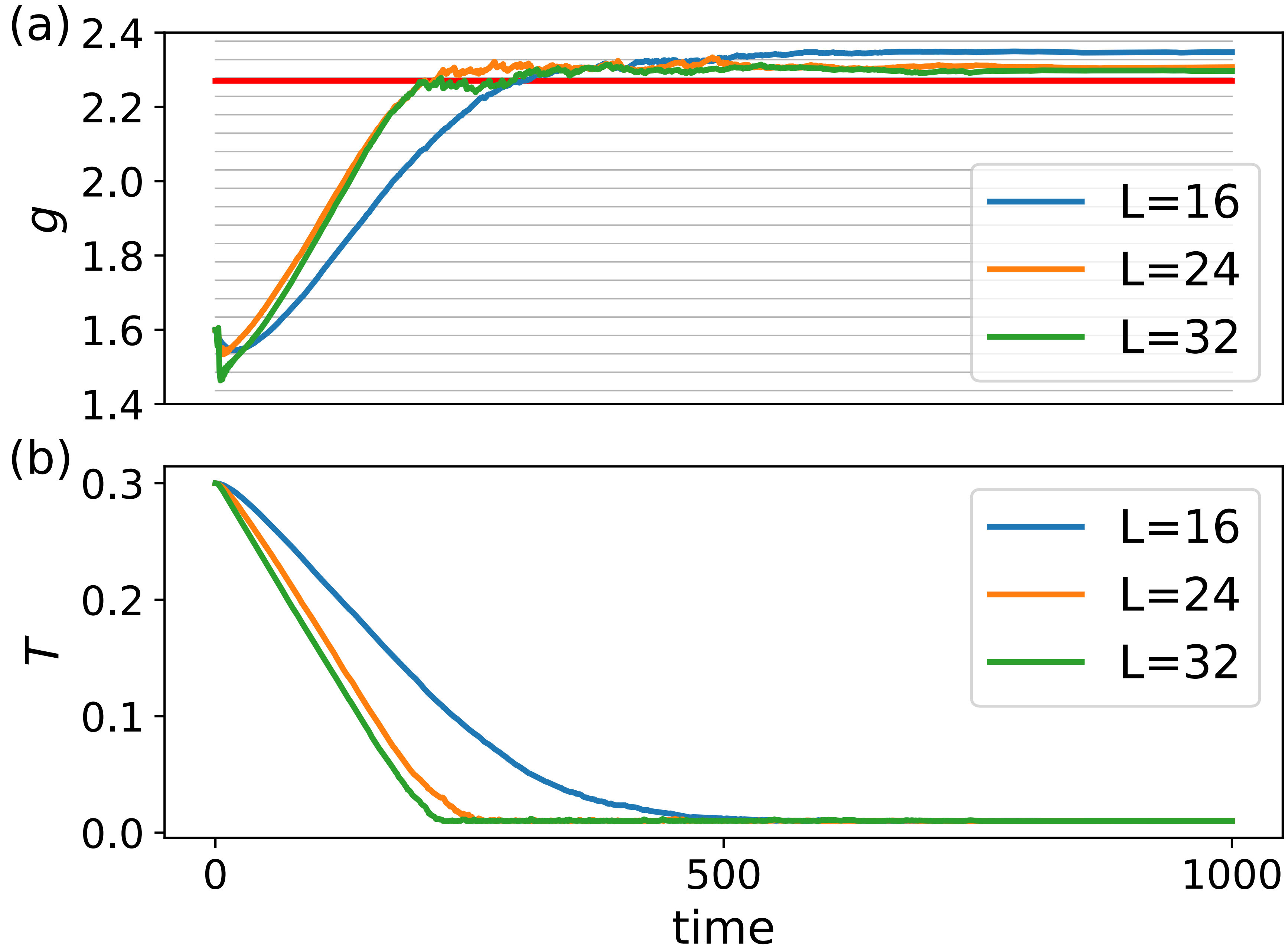
Chain rule:

$$\frac{\partial C}{\partial \mathcal{G}} = -\log \mathcal{N} + \log(1 - \mathcal{N}),$$

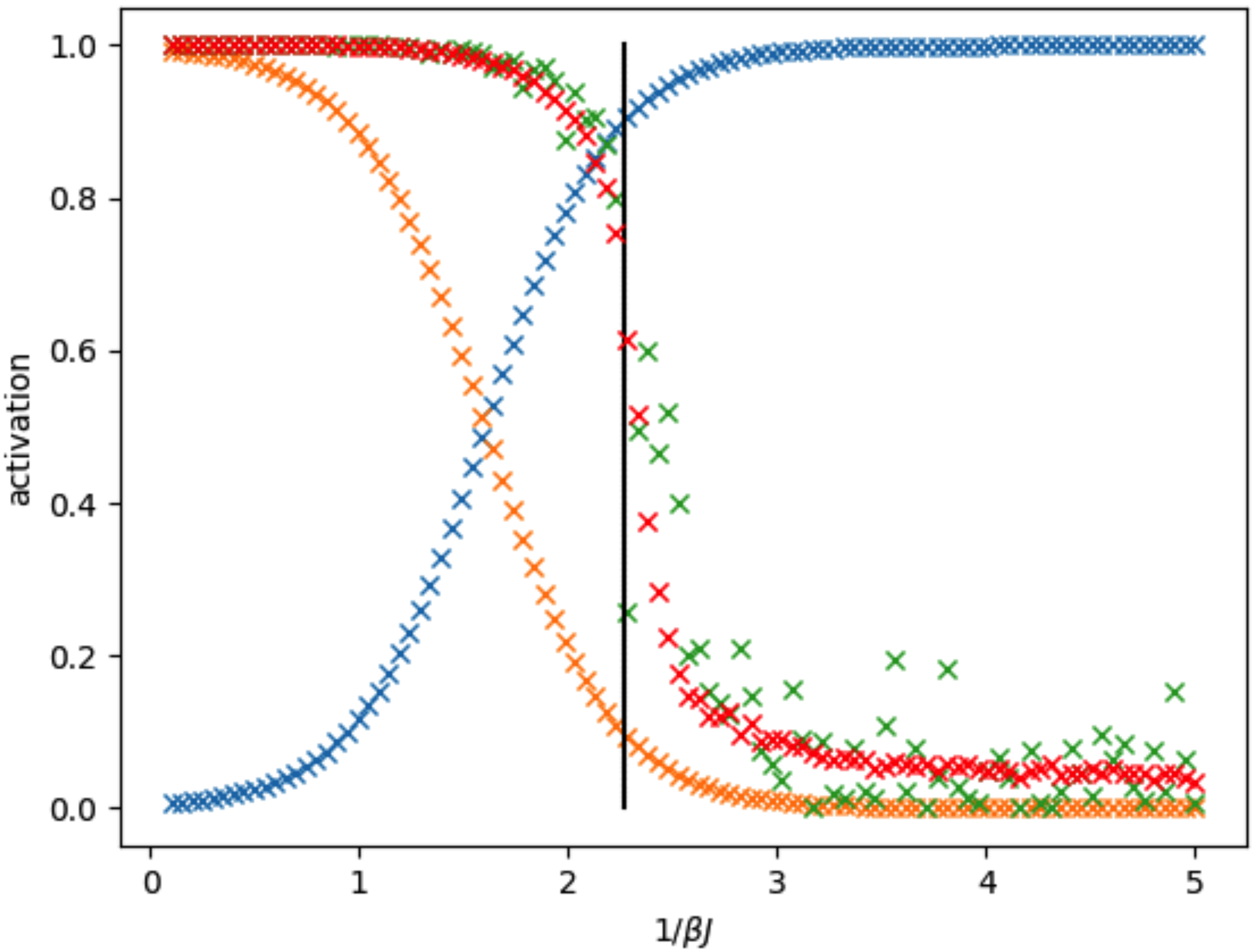
$$\frac{\partial \mathcal{G}_{0,1}}{\partial g} = -\frac{s_{0,1}}{4T \cosh^2 \left[(p-g)/2T\right]},$$

$$\frac{\partial \mathcal{G}}{\partial T} = \frac{p-g}{T} \frac{\partial \mathcal{G}}{\partial g},$$

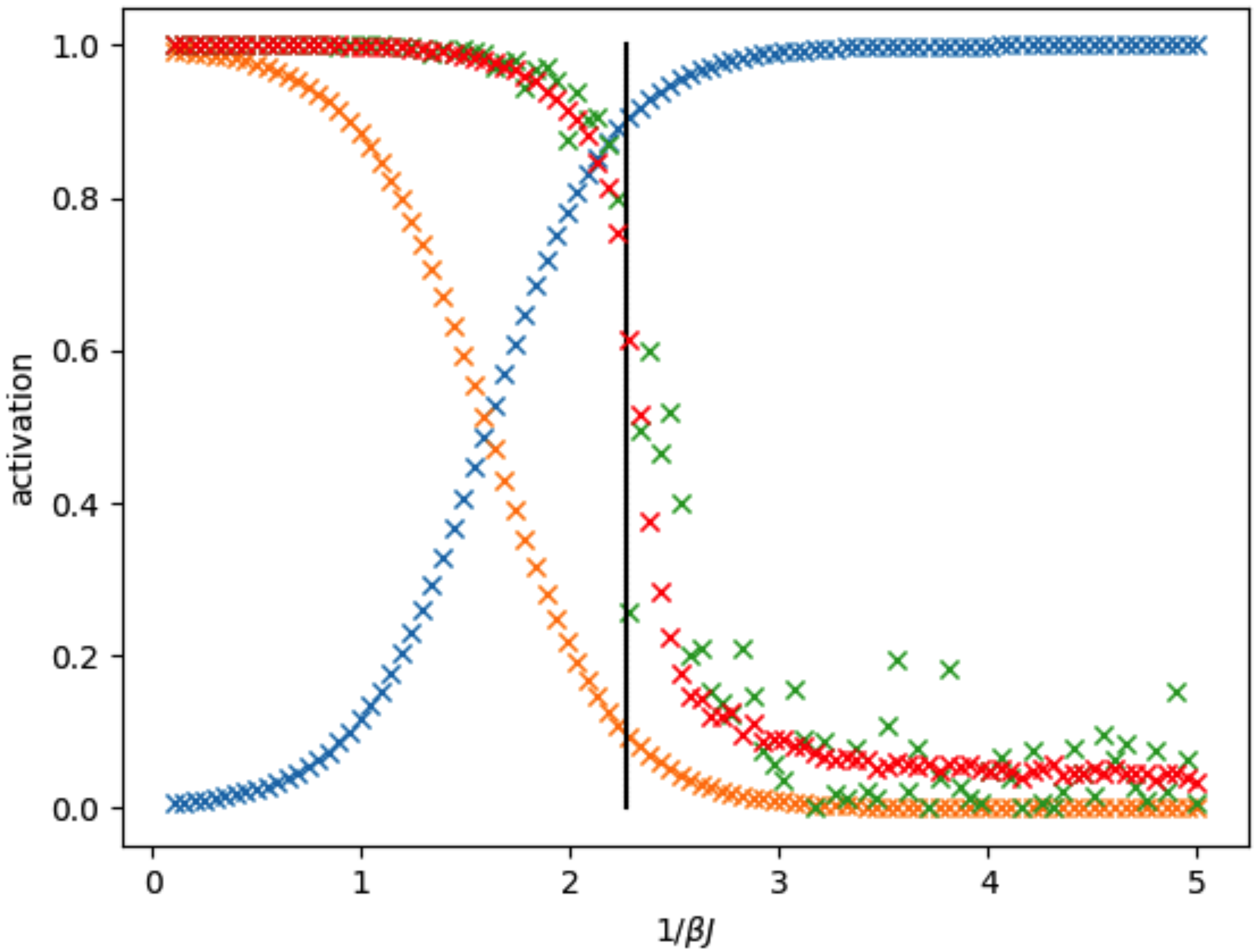
Self-Learning the Ising transition



Larger system size: more accurate and sharper.

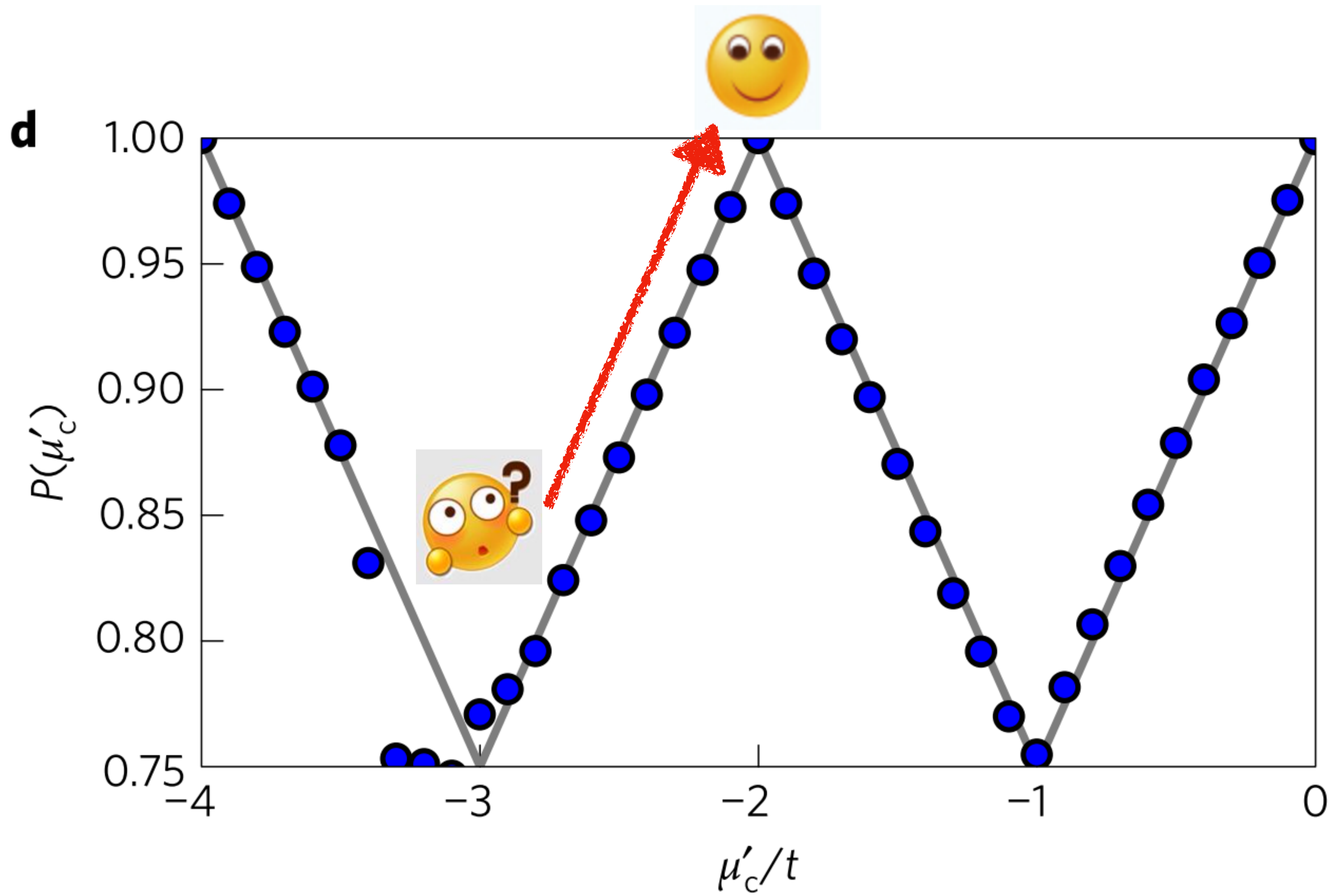


1.0
0.8
0.6
0.4
0.2
0.0
activation
0
1
2
3
4
5
$1/\beta J$



1.0
0.8
0.6
0.4
0.2
0.0
activation
0
1
2
3
4
5
1/βJ

What We Have Achieved

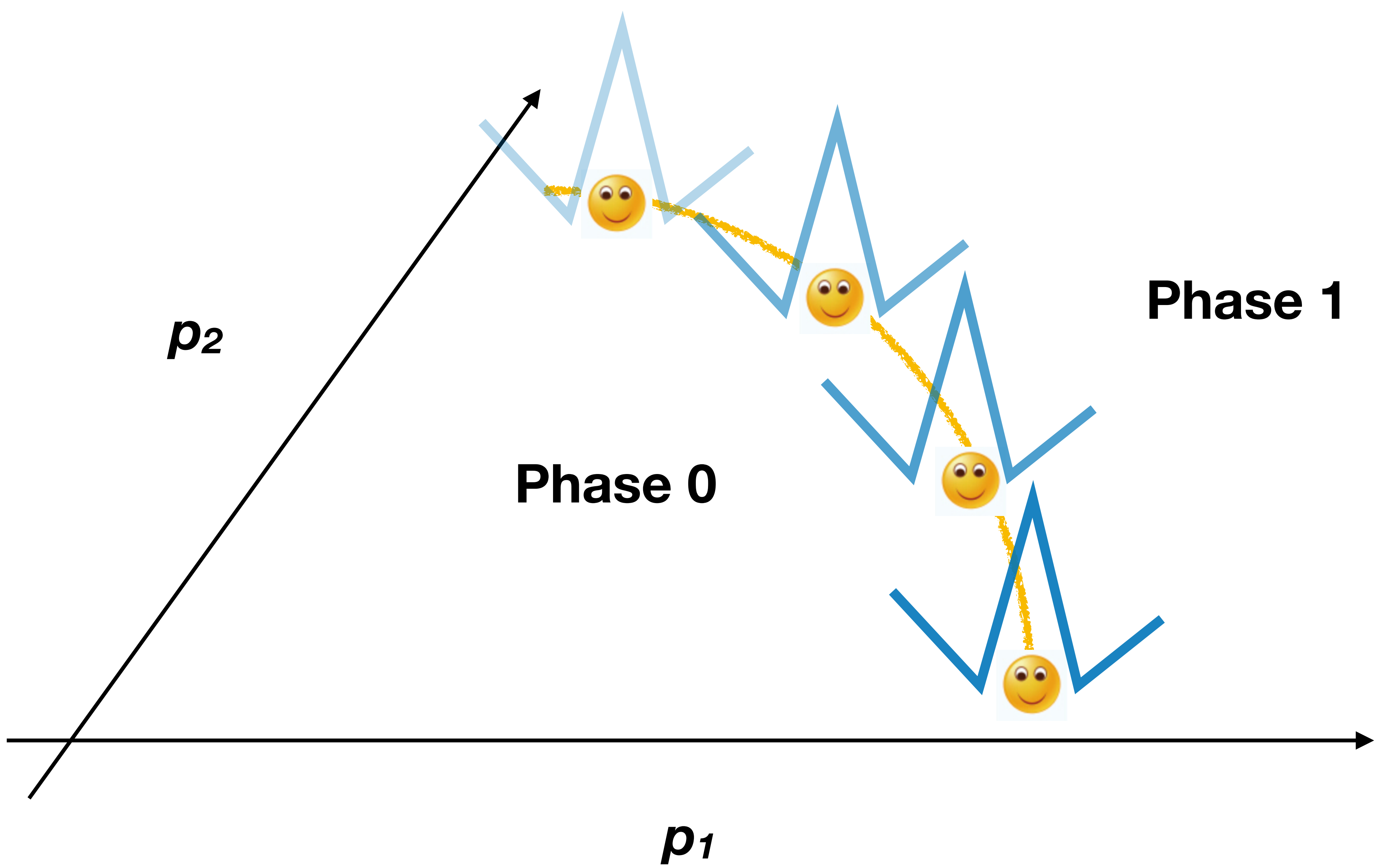


- ***Confusion descent:***

  Eliminated the need for the "guess-sweep".
- ***Higher efficiency:***

  Could apply to higher-D parameter spaces.

2D Parameter Space



- 2D is locally 1D.
- How to parametrize the guesser?

Self-Learning Snake

2D Image-Feature Extraction with *Snake*

© Can Stock Photo

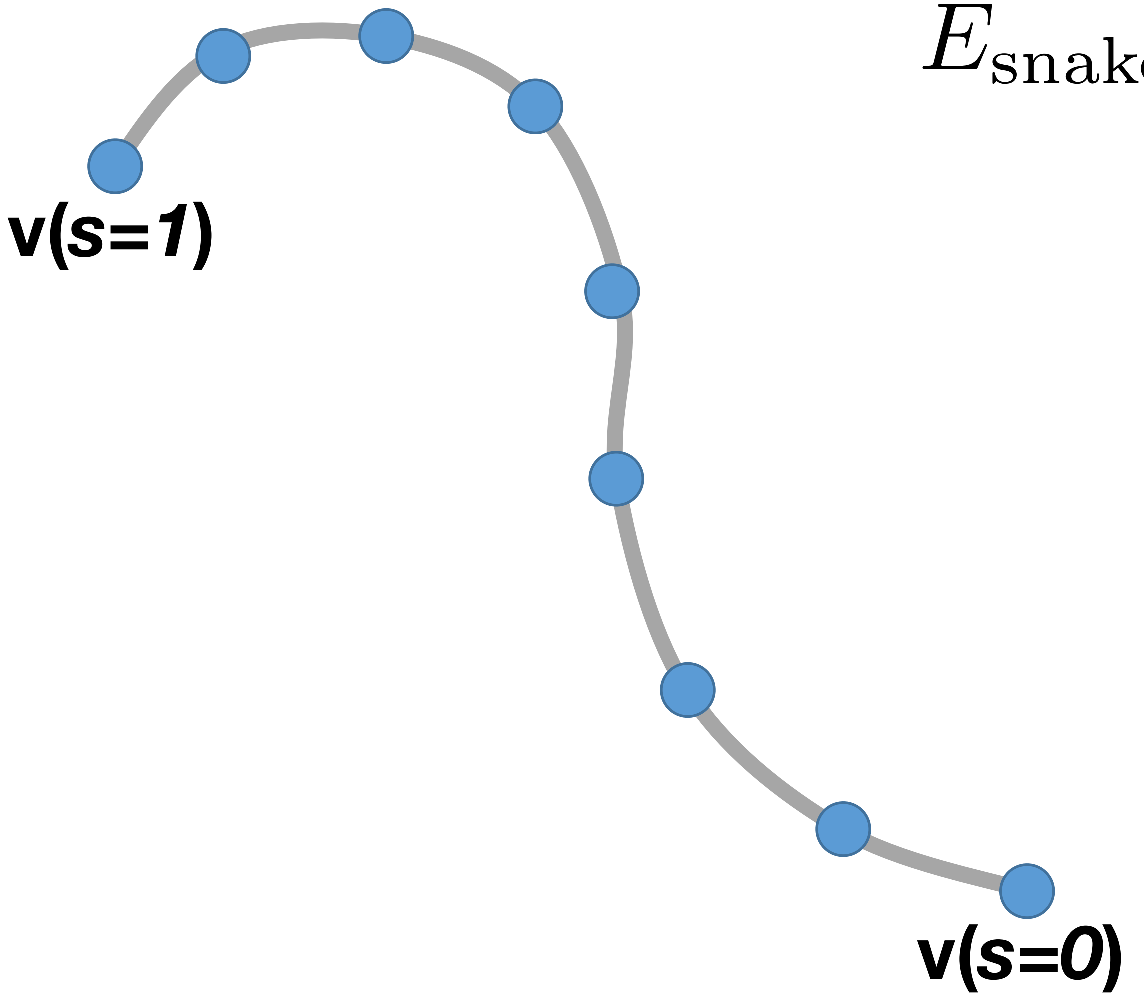
Snakes: Active Contour Models

MICHAEL KASS, ANDREW WITKIN, and DEMETRI TERZOPOULOS
Schlumberger Palo Alto Research, 3340 Hillview Ave., Palo Alto, CA 94304

[PDF] Snakes: Active Contour Models - Cs.UCLA.Edu
www.cs.ucla.edu/~dt/papers/ijcv88/ijcv88.pdf ▾ Diese Seite übersetzen
von M KASS - 1988 - Zitiert von: 18416 - Ähnliche Artikel
MICHAEL **KASS**, ANDREW WITKIN, and DEMETRI TERZOPOULOS ... Abstract. A **snake** is an energy-minimizing spline guided by external constraint forces and ...

Snake: a very influential model in computer vision.

Snake Dynamics in an Image



$$E_{\text{snake}} = \int_0^1 ds\, E_{\text{int}}(\vec{v}(s)) + E_{\text{image}}(\vec{v}(s))$$

$$\frac{d}{dt}\vec{v}(s) = -\eta \frac{\delta E_{\text{snake}}}{\delta \vec{v}(s)}$$

$$E_{\text{int}} = \alpha \left| \frac{d\vec{v}}{ds} \right|^2 + \beta \left| \frac{d^2\vec{v}}{ds^2} \right|^2$$

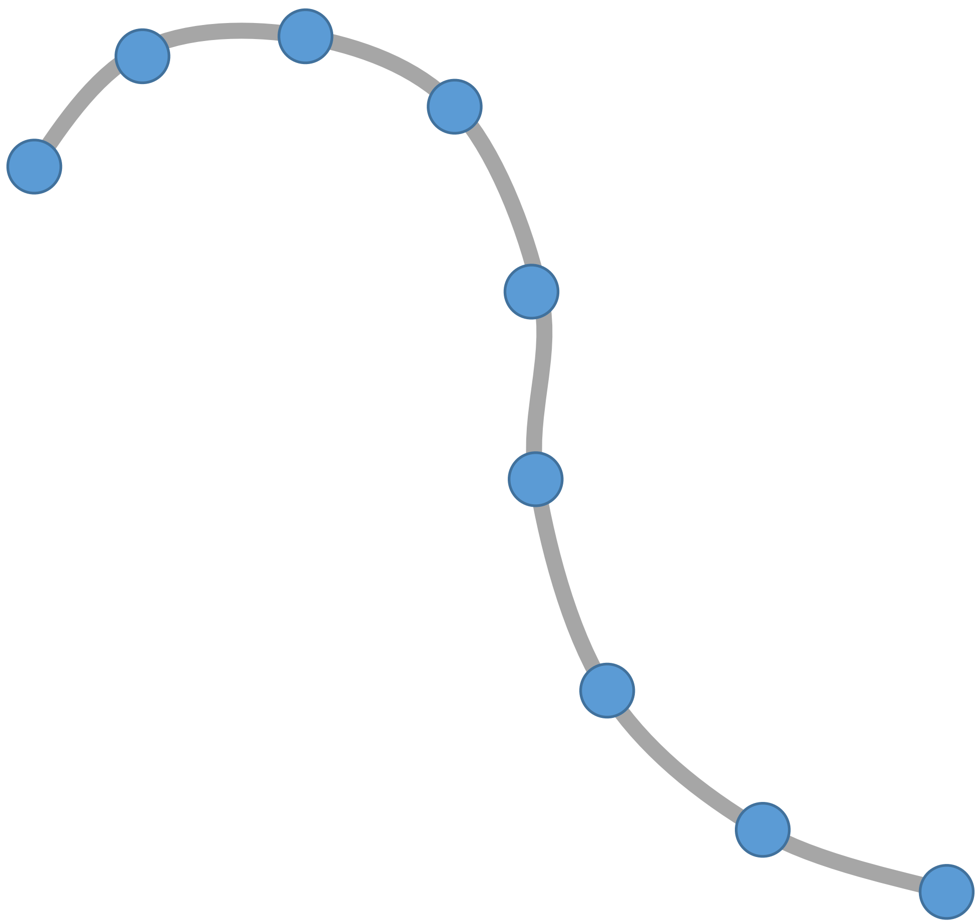
elastic **solid**

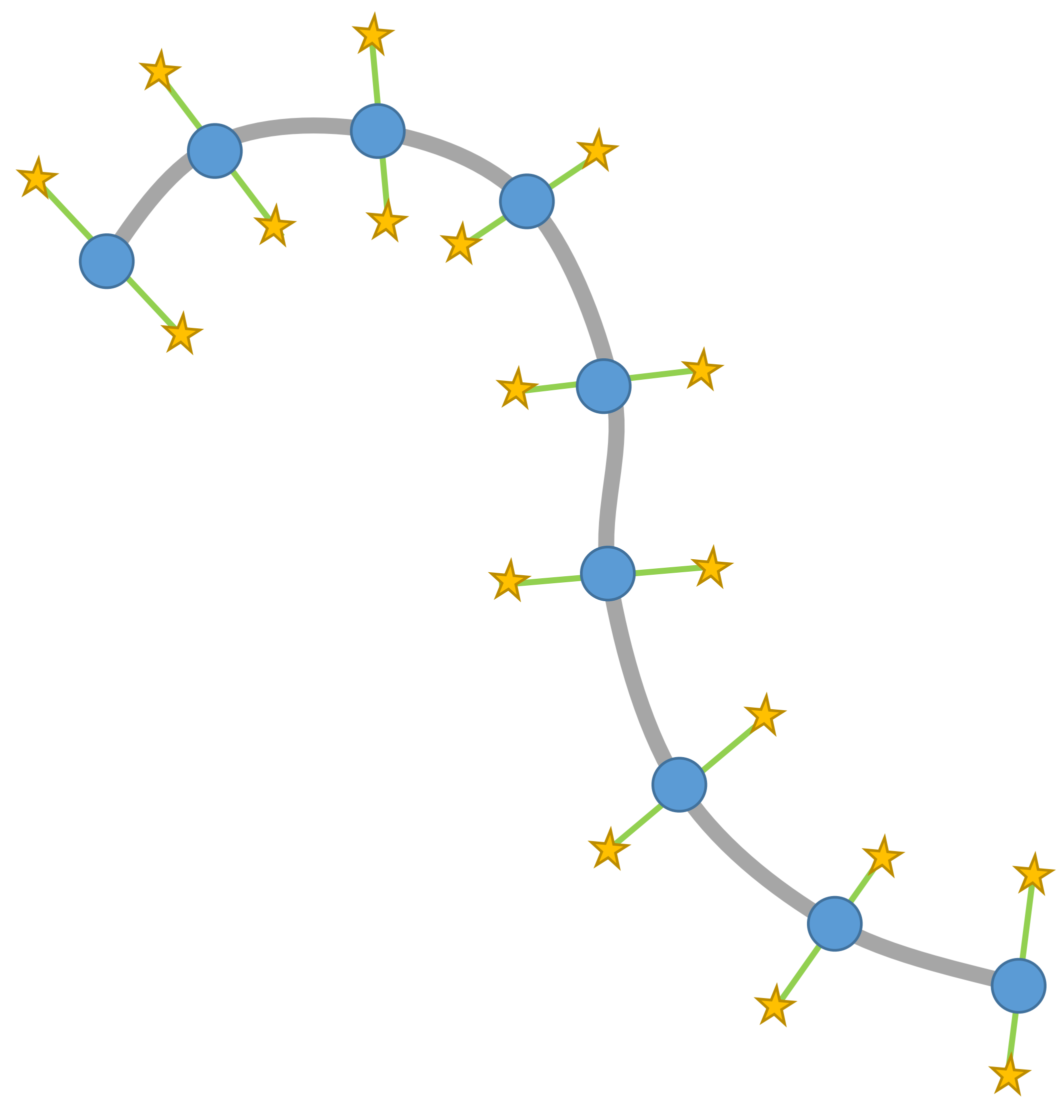
anti-stretching **anti-bending**

Line Detector: $$E_{\text{image}} = I(\vec{v}(s))$$

Edge Detector: $$E_{\text{image}} = -\left|\nabla I(\vec{v}(s))\right|^2$$

Intelligent Snake Self-Generates Training Samples





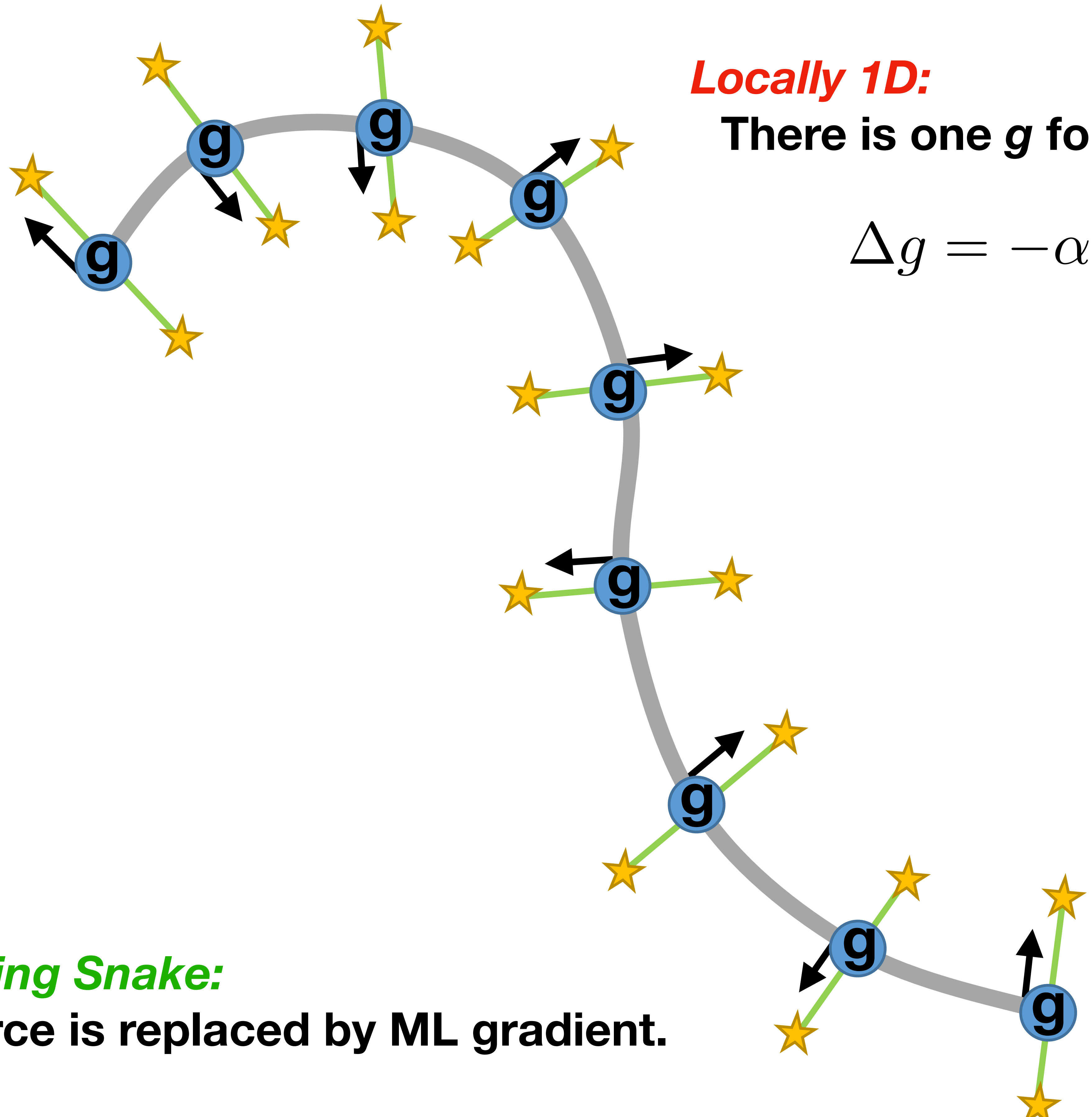
The snake senses its surroundings.

Snake! Are you OK?
Are you confused?

Locally 1D:

There is one *g* for each node.

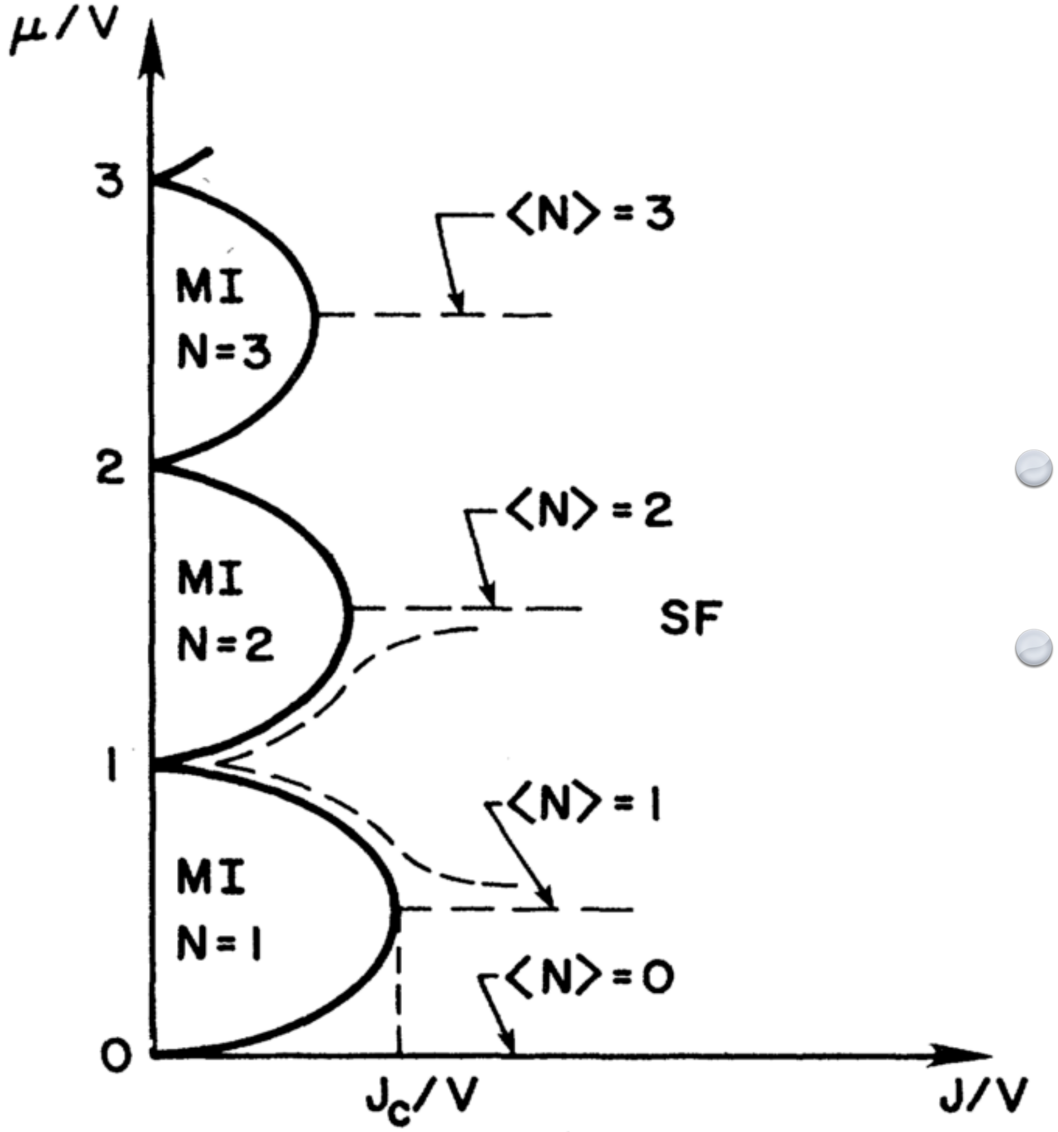
$$\Delta g = -\alpha_g \frac{\partial C}{\partial g}$$



Self-Learning Snake:

Image force is replaced by ML gradient.

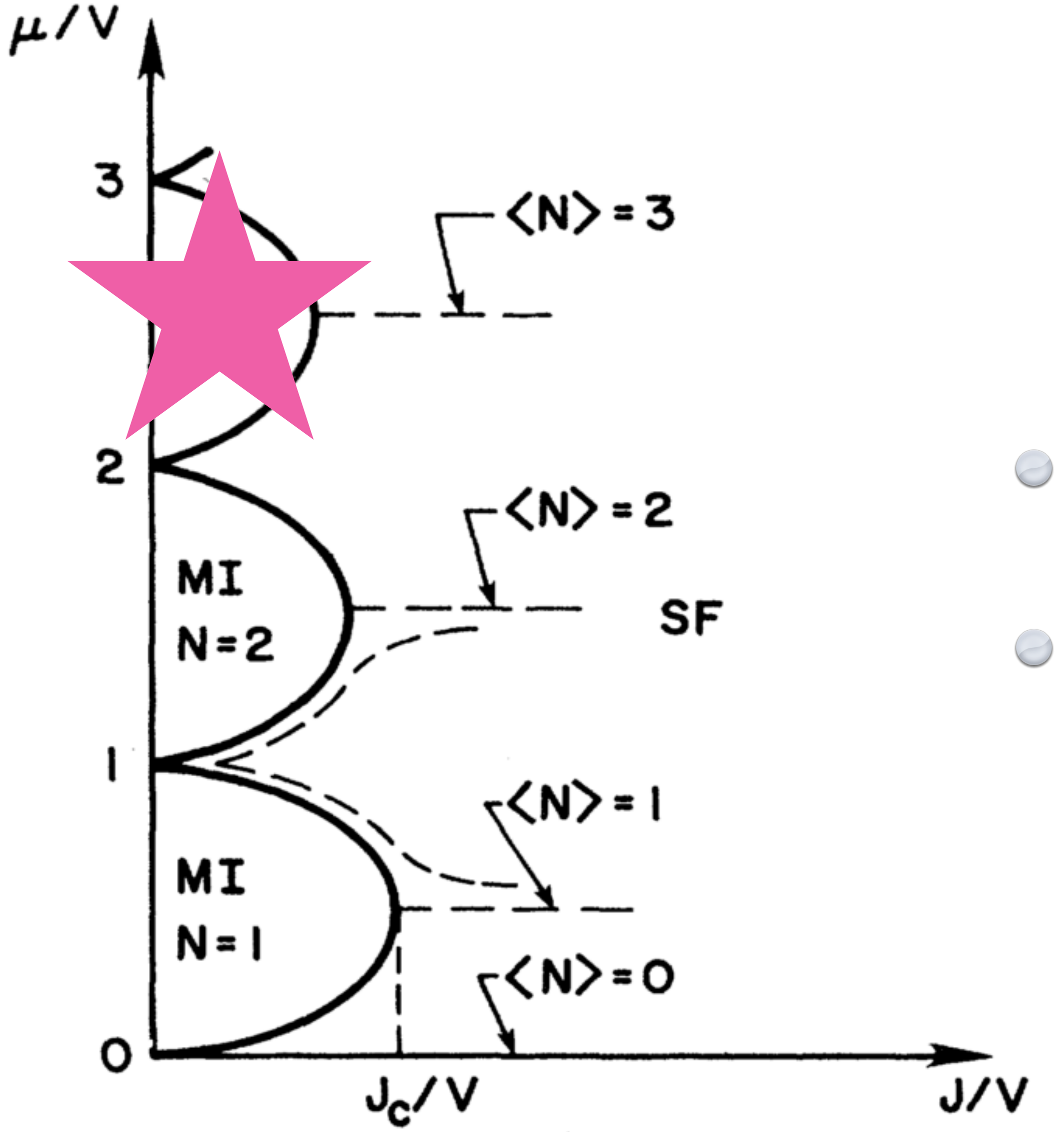
Bose Hubbard Model — Physics



- **Small hopping leads to localization.**
- **Integer filling leads to localization.**

$$H = -J \sum_{\langle i,j \rangle} (b_i^\dagger b_j + b_j^\dagger b_i) + \sum_i \left[\frac{U n_i (n_i - 1)}{2} - \mu n_i \right]$$

Bose Hubbard Model — Physics



- **Small hopping leads to localization.**
- **Integer filling leads to localization.**

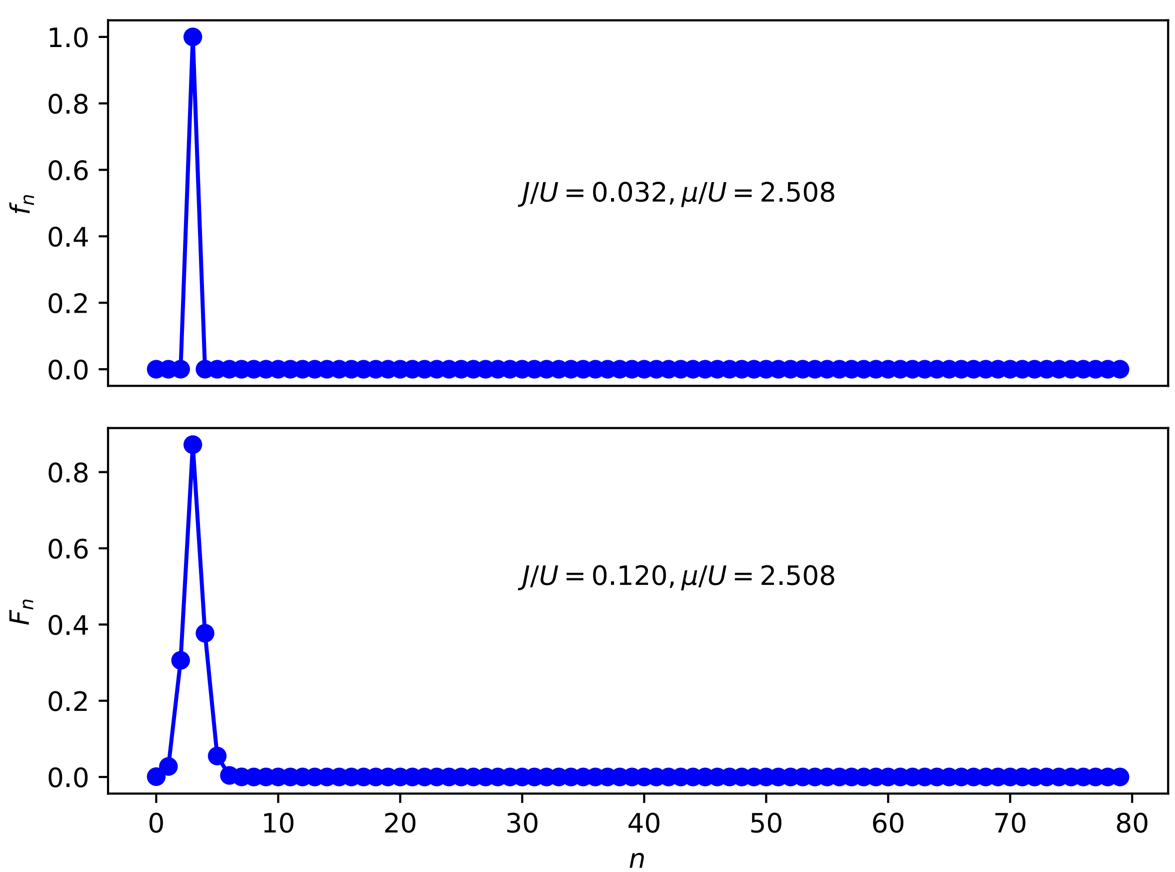
$$H = -J \sum_{\langle i,j \rangle} (b_i^\dagger b_j + b_j^\dagger b_i) + \sum_i \left[\frac{U n_i (n_i - 1)}{2} - \mu n_i \right]$$

Bose Hubbard Model — Training Data

Mean-field theory:

$$E = -\frac{dt}{\rho}\left[\sum_{n=0}^{\infty}\sqrt{n+1}\,f(n)f(n+1)\right]^2$$

$$+\frac{U}{2\rho}\sum_{n=0}^{\infty}n(n-1)f^2(n)\ .$$



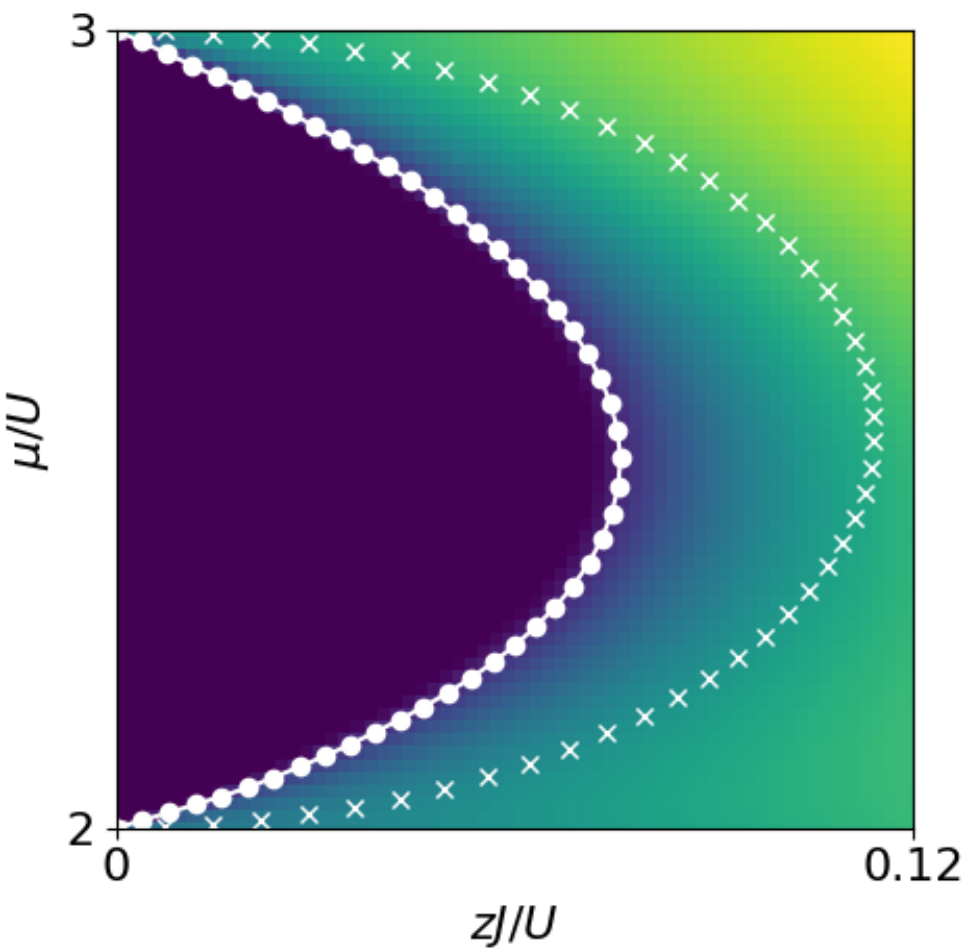
- **Insulating state:**

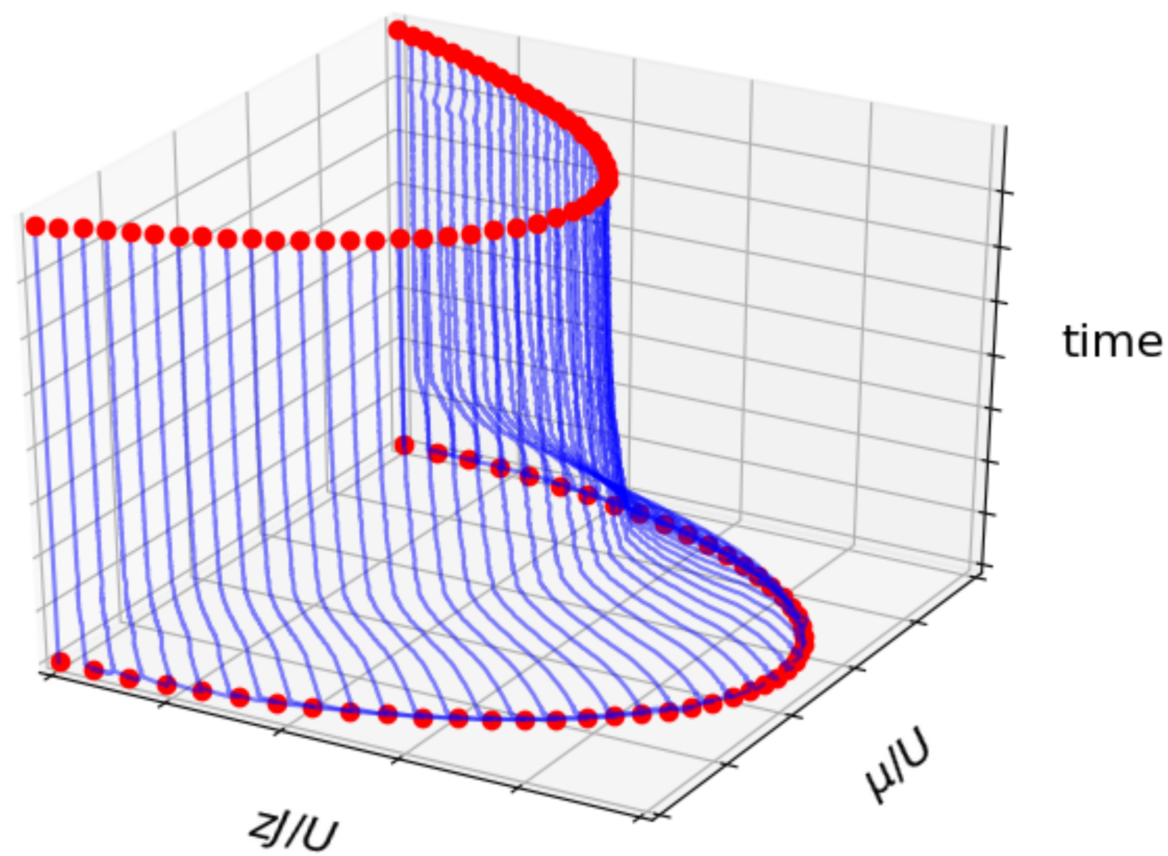
  An eigenstate of N.

- **Superfluid state:**

  There is phase coherence.

Bose Hubbard Model — Self-Learning

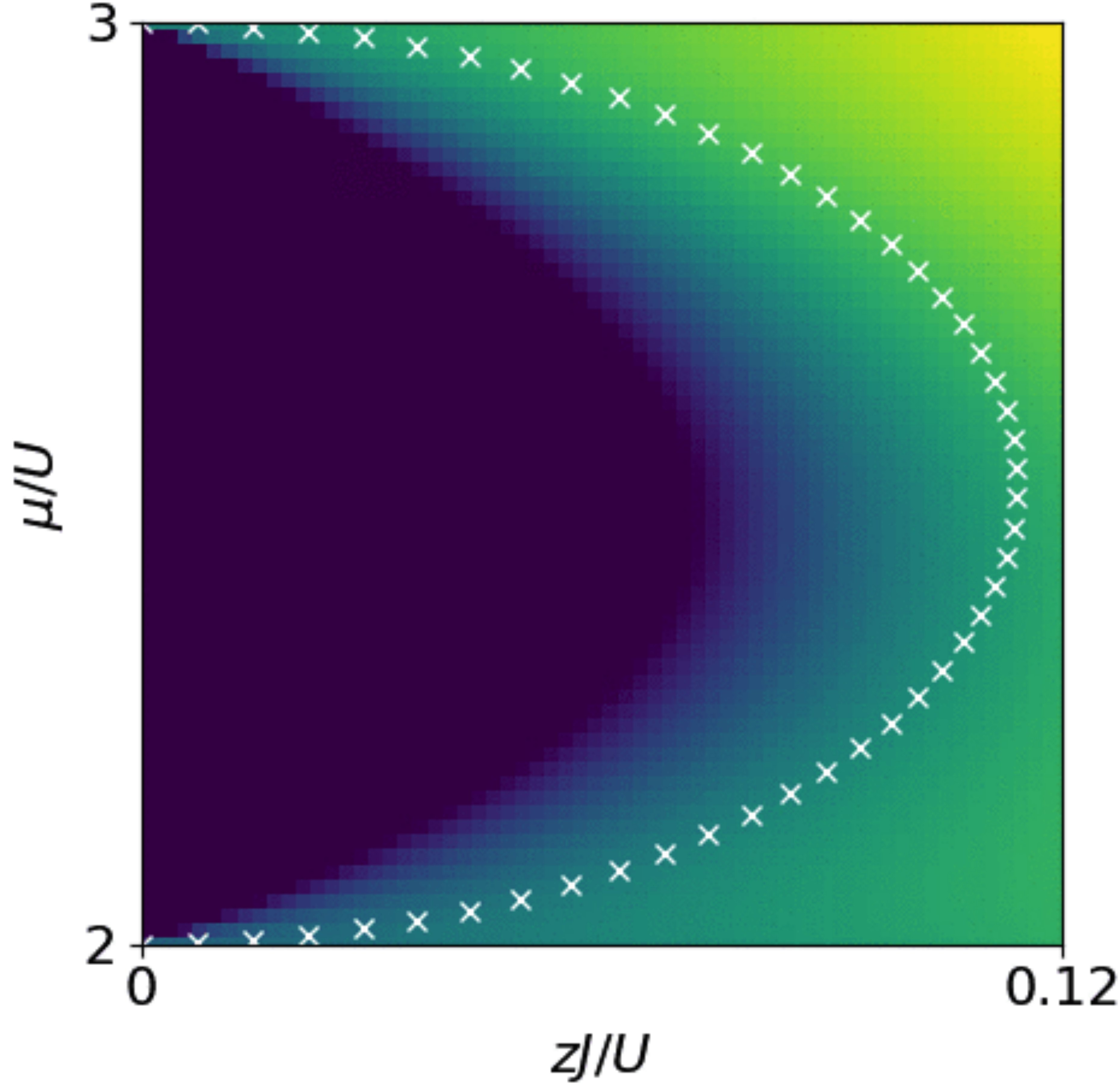




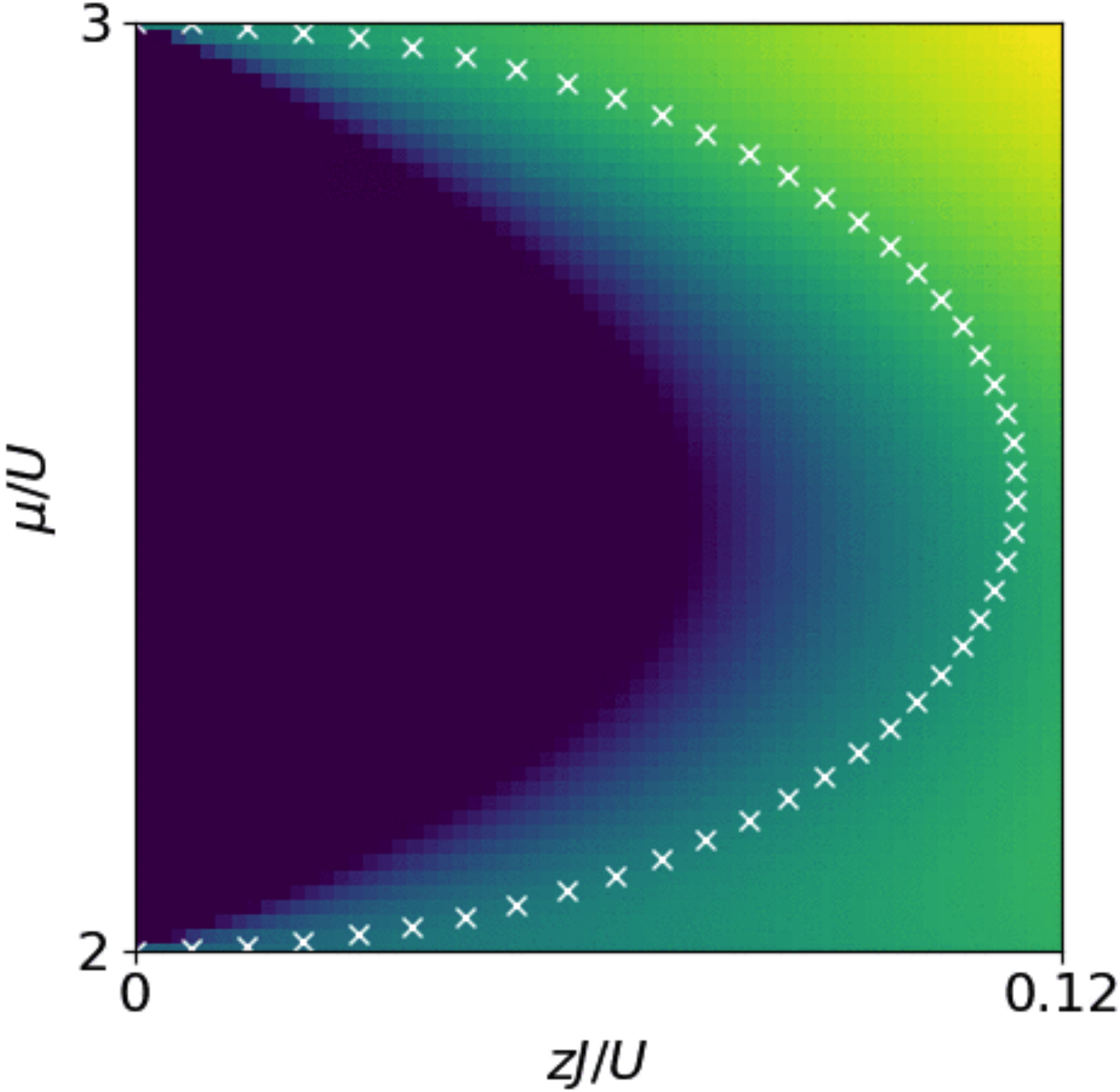
Initial large snake shrinks to the correct phase boundary.

Background plot is the average hopping.

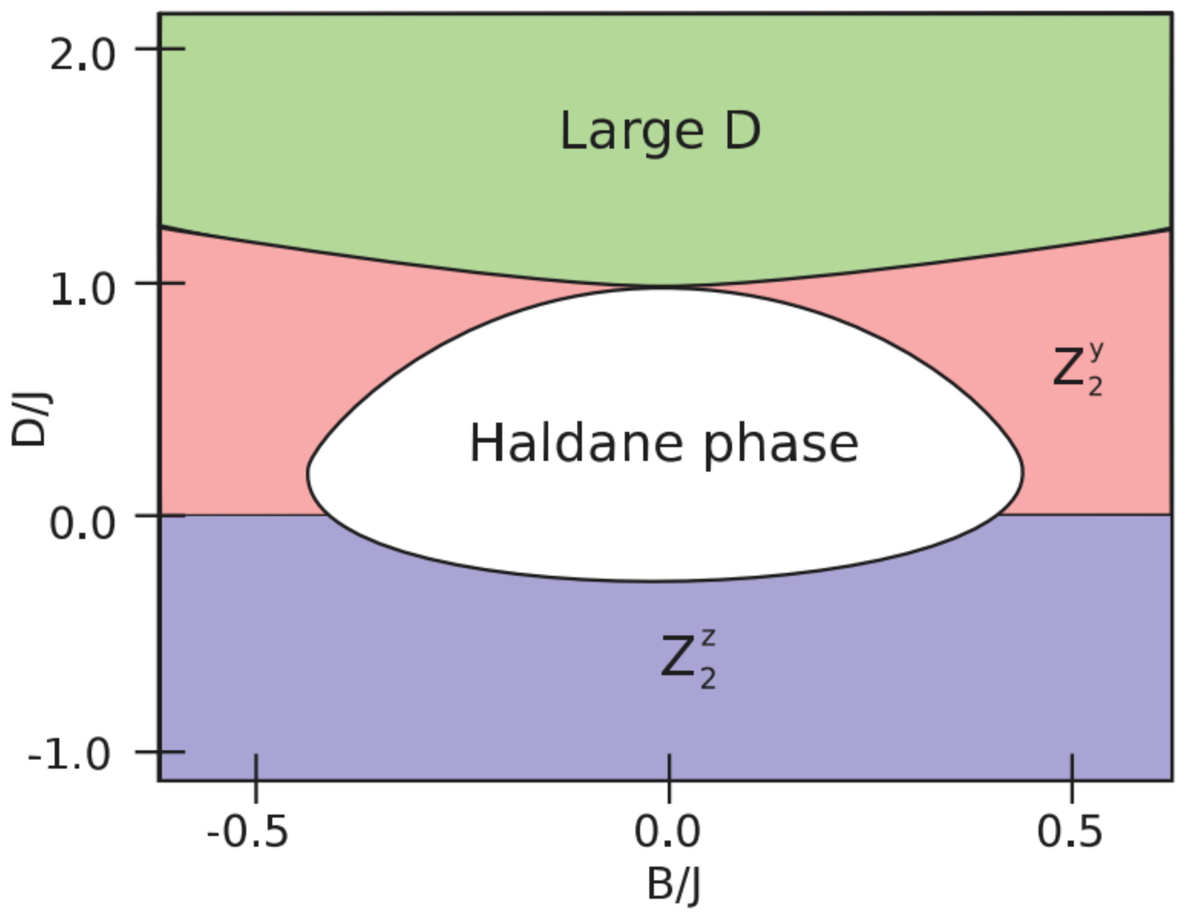
But the snake is fed with f(n).



3
2
μ/U
0
0.12
zJ/U


3
2
μ/U
0
0.12
zJ/U

Spin-1 Heisenberg Chain — Physics

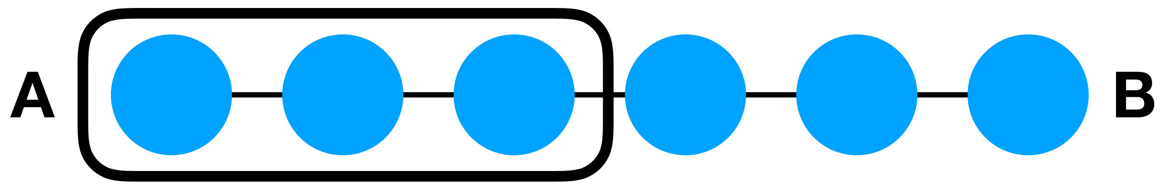


- ***B=0* & *D=0*: Haldane phase.**
- **Haldane phase:**
  A topologically nontrivial.

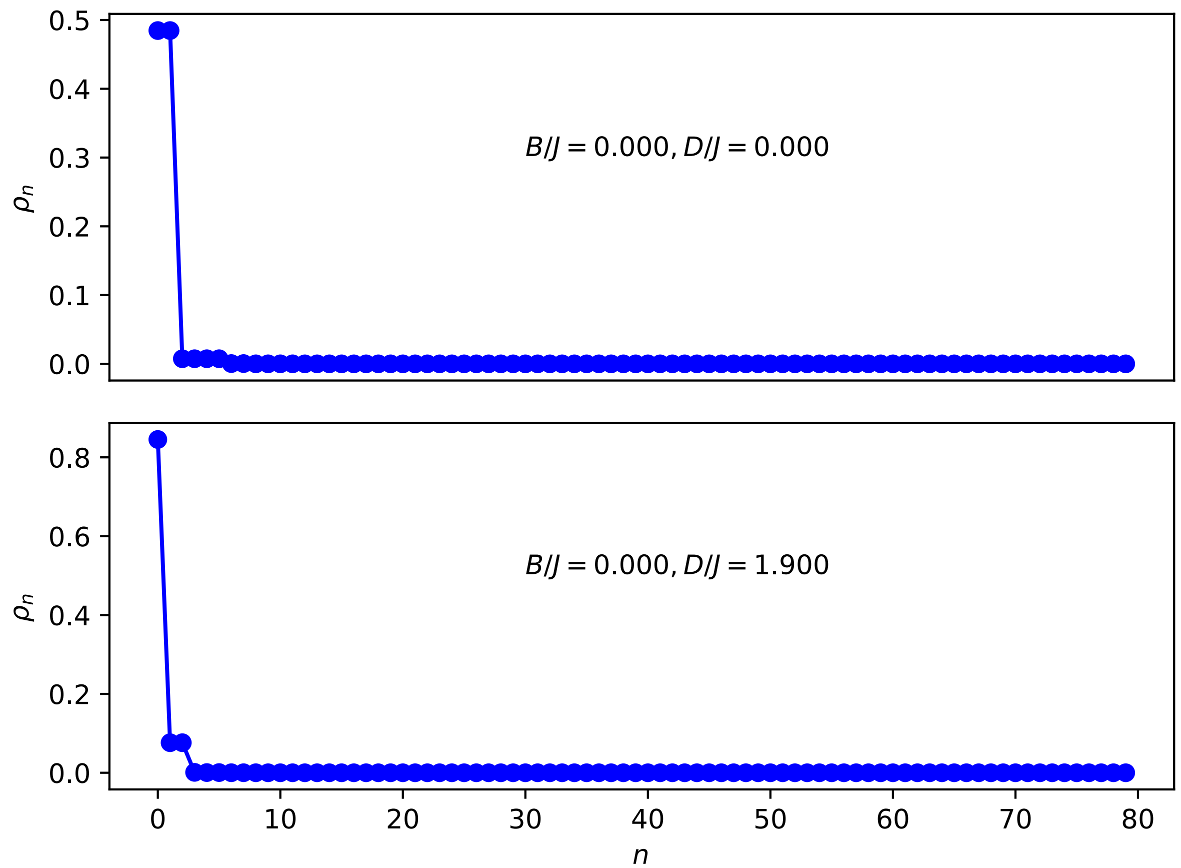
$$H = J\sum_i \mathbf{S}_i \cdot \mathbf{S}_{i+1} + \sum_i \left[D(S_i^z)^2 - BS_i^x\right]$$

Spin-1 Heisenberg chain — Training Data

iTEBD: Entanglement Spectrum



$$\mathrm{ES} = \mathrm{eig}(\mathrm{tr}_{\mathrm{B}}\rho_{\mathrm{AB}})$$



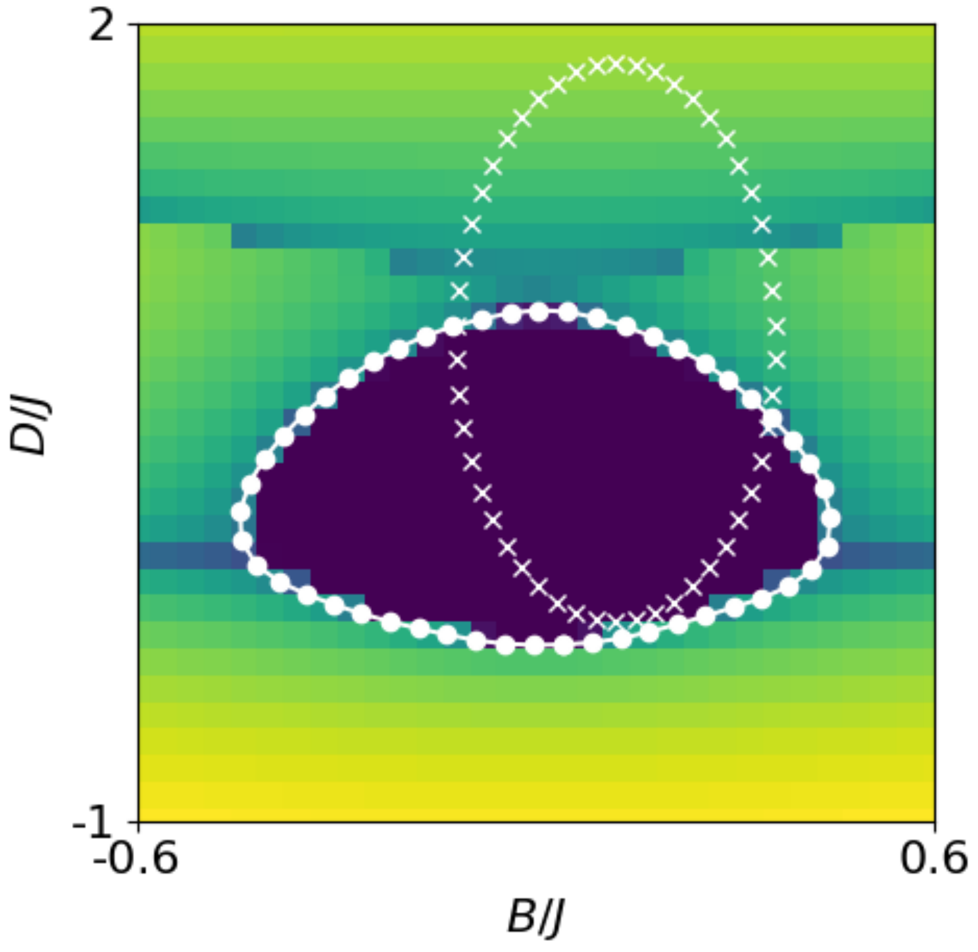
- **Nontrivial phase:**

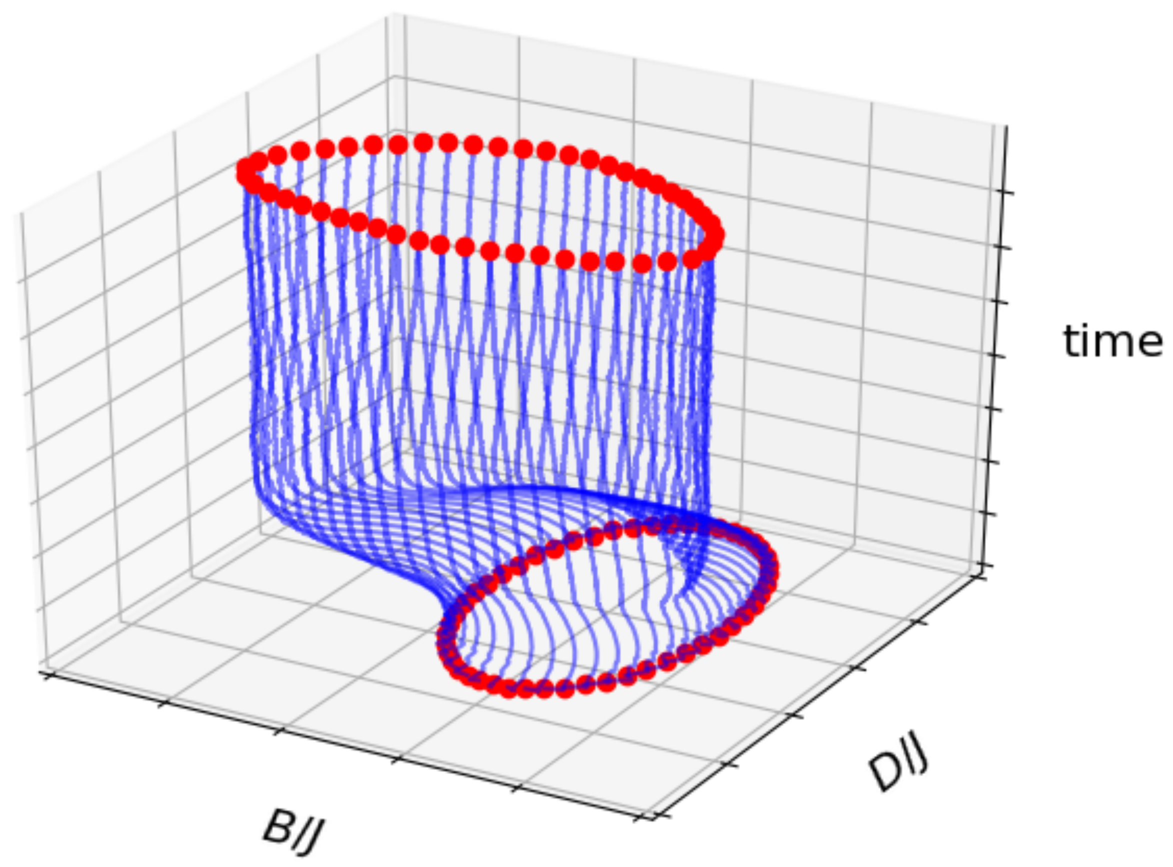
  Degeneracy of largest eigenvalue.

- **Trivial phase:**

  Isolated largest eigenvalue.

Spin-1 Heisenberg chain — Self-Learning

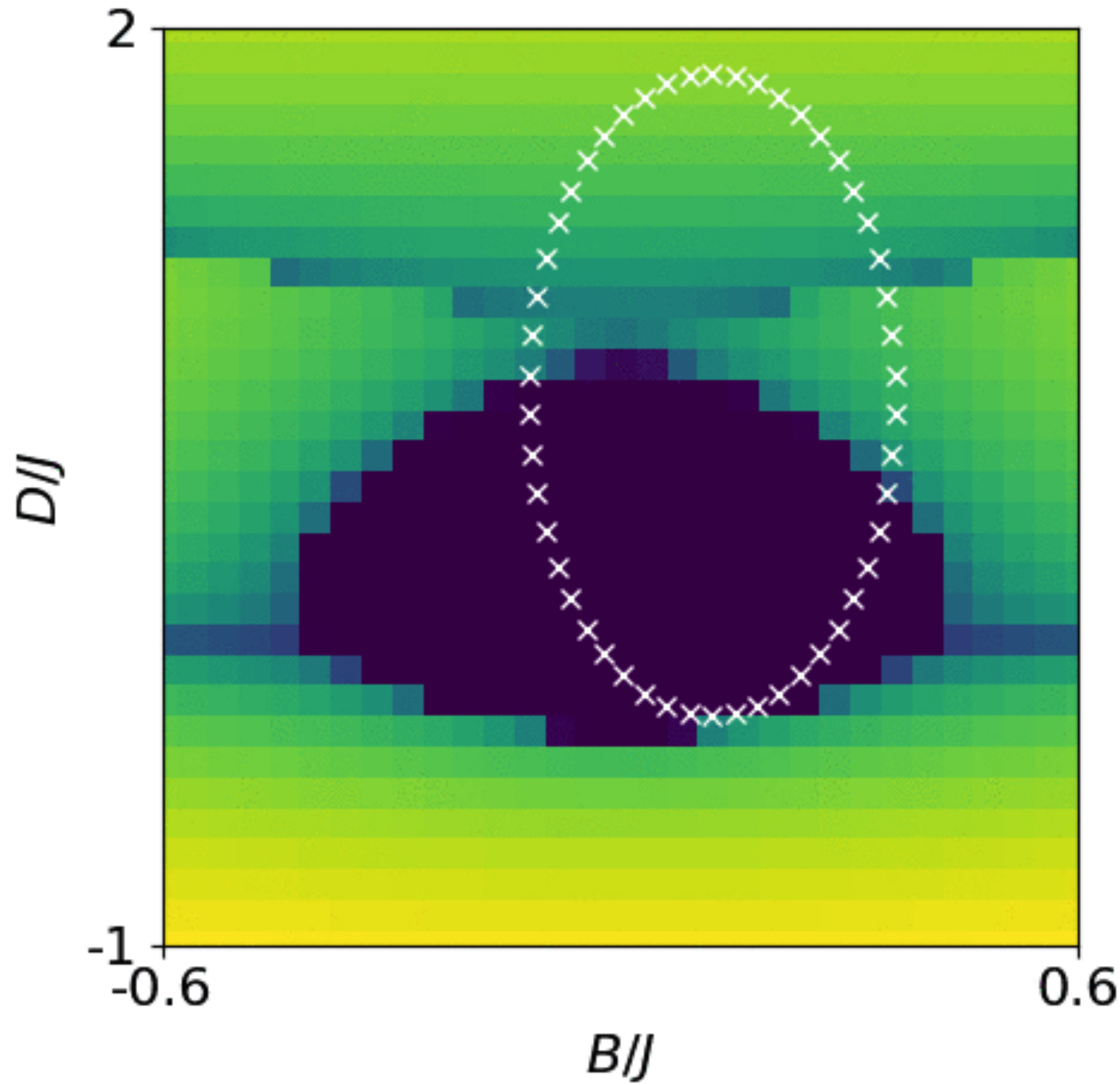


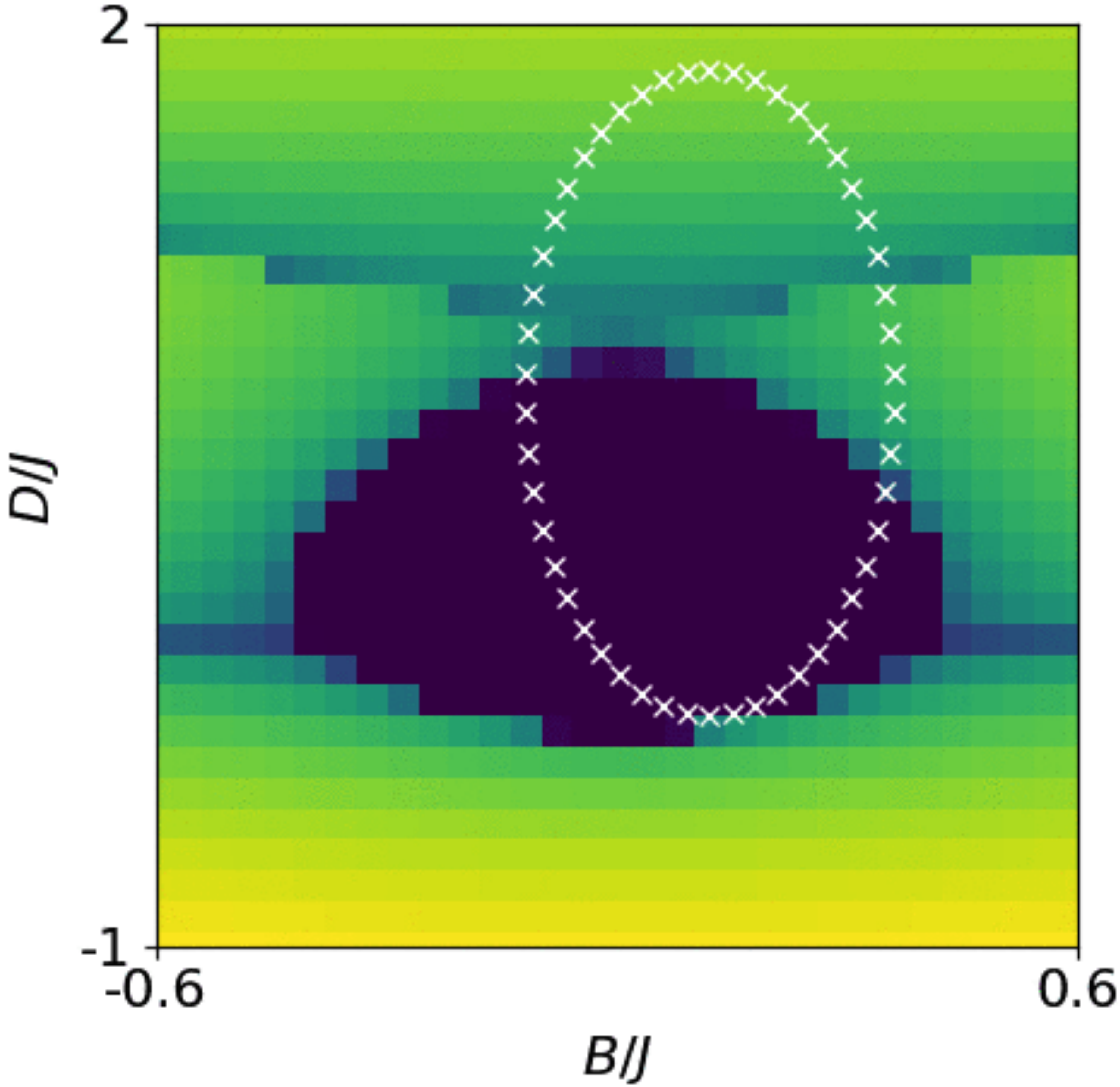


Initial large snake moves, rotates, and deforms to the Haldane pocket.

Background plot is ES gap.

But the snake is fed with the full ES.


2
-1
D/J
-0.6
0.6
B/J



2
-1
D/J
-0.6
0.6
B/J

Summary

- **A hybrid scheme for learning phase transitions: confusion scheme**
- **A guesser agent cooperates with the learner: self-learning scheme**
- **2D parameter spaces: self-learning snake**

E. P. L. van Nieuwenburg, YHL, and S. D. Huber, Nat. Phys. **13**, 435 (2017)

YHL and E. P. L. van Nieuwenburg, arXiv:1706.08111

Summary

- **A hybrid scheme for learning phase transitions: confusion scheme**
- **A guesser agent cooperates with the learner: self-learning scheme**
- **2D parameter spaces: self-learning snake**

E. P. L. van Nieuwenburg, YHL, and S. D. Huber, Nat. Phys. **13**, 435 (2017)

YHL and E. P. L. van Nieuwenburg, arXiv:1706.08111

Thank you for your attention!