Learning Renormalization Group Flows

Jay Shen[1] Prof. Ying-Jer Kao[2]

[1]University of Chicago

[2]National Taiwan University

UCTS Final Presentation. August 16, 2024

Table of Contents

1. What is Renormalization?
2. The Renormalization Group Flow
3. Why use Machine Learning?
4. The Ising Model Problem
5. A Machine Learning Algorithm for Ising Model Renormalization
6. Results

1 What is Renormalization?

2 The Renormalization Group Flow

3 Why use Machine Learning?

4 The Ising Model Problem

5 A Machine Learning Algorithm for Ising Model Renormalization

6 Results

At a high level...

Renormalization is a theory of theories.

Guiding Question

How does our model for a system change as we consider it from different **length scales**?

A Toy Example

Say we want to simulate a cubic millimeter of water. What does this system look like?

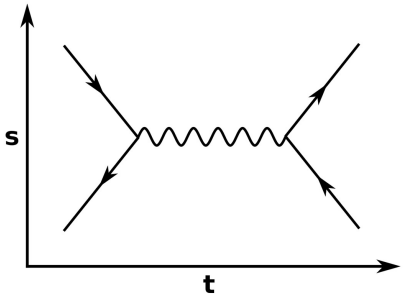
- 10^{19} water molecules
- 3 atoms in each molecule
- A bunch of fundamental particles in each atom

What are the models we have to understand such a system?

A Toy Example: Smallest Scale Model

At the particle scale, field theories:

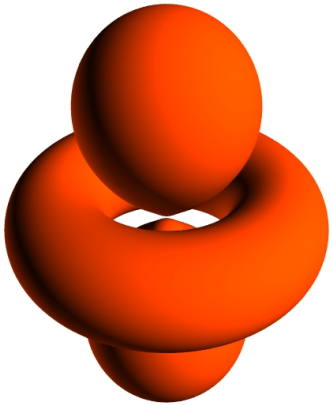
- Incredibly complex model with many variables and parameters to keep track of
- Practically intractable integrals to compute
- Completely infeasible to simulate even a drop of water



A Toy Example: Smaller Scale Model

At the subatomic scale, quantum dynamics:

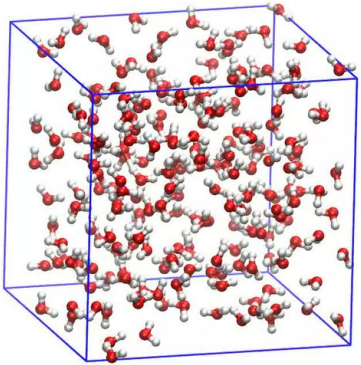
- Complex model with fewer, but still many variables and parameters
- Difficult integrals to compute
- Simulation is possible, but still intractable in general



A Toy Example: Small Scale Model

At the atomic scale, classical dynamics:

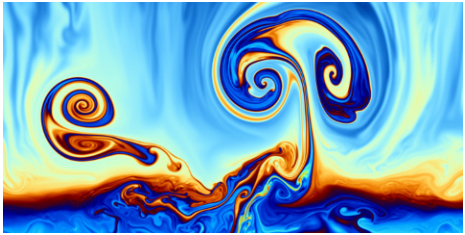
- Intuitive model with only a couple of degrees of freedom and parameters per molecule
- Integrals are easy
- Simulation is expensive, but doable for up to decently large systems



A Toy Example: Large Scale Model

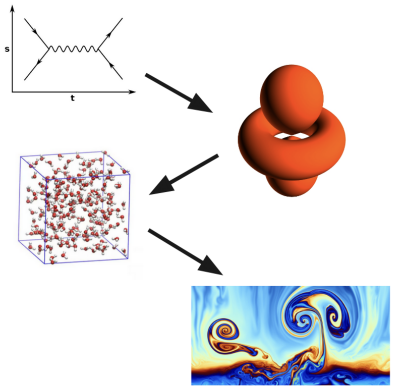
At the human scale, fluid dynamics:

- Modeled by a single PDE with a few constants (viscosity, temperature, pressure, ...) that fully specify the system
- Integration done efficiently using numerical solvers
- Simulation practical and accurate up to very large volumes



Renormalization is a Theory of Scaling

Renormalization is a theory of how these models, all at different scales, can be linked in a highly systematic way.



A Bit of Mathematical Context

What does renormalization look like mathematically?

Consider a system with some degrees of freedom $x_0, x_1, \ldots$ [1] modeled by, say, a Hamiltonian:

$$H = f(x_0, x_1, \ldots, c_0, c_1, \ldots)$$

Here, f is some specific function and $c_0, c_1, \ldots$ [2] are some constant parameters also incorporated into f.

[1] these may be positions, momenta, etc.

[2] these may be masses, fundamental constants, etc.

A Bit of Mathematical Context

When we "zoom out" to a larger length scale, we typically apply some approximation, resulting in new degrees of freedom $x_0', x_1', \ldots$.

Now, in certain cases, we can make the **strong** assumption that this "zoomed out" system can be modeled by the same Hamiltonian form, just with different parameters $c_0', c_1', \ldots$:

$$H = f(x_0', x_1', \ldots, c_0', c_1', \ldots)$$

This adjusting of the model for the new "zoomed out" system is what is called **renormalization**.

1 What is Renormalization?

2 The Renormalization Group Flow

3 Why use Machine Learning?

4 The Ising Model Problem

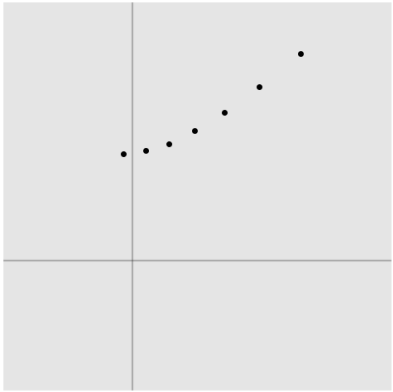
5 A Machine Learning Algorithm for Ising Model Renormalization

6 Results

"Zooming Out" As a Flow

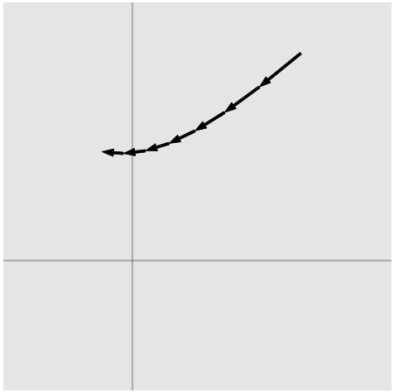
Imagine repeating the renormalization process again and again, each repetition giving new degrees of freedom and new renormalized parameters for our increasingly "zoomed out" model.

The changing parameters will form a path in the high-dimensional space of parameters.



"Zooming Out" As a Flow

We can let these "steps" define vectors that point to the next set of parameters...

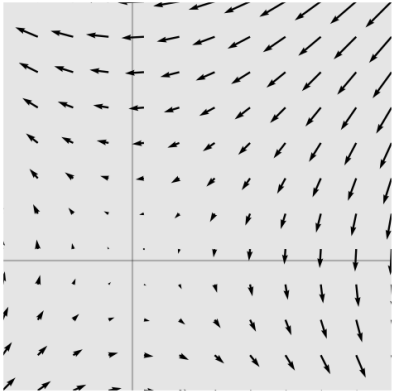


"Zooming Out" As a Flow

Then if we solve for these vectors continuously over the whole space, we get a vector field.

This is the renormalization group (RG) flow of the parameters.

But why is this important?

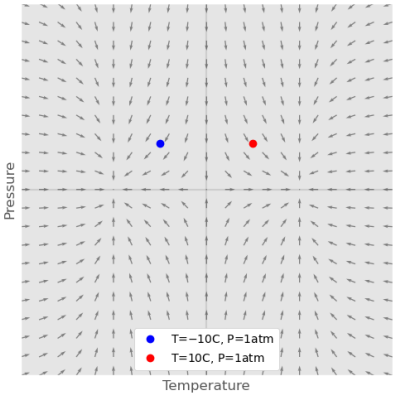


A Toy Example of Phase in RG Flows

Let's go back to our toy example of a box of water.

Say our two initial parameters are temperature and pressure.

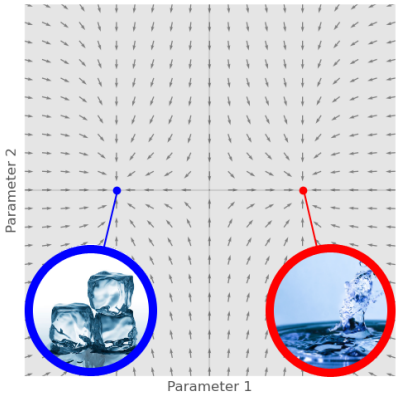
Let us consider two systems, one below freezing and one above. The initial parameters in parameter space look something like:



A Toy Example of Phase in RG Flows

Applying a renormalization procedure and following the RG flow, our parameters end up in two fixed points.

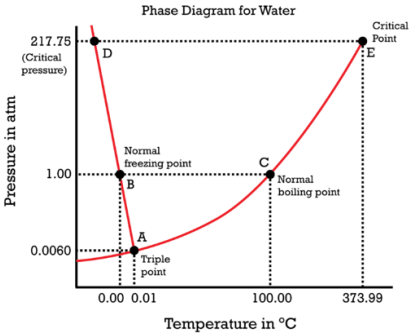
Physically, these will correspond to two phases of matter, ice and water.



RG Flow and Phase Behavior

In general, fixed points and other attracting manifolds correspond to distinct phases of the physical system.

As a result, characterizing the RG flow provides enormous insight into critical behaviors, enabling, for example, the construction of phase diagrams.



1 What is Renormalization?

2 The Renormalization Group Flow

3 Why use Machine Learning?

4 The Ising Model Problem

5 A Machine Learning Algorithm for Ising Model Renormalization

6 Results

Renormalization is **Hard**

Renormalization promises systematic understanding of phase and its relation to scale and universality.

Unfortunately, this comes at a cost.

Traditionally, renormalization requires human physicists to:

- use their own intuition to design an approximation scheme for "zooming out"
- solve the corresponding theories to determine parameter flow
- determine critical behaviors from the formulated flow

How Can Machine Learning Help?

Renormalization and machine learning have a lot in common[3]:

- Both are ways of distilling essential information and correlations from complex systems.
- Both are resistant to statistical noise, whether it be minibatches or microstates.
- Both map high dimension information to lower dimension representations.
- Both use similar mathematical frameworks of generative models.

Ultimately, these similarities can be leverage to automate renormalization and the study of complex physical systems.

[3] Hou and You 2023.

1 What is Renormalization?

2 The Renormalization Group Flow

3 Why use Machine Learning?

4 The Ising Model Problem

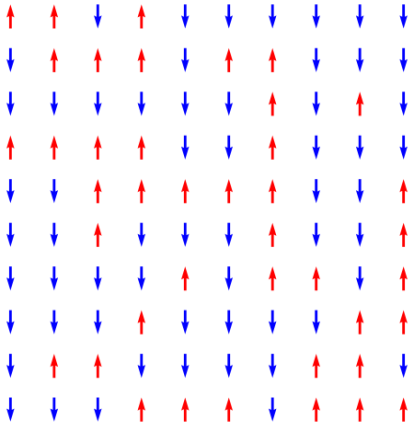
5 A Machine Learning Algorithm for Ising Model Renormalization

6 Results

The Ising Model

The Ising Model is a historically important problem in statistical physics.

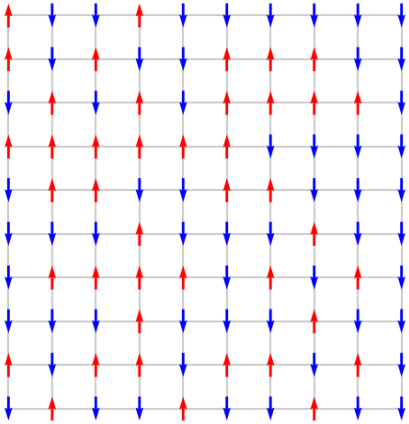
- Originally developed as a model of ferromagnetism, thought it has since been applied to a wide range of problems.
- One of the first probabilistic graphical models, the precursors to modern neural networks.
- Analytical solutions exists, providing good behavioral benchmarks.



The Ising Model, Conceptually

Conceptually, the Ising model is fairly simple:

- Atoms are arranged on some lattice
- Atomic spins are either up or down, and are correlated with the spins of neighboring atoms



The Ising Model Hamiltonian

This principle of localized correlations can be modeled by a Hamiltonian, or total energy function.

Let $\sigma_0, \sigma_1, \cdots \in \{-1, 1\}$ be the values of the atomic spins.
Then the Hamiltonian[4] is:

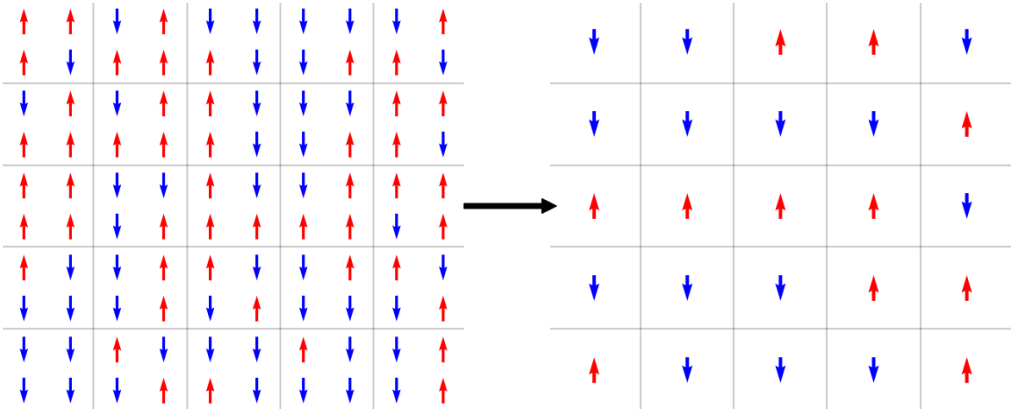
$$H = -\sum_{i \leftrightarrow j} J\sigma_i\sigma_j$$

$i \leftrightarrow j$ signifies that the ith and jth spins are adjacent on the lattice.
J is the correlation strength.

[4] here we assume net zero magnetic field and constant correlation strength

Renormalizing the Ising Model

Say we want to "zoom out" by applying a blocking approximation, mapping a square block of spins to a single spin.



Renormalizing the Ising Model

This approximate lattice is also an Ising model, and can be modeled using the same Hamiltonian form:

$$H' = -\sum_{i' \leftrightarrow j'} J' \sigma_{i'} \sigma_{j'}$$

Here, J' is the adjusted, or renormalized, correlation strength.

Note

We are able to do this because the Ising model is a **scale invariant** theory. Scale invariance is one of the prerequisites for renormalizability.

1 What is Renormalization?

2 The Renormalization Group Flow

3 Why use Machine Learning?

4 The Ising Model Problem

5 A Machine Learning Algorithm for Ising Model Renormalization

6 Results

The MLRG Algorithm

In Hou et al.[5], the authors propose an learning algorithm that promises:

- Fast, automated description of RG flow on Ising models
- Accessible analysis of phase behavior via numerical methods
- Applicability to a wide variety of systems

Let's break the algorithm down.

[5] Hou and You 2023.

Learning Objective

Our objective is to learn a representation of the RG flow.

Hou et al. propose to learn the C-theorem[6] monotone function. We will directly model the RG flow vector field, which is a function:

$$\vec{f}(J) : \mathbb{R}^n \to \mathbb{R}^n$$

Here, n is the number of parameters in the theory. The Ising model has only one parameter, the correlation strength J, so f is a just scalar function:

$$f(J) : \mathbb{R} \to \mathbb{R}$$

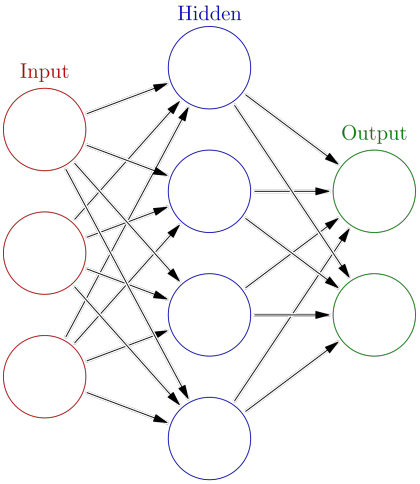
[6] Zamolodchikov 1986.

Learning Model

To model f, we will use a neural network as a function approximator.

Let our neural network be:

$$f_\theta(J) : \mathbb{R} \to \mathbb{R}$$



Deep Learning Procedure

To adjust f_θ so that it bests resembles f, we will follow a typical supervised learning procedure:

1. **Sample** values of J.
2. **Forward Pass** to evaluate $f_\theta(J)$.
3. **Compute Error** between $f_\theta(J)$ and $f(J)$.
4. **Backward Pass** to adjust f_θ according to that error.

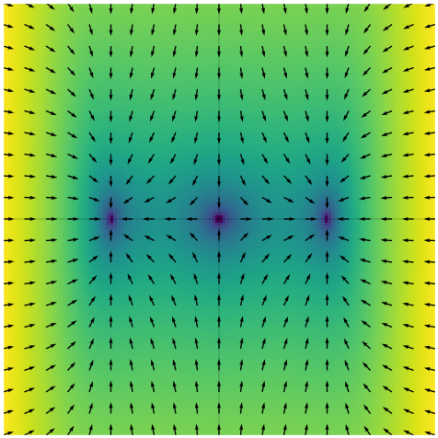
Let go through this procedure step by step.

Step 1: Sampling

In general, the accuracy of neural networks improves wherever sampling is heaviest.

We are interested in fixed/critical points, where the flow goes to zero.

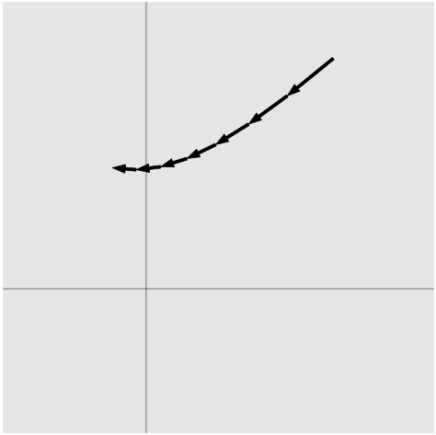
We can sample around them using an annealed Hamiltonian Monte Carlo sampler that minimizes $||f_\theta(J)||$.



Step 2: Forward Pass

Given a sample J for a system, the value $f_\theta(J)$ represents the change in the correlation strength when renormalized.

Accordingly, $J' = J + f_\theta(J)$ is the renormalized correlation strength for the "zoomed out" system.



Compute Error

How can we judge whether f_θ is sending J in the right direction?

Ideally, the behavior of the original and renormalized system should be as similar as possible.

The behavior of an Ising model is defined by the Gibbs distribution of the atomic spins:

$$P(\sigma_0, \sigma_1, \ldots, J) = \frac{1}{Z} e^{-H(\sigma_0, \sigma_1, \ldots, J)}$$

Compute Error

Given values of J and $J' = J + f_\theta(J)$, we have distributions for the original and renormalized systems.

$$P(\sigma_0, \sigma_1, \ldots, J) \qquad\qquad P(\sigma_0', \sigma_1', \ldots, J')$$

The Kullback-Liebler (KL) divergence (D_{KL}) measure the distance between these two distributions. This defines our error:

$$Error = D_{KL}\left(P(\sigma_0, \sigma_1, \ldots, J)\,,\; P(\sigma_0', \sigma_1', \ldots, J') \right)$$

Backward Pass

Now that we have an error value, we can use the backpropagation algorithm to adjust f_θ so that it does better in the future.

$$\Delta\theta = \frac{\partial Error}{\partial\theta}$$

1 What is Renormalization?

2 The Renormalization Group Flow

3 Why use Machine Learning?

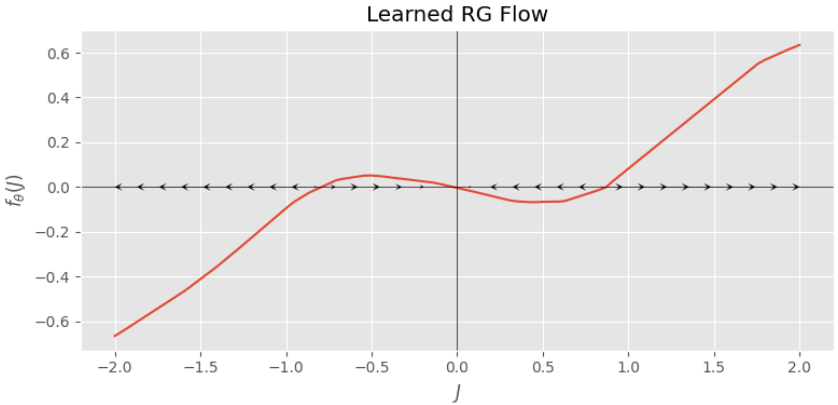
4 The Ising Model Problem

5 A Machine Learning Algorithm for Ising Model Renormalization

6 Results

Learned RG Flow

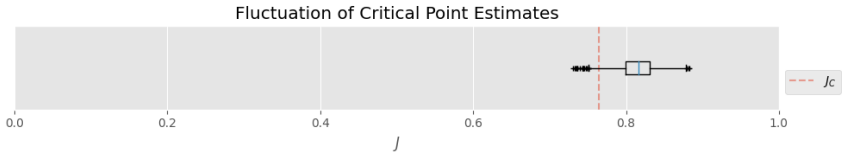
Following this procedure, we train a neural network $f_\theta : \mathbb{R} \to \mathbb{R}$.
Upon convergence, f_θ looks like:



Critical Point Estimates

Given the learned f_θ function, we use Newton's method to numerically solve for fixed and critical points where $||f_\theta|| \to 0$

The values vary, but we can plot their distribution by training many models:



A Problem of Bias

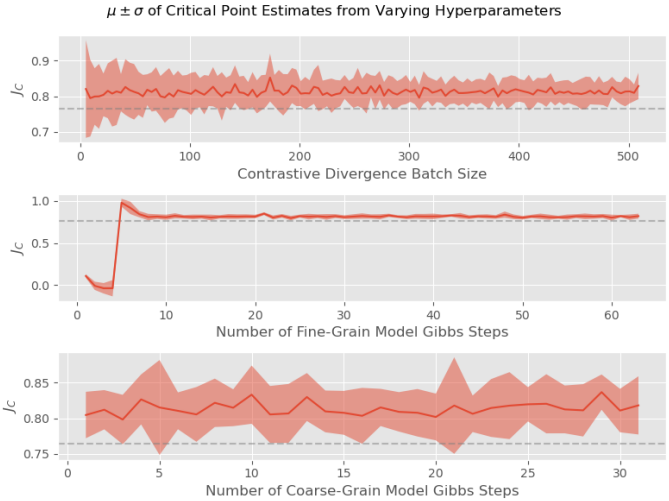
Evidently, our estimates are biased from the analytically derived results. Why?

First, let's consider possible sources of bias within our training procedure.

Reducing Bias from Contrastive Approximation

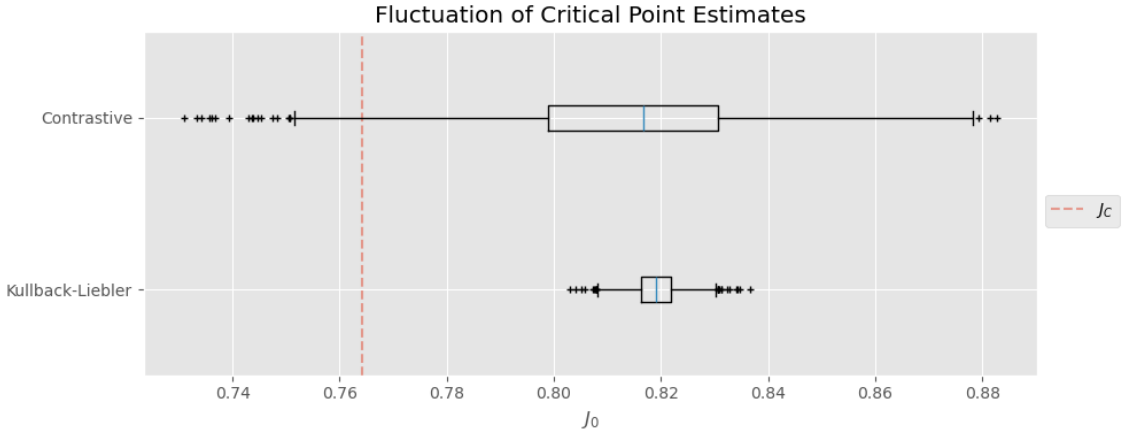
One possible source of bias is the use of contrastive approximations to the KL divergence.

We can refine approximation with more Gibbs sampling, but this only decreases bias up to a certain extent.



Reducing Bias from Contrastive Approximation

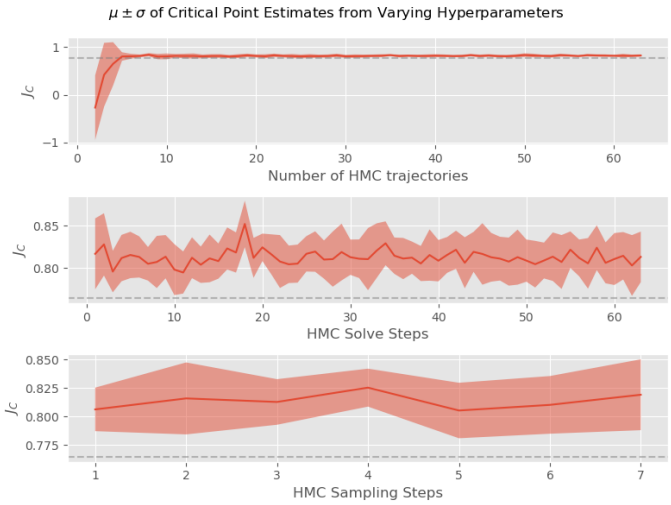
In fact, even computing the actual KL divergence doesn't reduce the bias, only serving to reduce variance:



Reducing Bias from Sampling Inefficiency

The other possible source of random bias is some inefficiency in the sampling procedure.

Experimenting with a range of sampler settings shows only limited effects on the bias.



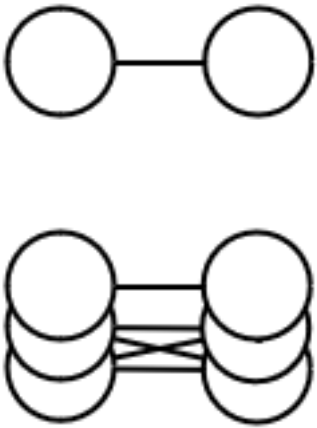
Reducing Bias by Increasing Representation Complexity

Hou et al. conjecture that the bias results from the simplicity of the Ising model, i.e. the blocking approximations are too coarse.

To ameliorate this, they propose stacking additional spins and correlation constants, increasing the complexity and expressiveness of the Ising model.

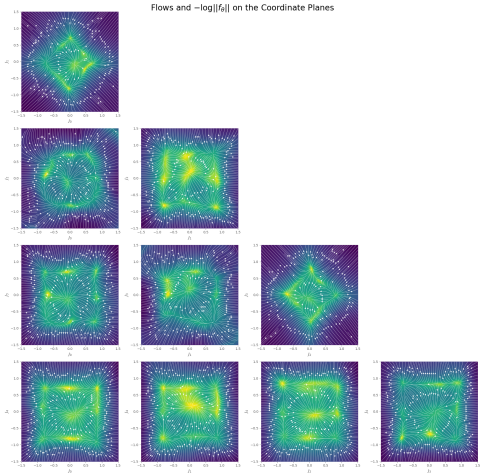
This means we now need to model an n-dimensional flow:

$$\vec{f_\theta} : \mathbb{R}^n \to \mathbb{R}^n$$



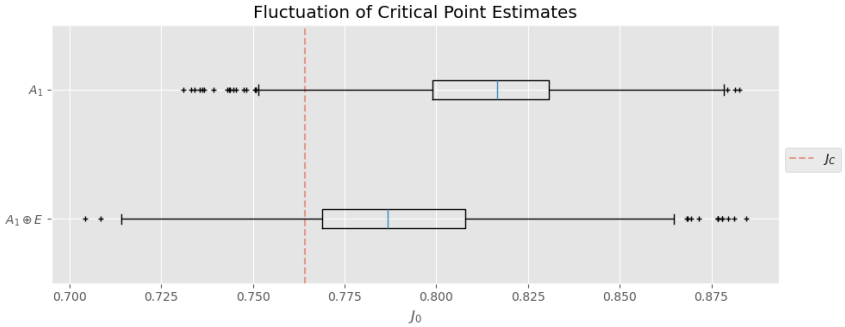
Reducing Bias by Increasing Representation Complexity

Using our algorithm, we model such a flow over 5 correlation strength parameters.



Reducing Bias by Increasing Representation Complexity

The estimates of the critical points of this 5-dimensional flow exhibit reduced bias:

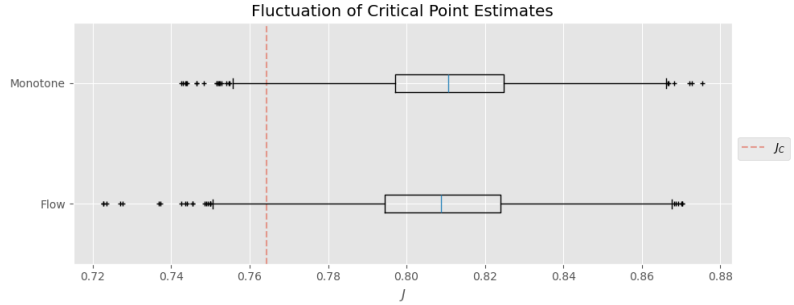


Hou et al. show that this trend continues as the flow dimension increases.

Performance Improvements from Learning the Flow

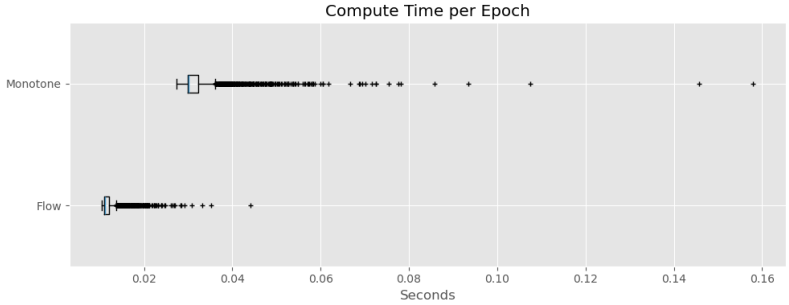
Hou et al. ground their algorithm in field theory by learning the monotone function.

We assert that this is not necessary, as learning the flow directly gives equally good results:



Performance Improvements from Learning the Flow

Moreover, learning the flow is significantly faster, and imposes fewer constraints on the neural network architecture[7].



[7] learning the monotone requires smooth nonlinearities

Conclusions

The MLRG algorithm is a novel way of characterizing phase behavior using renormalization.

Estimates are biased, but we showed bias can be mitigated by:

- Tuning contrastive divergence hyperparameters
- Tuning sampler hyperparameters
- Increasing Ising model expressiveness as proposed by Hou et al.

We also show that large performance gains can be made by directly learning the renormalization group flow, at the cost of theoretical soundness.

Conclusions

For the future, there are many exciting avenues to explore:

- What other sources of bias can we identify?
- How can we improve the algorithm for efficient modelling of highly expressive, high dimension lattices?
- What systems beyond the Ising model can we characterize using this method?