# A Neural Network approach to the Many Body Localized Phase

David Villarreal

Faculty Mentor: Dr. Bryan Clark

University of Illinois at Urbana-Champaign

2017

# Abstract

The Many Body Localized Phase (MBL) is a phase that exists in distinction to the Ergodic phase. As such, the MBL phase violates the Eigenstate Thermalization Hypothesis and does not undergo thermalization. In this paper, the Ergodic to MBL phase transition will be explored with Artificial Neural Networks (ANNs) trained on a Heisenberg spin chain with disorder. One of the main aspects explored in this article is if neural networks can in fact be trained to perceive a difference in both of these phases, and moreover, examine the as-of-now unanswered question of whether or not information about the MBL phase is encoded in the many-body density matrix at infinite temperature $\rho^{eq}(T \to \infty)$. A phase diagram is generated, along with probability curves that predict MBL given a level of disorder. Finally, reverse engineering is done to understand the ANN and evidence is found that the ANN is learning about the entanglement entropy.


# Introduction

Within the field of Condensed Matter Physics, the Many Body Localized Phase has been a topic of interest because of its exotic properties. First and foremost, it is a system that fails to thermalize thus violating the Eigenstate Thermalization Hypothesis. Moreover, it exhibits other behaviors quite distinct from the ergodic phase: the eigenstates follow an area-law entanglement, it has zero DC conductivity, its entanglement spreads logarithmically, the system preserves memory of initial conditions after long times, and there are local conserved quantities of motion, among other things [1].

Not much is known of the MBL phase and there is extensive literature discussing the topic. Something that is unknown about this particular topic is the nature of the ergodic to MBL phase transition. There are theories about mobility edges, where after a certain threshold excitation energy, there exists a transition, given enough disorder. It is known that certain systems, like Floquet systems or the Heisenberg spin chain model exhibit MBL after certain levels of disorder [1]. Many formalisms exist that characterize the MBL phase, which cover transforming the Hamiltonian of the system into an l-bit basis, where localization is apparent and the conserved quantities of motion are clearly represented [2]. Numerical treatments of the phase transition have commonly been done using Exact Diagonalization and studying aspects known a-priori of the MBL phase like the eigenvalue statistics (GOE vs Poisson), the scaling of the entanglement entropy (volume vs area law), or the Kullback-Leiber divergence for neighboring eigenstates [3].

In this paper this phase transition will be studied using Artificial Neural Networks (ANNs). Heuristic algorithms like ANNs are a very good approach to a problem that has a very complex solution, or perhaps no solution at all. They are algorithms that learn, and as such require minimal previous knowledge about the subject from the programmer. Exploring ANNs is an important approach since using Machine Learning for Physics is considered novel. It can be considered another algorithm in the toolbox of the numerical analyst. Since the ANN seems to learn the appropriate physics, the conclusion it gives that information about MBL is stored in the infinite temperature density matrix is highly likely.

I will first explore the Heisenberg spin chain's eigenstates, by training an ANN on its coefficients in the tensor product space basis. From this, the ANN will generate information about the location of the phase transition, along with other properties, such as the effects of increasing the size of the physical system. After doing this, I will study the MBL phase transition with an ANN trained on the many body density matrix at infinite temperature $\rho^{eq}(T \to \infty)$. It is shown that information about MBL is stored in this thermal density matrix. Finally, I feed the ANN entangled states and show that the ANN predicts low probability of MBL for highly entangled states, and high probability of MBL for minimally entangled states.

# Method

Using ANNs to study the MBL phase was done using Google's API TensorFlow. All code was done in Python, which in the TensorFlow environment, runs on a C++ backend. TensorFlow is an open source software library for numerical computation. The code written in Python defines a computational graph, and then this runs on highly optimized C++ code. The Hamiltonian modeled is the Heisenberg chain embedded in a random magnetic field.

$$H = \sum_i S_i^z S_{i+1}^z + \frac{1}{2}(S_i^+ S_{i+1}^- + S_i^- S_{i+1}^+) + h_i S_i^z \tag{1}$$

$$h_i \in \mathrm{unif}(-W, W)$$

All information regarding the physical states, was converted into an image format, thereby allowing the ANNs capacities for image recognition be fully exploited. The ANNs employed are Feedforward Neural Networks with up to 15 layers.

In order to convert information regarding a physical state into an image two procedures were followed. The first just uses the sign structure of the physical state. This means that for the density matrix, or the eigenstate representation of a physical state, only the sign of the coefficients or matrix elements were kept. These signs were then mapped to three different positive integers that could be represented as pixels in an image with uint8 encoding. The other approach followed was to store information about both sign and amplitude. This results in an image two times the size of the previous one, and there is no difference in accuracy. Amplitudes were rounded and mapped to integers in the range 0 to 255, and signs were mapped to three different integers also in the range 0 to 255. The first half of the image represents sign structure, and the second half represents the overall amplitude.

For the case of the density matrix, subsampling the image was a necessary procedure as part of dimensionality reduction. This dimensionality reduction drastically improved the rate of convergence of the neural network. Subsampling was done such that for any matrix $\rho \in R^{256\times256}$, the resulting subsampled image was of size $32 \times 32$.

Several subsampling procedures were followed. One option done was to pick rows and columns with indices [0 8 16 ... 248]. Another option was to select the first $32 \times 32$ corner of the density matrix. Another option was to randomly select two indices for both axes of the image and construct the image by choosing the next 31 indices. Another option was to select the same random index for both axes and then construct the image choosing the next 31 indices. The nature of all these images are different, and but they all have the same results, strongly suggesting that information is in fact encoded in the density matrix $\rho^{eq}(T \to \infty)$. Moreover, choosing images from a certain type, can make precise predictions about images from another type.

In order to generate phase diagrams I train my neural network on disorder realizations known to be ergodic and MBL. Then I feed other disorder realizations and generate histograms of the predicted probability of MBL for that value of $W$. I also plot the average probability of MBL for several values of W. There is a value of W at which that probability is 0.5 and I call that the value of the phase transition. The histograms serve to corroborate this fact because at the neighboring values there occurs a flip over in the histograms' distribution. For example, a histogram at $W = 3$ will be strongly skewed towards a probability of 0 for being MBL while a histogram at $W = 4$

will be strongly skewed towards a probability of 1 of being MBL. I say that the histograms flipped over here and this is in line with the notion that the histograms become uniform at the value of the phase transition where the probability of it being MBL is considered to be 0.5. This procedure is followed for all energy densities, and with this information we can generate a phase diagram with axes $\varepsilon$ and $W$ where $\varepsilon$ is the energy density $\frac{E-E_{min}}{E_{max}-E_{min}}$ and $W$ is the particular level of disorder. To further corroborate the location of this phase transition we can perform an adversarial neural network approach. This consists in lying to the neural network by saying that neighboring values of $W$ correspond to ergodic and MBL states. The top predictive performance should be seen at the phase transition edge, as explained in the results section of this paper.

To perform reverse engineering on our neural network I considered the expansion of any state vector $|\psi\rangle$ as

$$|\psi\rangle = \sum_{\{L\}\{R\}} C_{\{L\}\{R\}} |L\rangle \otimes |R\rangle \tag{2}$$

where I have divided the one dimensional spin chain in half, and considered the bipartite expansion dividing the system into a left (L) half and right (R) half. The coefficients $C_{\{L\}\{R\}}$ form a matrix, and these coefficients are what end up being encoded as images. This matrix has a singular value decomposition (SVD), and the rank of the singular value matrix is one measure of the entanglement entropy. What I do is generate a random matrix, compute its singular value decomposition, modify the singular values according to a certain desired rank, and recompute $C_{\{L\}\{R\}}$ to generate an image with a desired level of entanglement. After training my neural network on $W = 0.5$ and $W = 15$, I feed it these entangled states and see what it predicts as the probability of them being MBL.

# Results

## Eigenstates

Training is done on $W = 0.5$ (Ergodic) and $W = 16$ (MBL). The weights and biases of the neural network generated in this training are then used to do prediction for all $(\varepsilon, W)$.

Examples of what Ergodic and MBL images look like are figures 1 and 2

**Table 1:** Ergodic and MBL images

**Figure 1:** Ergodic, W = 0.5

**Figure 2:** MBL, W = 16

Using many samples of these images we can train our NN to detect the MBL and Ergodic phase. With this fully trained NN we can then do prediction on several levels of disorder $W$ and predict if the state is MBL or Ergodic. Table 2 shows some of the histograms generated.

**Table 2:** Histograms for the probability of a state at W being MBL. Energy density = 0.5

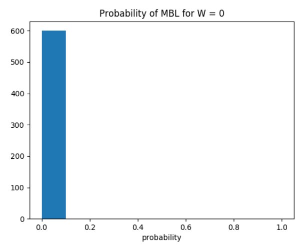


**Figure 3:** W = 0

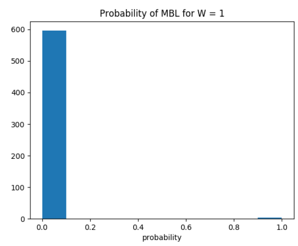


**Figure 4:** W = 1

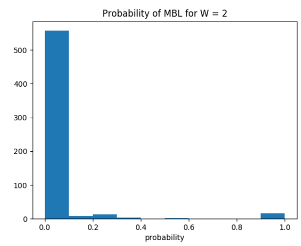


**Figure 5:** W = 2

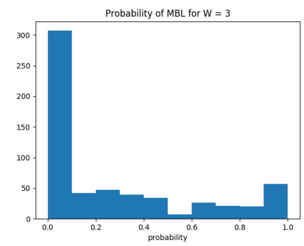


**Figure 6:** W = 3

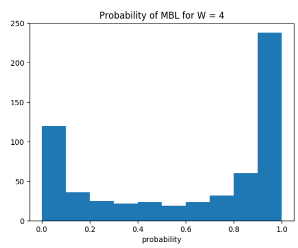


**Figure 7:** W = 4

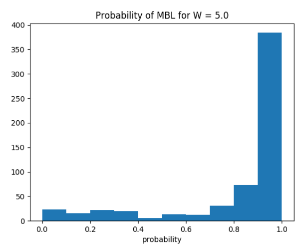


**Figure 8:** W = 5

Between $W = 3$ and $W = 4$ the phase transition must occur, since it's in this neighborhood that the histograms flipped over. Another metric to predict a phase transition is to effectively lie to the neural network saying that neighboring values of disorder correspond to ergodic and MBL states. For example, we lie to the neural network and say that $W = 1$ is ergodic and $W = 2$ is MBL. We do this for various values of neighboring $W$ and plot the top predictive performance of the algorithm. What we observe is seen in figure 9.

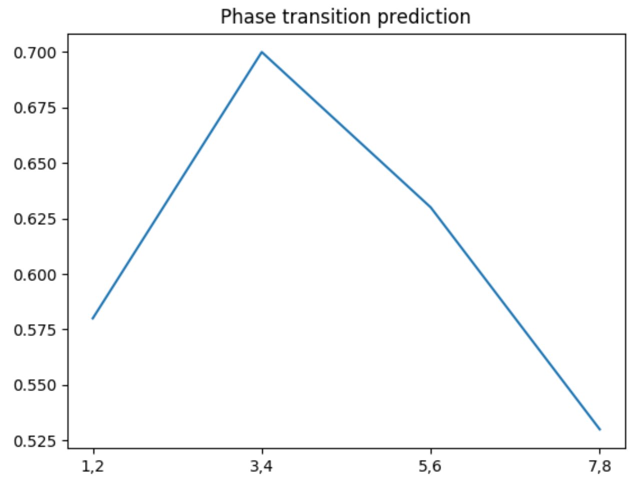


**Figure 9:** Top predictive performance for neighboring values of disorder. We see the top performance when we tell the neural network that $W=3$ is ergodic and $W=4$ is MBL. This plot has been done for $L=8$.

Moreover, I generated the average prediction probability of a state being MBL as a function of the disorder $W$. The resulting graph is figure 10 where the value of the phase transition is that value of $W$ for which $p_{mbl}=0.5$. As we can see this value lies between $W=3$ and $W=4$, and was found using linear interpolation to be $W_{\text{transition}}(\varepsilon=0.5)=3.48$.

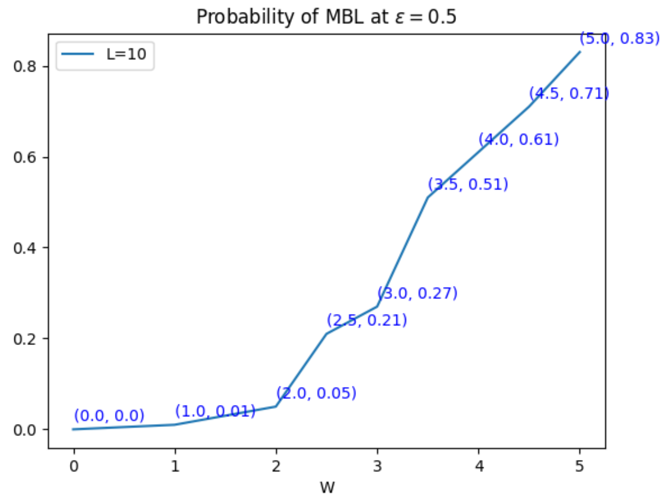


**Figure 10:** Average probability of MBL at various levels of disorder for $\varepsilon = 0.5$ and $L = 10$. The value of the phase transition is 3.48 since it is here that the probability of MBL is 0.5.

I follow the above procedure across different zones of the $(\varepsilon, W)$ plane and generate a phase diagram. The outcome is in figure 11

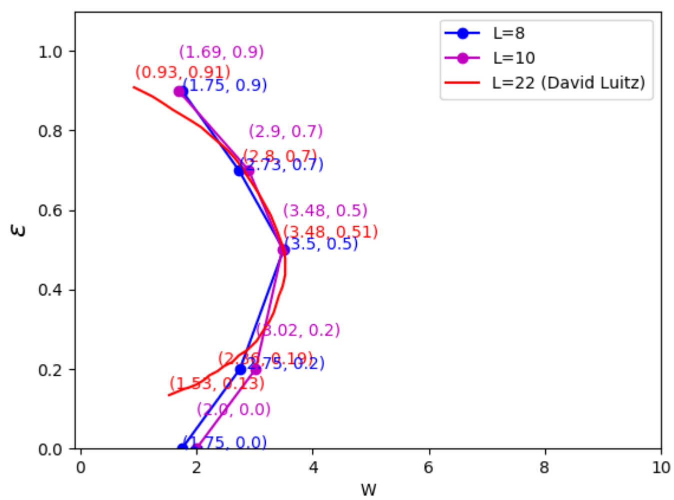


**Figure 11:** L = 8 (blue). L = 10 (purple). L=22 (red), was generated by [3] using Exact Diagonalization and analyzing Eigenvalue Statistics, KL-Divergence, Entanglement Entropy, among other things.

There is also evidence that the neural network learns that phase transitions become sharper as the size of the system increases. This is seen in figure 12.

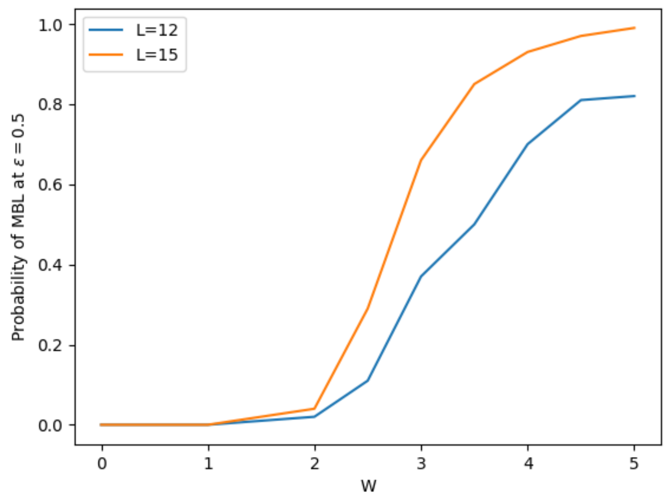


**Figure 12:** $L = 12$ and $L = 15$ systems were analyzed only for the half-filled sector. We see the transition for $L = 15$ is much more like a step function, while the $L = 12$ is broadened. The predicted phase transition for $L = 15$ is not at $3.5$, but close to $2.5$. This data is for $\varepsilon = 0.5$.

## Density matrix

We can perform a similar analysis to the above using the density matrix at infinite temperature. This matrix is given by

$$\rho^{eq}(T \to \infty) = \sum_i |i\rangle \langle i| \tag{3}$$

Where the summation over $i$ is a summation over the eigenstates of the system. I have ignored a normalization constant here that is completely redundant in light of how I encode this matrix into an image. This results in a $2^L \times 2^L$ matrix, which is too hard to analyze using a neural network. I was forced to reduce the dimensionality of the problem by subsampling the matrix into a $32 \times 32$ image. There are several options to do this, as specified in the methods section. The case where the same set of indices used for rows and columns generate images like the figures 13 and 14.

**Table 3:** Ergodic and MBL images, density matrix

**Figure 13:** Ergodic, W = 2

**Figure 14:** MBL, W = 16

We can see that the Ergodic image is more dense, and the MBL phase is more sparse, but moreover, the MBL acquires a peculiar curved structure at the off diagonal zones. This pattern is not always apparent, but it is frequent enough for the programmer to identify it by merely looking at the data. The case where different indices for row and column are used do not generate these highly structured images and look like the figures 15 and 16.

**Table 4:** Ergodic and MBL images, density matrix

**Figure 15:** Ergodic, W = 0.5

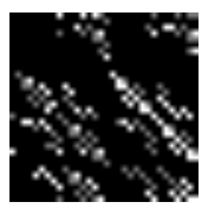

**Figure 16:** MBL, W = 16

Where we can see that there is less structure in the data (all images look very similar to these) and this is also harder to train.

After training the NN on $W = 0.5$ and $W = 16$ we can do prediction on various levels of disorder and trace the average probability of a certain $W$ being MBL. For the three types of subsampling done we get the result on figure 17.

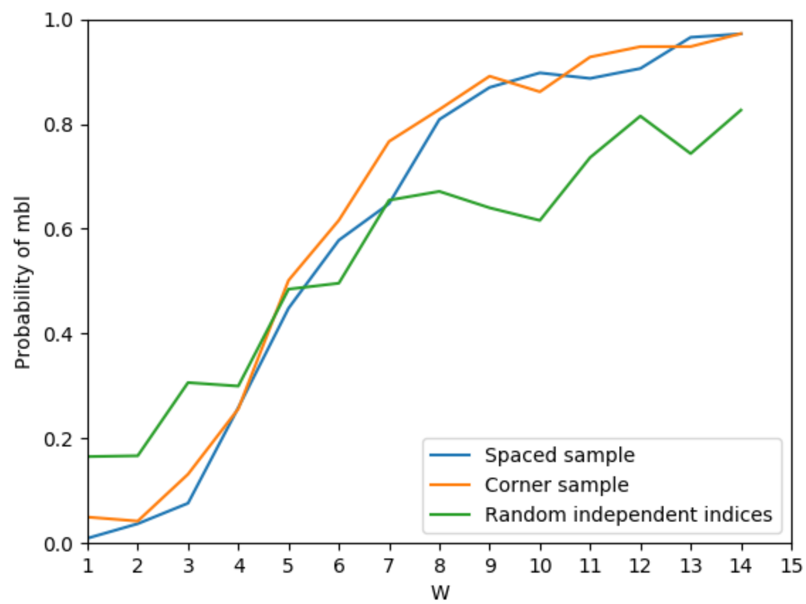


**Figure 17:** Prediction curves after training on $\rho^{eq}(T \to \infty)$, L = 8. The case for random independent indices on both axes is harder to train. However, it is clear that the training is independent to the subsampling method.

We conclude the method is not specific to subsampling, so I compare $L = 8$ and $L = 10$ using only the topmost $32 \times 32$ sector of the matrix. I get the curves on figure 18.

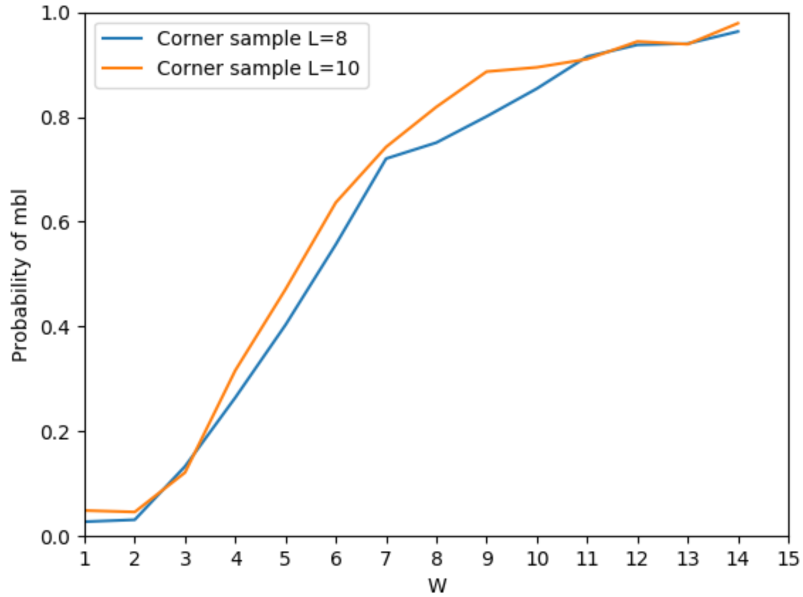


**Figure 18:** Prediction curves after training on $\rho^{eq}(T \to \infty)$, L = 8 and L = 10.

Thus concluding that there is information about MBL stored in the infinite temperature density matrix.

## Reverse engineering

There is strong evidence that the Neural Network is learning about entanglement entropy. The first set of evidence consists of the following graphs

**Table 5:** Correlation with Entanglement Entropy

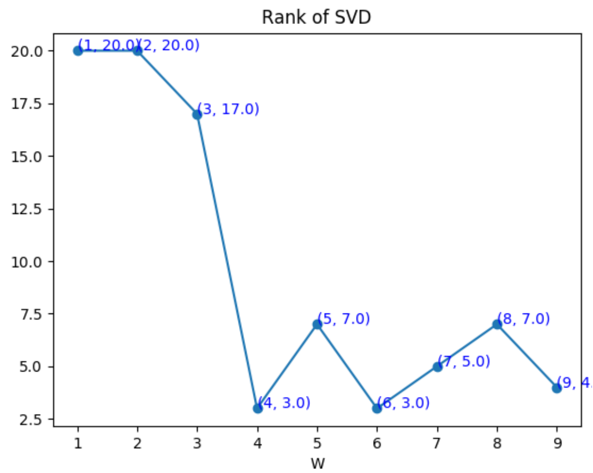


**Figure 19:** Typical rank of SVD for different $W$'s. $L = 10$. There is a sharp decrease at the transition.

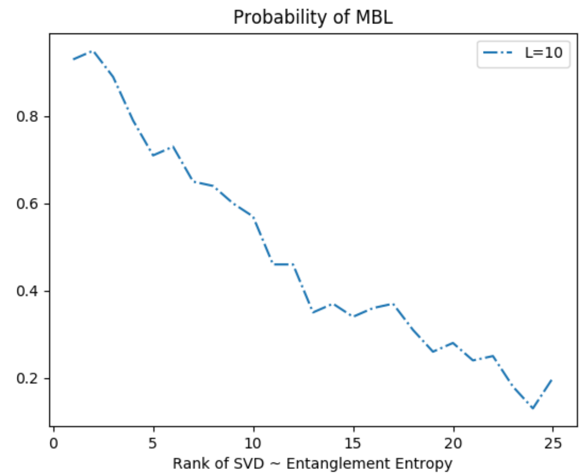


**Figure 20:** Probability of MBL according to a rank of SVD. $L = 10$. There is an obvious negative correlation between $p_{MBL}$ and rank(SVD). Moreover, it is in agreement with figure 18.

Figure 19 plots the rank of the SVD for $C_{\{L\}\{R\}}$ in equation 2, while figure 20 plots what the NN predicts the probability of MBL is for a certain rank of SVD in a randomly created $C_{\{L\}\{R\}}$ matrix. The two graphs are in agreement with one another. In figure 19 we see how the rank plummets after the transition, and in figure 20 we see how the predicted probability is accurate in the ranges specified by figure 19.

The other argument in favor of the idea that the NN is learning about entanglement considers rounding effects. It turns out that if we round our $C_{\{L\}\{R\}}$ matrices to 16 decimal places, the NN does not learn very well and for large systems it doesn't learn at all. The best performance is done when the rounding criterion in equation 4 is followed.

$$10^{-\text{number of decimal places}} \times \dim(\text{Hilbert space}) = O(1) \tag{4}$$

To consider a concrete example, 3 decimal places have to be chosen for $L = 10$ since the dimension of the Hilbert space is 1024. Coincidentally we observe the following behavior in the rank of the SVD in figures 21 and 22.

**Table 6:** Comparison of rank of SVD with respect to rounding

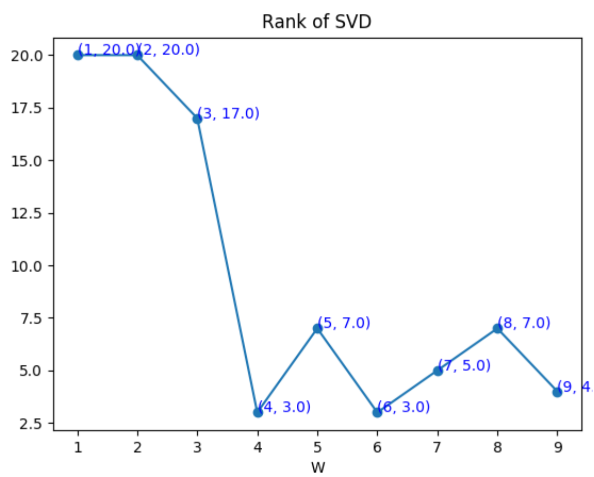


**Figure 21:** Rank of SVD for 3 decimals. As seen in the vertical scale, the rank of the SVD changes significantly for ergodic and MBL states. The NN performs very well here.

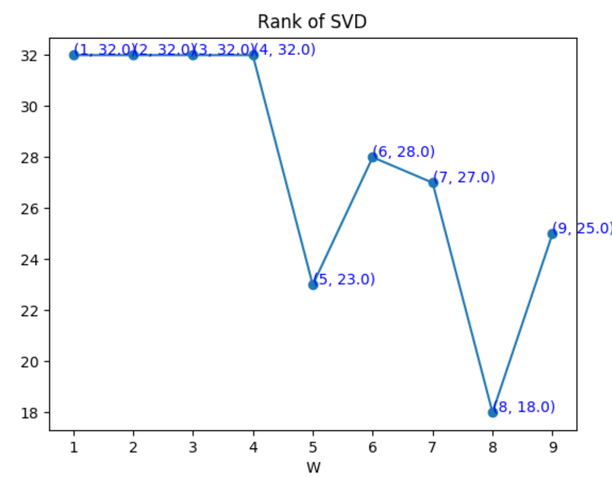


**Figure 22:** Rank of SVD for 16 decimals. As seen in the vertical scale, states known to be ergodic and states known to be MBL do not differ too much on their SVD rank now. The NN performs poorly here.

For a system of size $L = 10$, rounded to 3 decimal places the neural network performs best than any other rounding criterion. At the same time, the greatest change in the rank of the SVD for states known to be ergodic and states known to be MBL occurs here. We can compare this change in the rank of the SVD using the figures 21 and 22. This is evidence in favor of the idea that the neural network is learning about entanglement entropy since the neural network only performs well when there is great difference in rank of the SVD between states that are supposed to be MBL and those that are supposed to be ergodic.

## Conclusions

With this research, it is clear that ANNs learn the appropriate physics about the nature of the MBL phase transition in a Heisenberg spin chain. Few precautions have to be taken to ensure good performance of the neural network. For example, a good rounding criterion has to be selected and dimensionality reduction has to be performed in some cases. When the neural network learns the appropriate physics, machine learning can then be applied to study other aspects of the MBL phase like its phase diagram in the $(\varepsilon, W)$ plane. The phase diagram produced here is very much in line with the one produced by [3]. This project also shows that information about MBL is stored in the infinite temperature density matrix. It is probable, but not entirely proven, that the neural network learns about entanglement entropy. It is hoped for that neural networks and machine learning algorithms can be implemented in future work in physics as they seem to discover the correct physical patterns in data and moreover give credence to preexisting algorithms by giving answers that are in line with them.

## Acknowledgements

I would like to thank NSF and SROP for this great opportunity. Moreover, I would like to thank professor Bryan Clark at UIUC for his excellent guidance in the development of this project.

## References

[1] N. Rahul and D. A. Huse, "Many body localization and thermalization in quantum statistical mechanics," *Annual Review of Condensed Matter Physics*, 2015.

[2] D. A. Huse, R. Nandkishore, and V. Oganesyan, "Phenomenology of fully many-body-localized systems," *Phys. Rev. B 90, 174202*, 2014.

[3] D. Luitz, N. Laflorencie, and F. Alet, "Many-body localization edge in the random-field heisenberg chain," *Phys. Rev. B 91, 081103*, 2015.