Physics-Guided Problem Decomposition for Scaling Deep Learning of High-dimensional Eigen-Solvers: The Case of Schrödinger’s Equation

Sangeeta Srivastava
The Ohio State University
Anuj Karpatne
Virginia Tech
Samuel Olin
Binghamton University
Wei-Cheng Lee
Binghamton University
Viktor Podolskiy
University of Massachusetts Lowell
Anish Arora
The Ohio State University

ABSTRACT
Given their ability to effectively learn non-linear mappings and perform fast inference, deep neural networks (NNs) have been proposed as a viable alternative to traditional simulation-driven approaches for solving high-dimensional eigenvalue equations (HDEs), which are the foundation for many scientific applications. Unfortunately, for the learned models in these scientific applications to achieve generalization, a large, diverse, and preferably annotated dataset is typically needed and is computationally expensive to obtain. Furthermore, the learned models tend to be memory- and compute-intensive primarily due to the size of the output layer. While generalization, especially extrapolation, with scarce data has been attempted by imposing physical constraints in the form of physics loss, the problem of model scalability has remained.

In this paper, we alleviate the compute bottleneck in the output layer by using physics knowledge to decompose the complex regression task of predicting the high-dimensional eigenvectors into multiple simpler sub-tasks, each of which are learned by a simple “expert” network. We call the resulting architecture of specialized experts Physics-Guided Mixture-of-Experts (PG-MoE). We demonstrate the efficacy of such physics-guided problem decomposition for the case of the Schrödinger Equation in Quantum Mechanics. Our proposed PG-MoE model predicts the ground-state solution, i.e., the eigenvector that corresponds to the smallest possible eigenvalue. The model is 150x smaller than the network trained to learn the complex task while being competitive in generalization. To improve the generalization of the PG-MoE, we also employ a physics-guided loss function based on variational energy, which by quantum mechanics principles is minimized if the output is the ground-state solution.

CCS CONCEPTS
• Computing methodologies → Learning paradigms: Neural networks; Supervised learning by regression.

KEYWORDS
physics-guided machine learning, problem decomposition, deep learning, quantum mechanics, eigenvalue problems

1 INTRODUCTION
While neural networks often achieve state-of-the-art performance and generalization in problems with large amounts of data, such as natural language processing (NLP) or computer vision, they often do not work as well in scientific applications where system behavior is governed by high-dimensional eigenvalue problems (HDEs). This means that the power of neural networks is yet to be appropriately exploited in fields such as quantum mechanics, solid state physics, electromagnetism, and classical mechanics where HDEs are common. There are three main challenges that have limited the use of NNs in scientific fields: (i) Large amounts of annotated data are prohibitively expensive to obtain. (ii) Purely data-driven models may extrapolate to produce predictions on unseen data that are physically inconsistent, due to the bias induced by the training data. (iii) In many physical problems, the dimension of the eigenvectors scales in size with the number of system-dependent degrees of freedom, \( f \); for a \( N \)-particle system, the dimension of the eigenvector grows exponentially with \( f \), i.e., \( O(f^N) \); and therefore, a neural network trained to predict an eigenvector shares a similar exponential growth in its output layer, making the learning intractable for high values of \( N \).

We propose to leverage physics knowledge to overcome the aforementioned challenges of deep learning-based solutions for solving HDEs, yielding model scalability and generalization. For the former, i.e., to alleviate the memory requirements induced by exponential growth in the output layer, we first divide the given input-space into multiple input-subspaces in a physically sound manner and then train a specialized expert network \( E \) for each input sub-space. We refer to the architecture comprising of all the experts as Physics-Guided Mixture-of-Experts (PG-MoE) (and describe it further in Section 5.3). The learning task of each expert is “simpler” in two ways: (1) each expert specializes in only its subset of the input space, and (2) instead of predicting all the \( f^N \) coefficients of the eigenvector, each expert predicts only one coefficient at a time, allowing for independent learning of all the \( f^N \) output values. This decomposition approach allows us to avoid the communication overhead among the experts, while also reducing the output space to a single coefficient of the eigenvector, both of which are critical to achieve scalability for large systems. For the latter, i.e., to address generalization, we add a physics-guided (PG) loss that penalizes the NN for violating physical constraints, as has been done in previous work. By restricting the convergence space with PG loss, we are able to build physically consistent and scientifically sound predictive models even when there is scarcity of annotated data.

In this paper, we use Schrödinger’s equation in quantum mechanics as a case study to illustrate the benefits of our approach. We seek to find the eigenvector that corresponds to the smallest possible eigenvalue, also known as the ground-state wavefunction. Since
each particle in our problem possesses two quantum degrees of freedom - originating from its intrinsic angular momentum orientation \( \{ \uparrow, \downarrow \} \) (described in Section 2), the resulting eigenvector to be predicted has a dimension of \( 2^N \). With the proposed combination of the physics-guided decomposition and physics-guided loss, we are able to produce a parameter-efficient, yet generalizable PG-MoE.

The contributions of this work are as follows:

1. The proposed PG-MoE architecture trained on the sub-spaces is substantially smaller than the black-box baseline NN that is trained on the entire input space, and generalizes better on the unseen test data. As an example, for \( N = 10 \) and cosine similarity as the black-box loss, the PG-MoE NN is \( \sim 150 \times \) smaller than the baseline NN. And to achieve similar test performance as the baseline, PG-MoE trains faster requiring \( \sim 60\% \) fewer epochs.

2. To better enforce generalization, we introduce a new PG loss function based on the variational method, which is well established in quantum mechanics theory [13, 30] for searching for the ground state wavefunction. The method is theoretically advantageous because it is specific to the ground state wavefunction, as opposed to the eigenvalue equation which is satisfied for any eigenvector. This ensures that the loss function possesses a global minimum if and only if the predicted wavefunction is the true ground state. Further, this loss does not require a computation of the ground state energy in either the NN or the input data.

3. When physics loss is included in PG-MoE training, the cosine similarity on the unseen test set improves (by \( \sim 11\% \) for \( N = 10 \)) compared to when only black-box loss is used. The similarity is also competitive to that of the baseline model trained with the PG loss.

4. We show the impact of the design choices for PG-MoE training and identify the key parameters that must be tuned to enable convergence.

2 QUANTUM MECHANICS

An Ising model consists of \( N \) interacting electrons, each of which possess an intrinsic degree of freedom called the "spin" angular momentum. The possible values for the spin are defined with respect to an arbitrary axis, they are simply either the spin-up (\( \uparrow \)) or spin-down (\( \downarrow \)) state, which are typically expressed as a basis in Dirac notation: \( \{|z; \uparrow\}, |z; \downarrow\}\). Here, the \( z \)-axis is chosen, which is the convention for this work. A chain with \( N \)-spins can have \( 2^N \) unique combinations, also called configurations. These configurations form an orthonormal basis for the Schrödinger equation, resulting in a ground state wavefunction with \( 2^N \) coefficients.

Here, we consider the one-dimensional transverse Ising model [5] (Equation 1). It describes a system of \( N \) fixed spins and is useful to understand magnetism. The spins are subjected to both a transverse and longitudinal field, denoted \( B_x \) and \( B_z \) respectively. Neighboring spins are coupled through an interaction term, with strength \( J \). Note that we fix \( J = 1 \) for this work. The Schrödinger eigenvalue equation, denoted as \( H\Psi = E\Psi \), where \( H \) is the Hamiltonian (or "total energy") operator and \( E \) is the energy eigenvalue, may be solved to obtain \( \Psi \), the wavefunction, which carries all information about the system. The Hamiltonian for this system is

\[
H = \frac{-J}{N} \sum_{i=1}^{N} \sigma_i^x \sigma_{i+1}^x - \frac{1}{N} \sum_{i=1}^{N} (B_z \sigma_i^z + B_x \sigma_i^x)
\]

(1)

where \( \sigma_i^{x,z} \) is the corresponding Pauli matrix, and the system is subject to a periodic boundary condition. In this work, we seek to predict the ground-state wavefunction \( \Psi_0 \), corresponding to the smallest eigenvalue of the Ising Hamiltonian, \( E_0 \). Wavefunctions are subject to several requirements from the postulates of quantum mechanics [13], one of which is that the (Hilbert) vector space of states must possess an inner product (\( \langle \Psi_1, \Psi_2 \rangle \equiv \langle \Psi_1 | \Psi_2 \rangle \)). Further, the states are normalized such that the overlap of a state with itself is one, \( \langle \Psi | \Psi \rangle = 1 \).

The fields \( B_x, B_z \) may be used to create a quantum phase transition [22]. When \( B_x < 1 \), the system is in the ferromagnetic phase (spins preferring all up or all down in the \( z \)-basis). As the parameter is swept to \( B_x \rightarrow 0 \), the system transitions to the paramagnetic phase (either orientation is possible for each spin). Examples are shown in Figure 1. The magnetic phase of a material has enormous importance in both applied and theoretical condensed matter physics in classification of the electronic properties materials [3]. Further, an understanding of quantum spin forms the backbone of quantum computing [24].

3 PROBLEM SETUP

Consider a training dataset \( D_{Tr} = \{ D_1 \cup D_2 \cup D_3 \} \) where \( D_1 \) corresponds to a labeled dataset obtained by diagonalization solvers
and $D_{UL}$ refers to an unlabeled dataset. $D_L = \{(X^i, H^i), Y^i\}_i=1^n$ consists of transverse and longitudinal magnetic field, $B_x$ and $B_z$ respectively, which contribute elements for the corresponding Hamiltonian matrix $H$. For each data-point, the set of all the spin configurations is denoted by $C = \sigma_1 \sigma_2 \ldots \sigma_N$ where $\sigma_i$ represents the $i^{th}$ configuration. The ground truth $Y^i$ is the ground-state wavefunction $\Psi^i$ of the Hamiltonian $H^i$. The set of unlabeled examples, $D_{UL} = \{(X^i, H^i)\}_i=1^m$ can be leveraged for unsupervised training. The problem then is to learn an NN model with parameters $\theta$ to predict the wavefunction $\hat{\Psi}$ of any unseen $H$, i.e., $\hat{\Psi} = f_{NN}(\theta; X, H)$.

4 PROBLEM DECOMPOSITION

We seek to construct a physics-guided decomposition of the input data. Any decomposition of this nature must satisfy two broad constraints: firstly, it must be physically justifiable, i.e., we may not decompose coupled quantities and, secondly, there must exist a well-defined scientific criterion that can be consistently applied to each input data point. For the Ising model, we choose the total magnetization quantum number, $S_z$ (Equation 2), to decompose the input space of $2^N$ configurations into several input sub-spaces. The orthonormality of the configuration basis vectors (Section 2) satisfies the first constraint. As for the second constraint, it is known that spin configurations with comparable values of $S_z$ respond similarly with respect to external physical parameters. Additionally, the value of $S_z$ classifies the magnetic phase of the system, and it can be used to detect the phase transition. These two qualities make $S_z$ a viable choice for determining the decomposition criterion. We therefore use contiguous values of $S_z$ to form the sub-spaces, and learn an “expert” network for each sub-space. Table 1 illustrates one such decomposition for $N = 4$.

$$S_z = \frac{(\text{Total # of } \uparrow - \text{Total # of } \downarrow)}{2} \quad (2)$$

| Sub-spaces $S_z$ | configurations |
|------------------|----------------|
| $S_z = -2, -1$   | $\{\ldots \uparrow \uparrow \ldots \}$, $\{\ldots \downarrow \downarrow \ldots \}$ |
| $S_z = 0$        | $\{\ldots \uparrow \downarrow \uparrow \ldots \}$, $\{\ldots \downarrow \uparrow \downarrow \ldots \}$ |
| $S_z = 1, 2$     | $\{\ldots \uparrow \uparrow \uparrow \ldots \}$, $\{\ldots \downarrow \downarrow \downarrow \ldots \}$ |

Table 1: One possible decomposition of the configurations based on the $S_z$ value for $N = 4$

For the 10-spin Ising model, we train three $E$ networks where the network assignment is determined according to Table 2.

| Expert No. | $S_z$  | #configurations |
|------------|--------|-----------------|
| 1          | $-5 \leq S_z \leq 0$ | 638             |
| 2          | $1 \leq S_z \leq 3$  | 375             |
| 3          | $S_z = 4, 5$         | 11              |

Table 2: A configuration-to-expert mapping based on the $S_z$ quantum number

5 EXPERIMENTAL PLAN

5.1 Dataset

We consider an $N = 10$ spin system for the Ising model. The training data $D_T$ comprises 300 sets of physical parameters in the range of $J = 1$, $0 \leq B_x < 1$, and $0.06 < B_z \leq 2$, with 10 and 30 steps for $B_x$ and $B_z$ respectively. The minimum value of $B_z$ is set to 0.06 to eschew ground state degeneracy. From each of the 10 unique bins of $B_x$ values with 30 different $B_z$, we randomly choose 4 data points; 2 for the test set and 2 for the validation set. The resulting test set is denoted by $D_T$. The distribution of the data is listed in Table 3.

Due to the computation overhead of the exact diagonalization to produce annotated data, we also leverage unlabeled data in the training. We test the generalization of the NN for various sizes of annotated and unlabeled data. The network obtains the most supervision when all of the $B_x$ in $D_T$ are used in the annotated set $D_L$. With the decrease in the amount of data in $D_L$, the amount of supervision decreases. The most restrictive learning setting occurs when only half of the training data, i.e., from $B_x \leq 0.5$, make up the annotated set $X$, whereas the other half, from $0.5 < B_x < 1$, make up the unlabeled set $X_{UL}$.

| $B_x$ | #Train (labeled) | #Train (unlabeled) | #Valid | #Test |
|-------|-----------------|-------------------|--------|-------|
| $B_x < 1$ | 130             | 130               | 20     | 20    |
| $1 \leq B_x \leq 2$ | 0              | 0                 | 20     | 300   |

Table 3: Distribution of train, test and validation data

Additionally, a dataset $D_{UT}$ is generated from the range $1 \leq B_x \leq 2$ to test the generalizability of the trained NN on unseen transverse magnetic field values. Here, $UT$ stands for Unseen Test. Specifically, we train the NN with states only in the ferromagnetic phase and hold out the paramagnetic phase states as a stress test.

5.2 Loss

Cosine Similarity Loss. A pure black-box technique to train the NN involves optimizing the network for loss—such as cosine similarity (CS) loss or mean square error (MSE)—between the true wavefunction $\Psi$ and the predicted wavefunction $\hat{\Psi}$. We prefer CS loss over MSE loss because of the normalized inner product requirement satisfied by the wavefunction. Further, the eigenstates may have an overall phase ambiguity [30] and with MSE loss the predicted states may not capture phase whereas with CS loss both the magnitude and direction for the ground-state wavefunction will be captured. For $M$ number of data-points, the cosine loss, $L_{CS}$, is given by Equation 3.

$$L_{CS} = \frac{1}{M} \sum (1 - \cos(\theta)) \quad (3)$$

$$\cos(\theta) = \frac{\langle \Psi \mid \hat{\Psi} \rangle}{\|\Psi\| \|\hat{\Psi}\|} \quad (4)$$

Physics Loss. The physics loss we propose to use is based on the variational principle of quantum mechanics [13] that states that the expectation value of the energy computed with any state except for the ground state will overestimate the ground state energy;
Figure 2: Physics-Guided Mixture-of-Experts (PG-MoE) pipeline. The gating component distributes the $2^N$ configurations and the corresponding coefficients among the experts. The configurations with the same color are mapped to the same expert. Before the loss calculation, the concatenated output from all the experts $\hat{\Psi}$ are reordered to preserve the configuration ordering of the true $\Psi$. Here, we show $N = 4$ and number of experts as 3.

$$\langle \Psi | H | \Psi \rangle \geq E_0.$$ As such, the energy expectation value may only be a global minimum for the true ground state wavefunction - the quantity the NN learns to predict. We propose to use the NN predicted wavefunction $|\hat{\Psi}\rangle$ as a variational wavefunction in a new PG loss term such that the network must predict a wavefunction which lowers Equation 5 as much as possible.

$$L_{PG} = \frac{1}{M} \sum \frac{\langle \hat{\Psi} | H | \hat{\Psi} \rangle}{\langle \Psi | \Psi \rangle}$$

(5)

This loss only depends on $H$, which may be computed for any input without re-diagonalizing for the ground truth energy $E_0$.

The overall loss, $L$, is given by the Equation 6 where $\lambda_{CS}$ and $\lambda_{PG}$ are hyper-parameters to determine the scaling of each loss.

$$L = \lambda_{CS} L_{CS} + \lambda_{PG} L_{PG}$$

(6)

5.3 Architectures

Baseline: Similar to [12], we train a feed-forward network (FFN) with $2^N$ nodes in the output layer to predict the ground-state wavefunction with $2^N$ coefficients. With the $B_x$ and $B_z$ as the inputs, the network is optimized to minimize the loss in Equation 6. This serves as our baseline architecture. Figure 3 illustrates the design of a baseline and an expert model.

Physics-Guided Mixture-of-Experts (PG-MoE): The PG-MoE consists of a set of $n$ expert networks $E_1, \ldots, E_n$, a gating component $G$ and a reordering component $R$. Each expert is a FFN with only 1 output node to predict one coefficient $y_k$ of a wavefunction corresponding to a single configuration $\sigma_k$. Therefore, in addition to $B_x$ and $B_z$, each expert also takes the configuration $\sigma_k$ as the input. We use a vector of 1s and 0s representing $\uparrow$ and $\downarrow$ to feed $\sigma_k$ to the NN. For example, an input $\{1,0,0,0\}$ represents a configuration $\sigma = \{\uparrow\downarrow\downarrow\downarrow\}$. Unlike the gating component used in various MoE architectures in the literature [31] that need a nonlinear optimization routine, our gating component is non-parameterized and does not have to be trained. When given an input with all the $2^N$ configurations, the gating component simply calculates the value of $S_z$ for each and distributes the corresponding batches of $(B_x, B_z, \sigma_k) \rightarrow y_k$ pairs among the experts. The output from all the experts are then concatenated. As shown in Figure 2, contiguous configurations do not yield the same value of $S_z$ and thus alter the order of the predicted coefficients in $\Psi$ when concatenated. Therefore, the reordering component is used to re-arrange the outputs to their original ordering before calculating the loss. The design of the PG-MoE pipeline is shown in Figure 2.

5.4 Training and Evaluation

Since the PG loss is independent of the ground-truth $\Psi$, we calculate only the PG loss for data corresponding to unlabeled data $D_{UL}$. For the annotated data $D_L$, the network is optimized for both the $L_{CS}$...
and $L_{PG}$. We evaluate the generalization performance of NN model on the test sets $D_T$ and $D_{UT}$ by using mean value of the cosine similarity between our predicted wavefunction $\Psi$ and the ground-truth $\Psi$.

5.5 Hyperparameter Optimization

5.5.1 Architecture. We sweep across different combinations of the number of hidden layers (d) and number of hidden units (n) in each layer to find the architecture that optimizes the validation performance. For the PG-MoE, we use the same architecture for the experts. For the baseline, we use a FFN with $d=2$ and $n=2000$. As for each expert, we find $d=2$ and $n=200$ to be ideal.

5.5.2 Training. Given the small amount of data, we use a small batch size of 8 for all of the training. The loss scaling factors, $(\lambda_{CS}, \lambda_{PG})$, that yield the best validation performance for the baseline and the PG-MoE are (1, 100) and (100, 200) respectively. We train both the baseline and PG-MoE with stochastic gradient descent (SGD) [6]. We find the training of PG-MoE to be unstable, resulting in the oscillations in the gradient descent, and three ways to mitigate the instability: (1) using a small learning rate, (2) using a momentum [32] of 0.99, and (3) decaying the learning rate by a factor of 0.987 when the validation loss does not improve for 8 consecutive epochs. For all of the training, we use early stopping [8]–training is stopped when the validation performance on the seen data does not improve for 30 epochs. This avoids overfitting of the networks, especially the baseline architecture, which consists of ~6 million parameters.

Another key finding is that if PG loss is introduced early in the learning process, the network not only runs the risk of getting stuck in local minima but also the convergence becomes slower. As a result, we only use the cosine loss for the first few epochs before switching to the combination of both the loss components in Equation 6. This correspond to epoch = 80 and epoch = 55 for the baseline and PG-MoE when both are trained for a maximum of 300 and 400 epochs respectively. For each experimental setup, we either run the models for five distinct seeds and choose the one that performs the best across all runs for that study or use mean and standard deviation to combine the five numbers.

6 RESULTS

6.1 Evaluation of Decomposition

Figure 4 compares the generalization performance of the baseline and PG-MoE on the unseen test set $D_{UT}$. Both the networks are trained on different subsets of labeled data and optimized for CS loss (i.e., $\lambda_{PG}=0$). Although PG-MoE outperforms the baseline for all categories of training data, the gap between the two increases as the number of labeled data in $D_L$ decreases from 260 to 130. For $D_L$: $B_x < 0.5$, the mean CS of the PG-MoE on $D_{UT}$ is better than baseline by 2% while requiring 2x fewer epochs to train. On average, PG-MoE requires ~60% fewer epochs than the baseline to achieve similar test performance.

6.2 Effect of Physics Loss

Table 4 provides a more detailed comparison of the baseline and the PG-MoE model on both the seen and the unseen test ranges: $D_T$ and $D_{UT}$. To determine how sensitive the networks are, we run both models for 5 different seed values and compute the mean and standard deviation of cosine similarity for each. When trained on $D_T$: $B_x < 0.5$ and $D_{UT}$: $0.5 < B_x < 1$, PG-MoE requires ~150x lesser parameters than the baseline architecture to achieve competitive performance. For both networks, PG loss yields a perfect cosine similarity on the seen test set $D_T$. The test performance on $D_{UT}$ also improves by ~12% for both the baseline and the PG-MoE. It is worth noting that the standard deviation in CS for both models is much lower when they are trained with PG loss. This demonstrates how PG loss helps in limiting the convergence region, making the training less sensitive to variations.

Another observation is that the PG-MoE has a larger standard deviation than the baseline for the unseen test $D_{UT}$. This is because, as mentioned in Section 5.5.2, the training procedure of PG-MoE is susceptible to oscillations in the gradient descent, leading to variations in performance. Note that the optimization process of training multiple experts in PG-MoE is much more complex than training a single baseline, where the gradient descent of each expert to their respective local minima may show competing directions. Also note that the selection of the loss scaling factors, $\lambda_{CS}$ and $\lambda_{PG}$ in Equation 6, at different stages of PG-MoE training can influence variations in performance. Further algorithmic improvements in training mixture of experts are needed to reduce the variance in PG-MoE results.

6.3 Annotated Data and Generalization

As shown in Table 5, PG loss not only helps generalize better in the unseen test regime, it also helps achieve a consistent test performance as the amount of the labeled data decreases. When the amount of supervised data is decreased by 50%, i.e., from 260 to 130, PG-MoE optimized only with $L_{CS}$ shows a ~6% drop in performance whereas the network trained with the PG loss has similar performance.

It is also worth noting that the test performance on unseen $B_x > 1$ is not solely related to the number of labeled data-points. Consider two ranges of $D_L$: $B_x < 0.3 \cup 0.7 < B_x < 1$ and $B_x < 0.6$. Although

![Figure 4: Mean cosine similarity on unseen test data $D_{UT}$ of the baseline and the PG-MoE networks when trained for 300 and 500 epochs. The annotated labels show the epoch where the training early stopped.](image-url)
### Table 4: Comparison of cosine similarity of the Baseline and PG-MoE networks on the $D_T$ ($B_x < 1$) and $D_{UT}$ ($B_x > 1$) test sets. $d$ and $n$ correspond to the number of hidden layers and the number of neurons in each layer of the NN.

| Model    | Architecture | #params | CS (w/o $L_{PG}$) | CS (with $L_{PG}$) |
|----------|--------------|---------|-------------------|-------------------|
|          |              |         | $B_x < 1$ | $B_x > 1$ | $B_x < 1$ | $B_x > 1$ |
| Baseline | $d_{2,n,2000}$ | 6093024 | 0.996 ± 0.0008 | 0.874 ± 0.0028 | 1.000 ± 0.0000 | 0.981 ± 0.0003 |
| PG-MoE   | $d_{2,n,200}$ | 39903 | 0.996 ± 0.0008 | 0.882 ± 0.0083 | 1.000 ± 0.0000 | 0.977 ± 0.0044 |

It is intriguing that this observation is very consistent with the physical idea of the phase transition. A small but finite value of $B_z$ is enough to align the spins along the $z$ direction, and applying a magnetic field along $x$ direction could turn the spin’s orientation away from $z$ direction. In other words, $B_z$ and $B_x$ are in fact two competing energy scales to determine the spin orientation. As a result, the stronger $B_z$ is, the fluctuations introduced into the system are larger, which makes the prediction more challenging, and hence the performance is lower.

### 7 ABLENT STUDIES

We design a number of ablation experiments to explore the impact of various design decisions.

#### 7.1 Unlabeled Data and Generalization

While keeping the labeled data $D_L$ fixed, we vary the amount of unlabeled data in the training from $0.5 \leq B_x < 1.5$ and test the generalization on the $D_{UT}$ dataset. As shown in Figure 6, PG-MoE trained with PG loss but with no unlabeled data is only slightly better than the one trained only with the black-box loss. The mean CS on the test set consistently increases with the increase in the amount of unlabeled data. The mean CS of the PG-MoE trained with 260 unlabeled data-points is $13\%$ better than the model trained with no unlabeled input.
7.2 Comparison of Different Decomposition Strategies

In Table 6, we compare the cosine similarity of six types of expert assignments on the unseen test data \(D_{UT}\). In addition to the various ways in which we can partition the contiguous values of \(S_2\), we also evaluate two comparatively limiting cases of the \(S_2 \rightarrow E\) mapping: (1) randomly grouping \(S_2\) in 3 sub-groups for 3 expert architectures, and (2) assigning all the configurations, i.e., all the \(S_2\), to only one architecture. For the other partitions, we vary the number of experts from 2 (PG-MoE\(1\)) to 4 (PG-MoE\(4\)). PG-MoE\(3\) has the same decomposition as in Table 2. For each decomposition, we run the resulting models for 5 distinct seed values and compute the mean and standard deviation of CS for each.

When optimized with the PG-loss, the random mapping of \(S_2 \rightarrow E_i\) not only yields the worst test cosine similarity, it also has ~8x more spread than PG-MoE\(3\). The PG-MoE models become less susceptible to randomization (compare the deviation from PG-MoE\(1\) through PG-MoE\(4\)) as \(S_2 \geq 0\) is sub-divided. We believe this is related to the fact that the phase transition is expected to occur in the range \(0 \leq S_2 \leq N/2\), resulting in large variations in the output coefficients. This makes learning in this range more difficult than learning in the \(S_2 \leq 0\) regime, which comprises of very low coefficient values in the wavefunction. As a result, dividing \(S_2 \geq 0\) into multiple partitions helps in decomposing the overall task into simpler and smaller tasks, thus resulting in better CS values of PG-MoE\(2\), PG-MoE\(3\), and PG-MoE\(4\) relative to PG-MoE\(1\). Nevertheless, over-partitioning \(S_2 \geq 0\) also reduces the training sizes available to each expert and make them under-representative, thus degrading performance. As a result, we can see that the cosine similarity of PG-MoE\(4\) is slightly smaller than that of PG-MoE\(3\).

7.3 Hyperparameter Selection

We evaluate the impact of alternate hyperparameter selection with our chosen values in the first row of Table 7. Out of five hyperparameters, namely (i) learning rate \((lr)\), (ii) epoch where PG-loss is introduced \((pg\)e), (iii) decay factor for the learning rate scheduler \((y)\), (iv) momentum \((mom)\) for SGD, and (v) loss scaling factors \((\lambda_{CS}, \lambda_{PG})\), we vary one parameter at a time keeping others constant. The table’s empty cells imply that the values in the first row are used.

It is crucial to not train the model with the PG-loss from the very beginning, i.e., \(pg\)e = 0. This is true even for the baseline architecture, and is consistent with the findings in [12]. We find PG-MoE architecture to be prone to oscillations, therefore, it skips the global minima in the absence of momentum \((mom = 0)\) or learning rate decay \((y = 0)\), or when the initial learning rate is high \((lr = 0.0004)\). While both \(mom\) and \(y\) has a small search space of \([0.9, 0.985, 0.987, 0.99]\), the parameters \((\lambda_{CS}, \lambda_{PG})\) are particularly difficult to tune. We make two observations while tuning for these scaling factors. First, since the cosine loss can get to a very small value at the later stages of the training, \(\lambda_{CS} = 1\) can lead to vanishing gradients, we, therefore, set this to a high value and empirically find \(\lambda_{CS} = 100\) to be the best choice. (In future work, we will implement adaptive tuning to choose the best scalars). Second, we find that the network is able to extrapolate better when \(\lambda_{CS} < \lambda_{PG}\). This can be seen when we compare the scale factor \((100, 200)\) with \((100, 100)\) and \((100, 50)\) in the Table 7.

8 RELATED WORK

Quantum Ising Model. Neural networks have been employed to study the quantum Ising model previously. One commonly used method is the restricted Boltzmann machine (RBM) that is designed to learn the distributions of coefficients for spin configurations in the ground state using a generic model. Success has been shown to reproduce the results of physical properties including the magnetization, spin-spin correlation function, and entanglement entropy [1, 9, 10, 15, 17, 25, 33]. These studies focused on learning particular physical properties instead of the ground state wave function.
Physics-Guided Machine Learning. Physics-guided machine learning is becoming an increasingly common method for solving problems in a wide variety of physics dependent fields such as fluid mechanics [7, 20, 28, 34], electromagnetism [19, 26, 37], thermodynamic modeling [2, 16, 27, 36], and even in medical engineering [29]. By imposing physical constraints to respect any symmetries [35], invariances [20], or conservation principles [4], researchers are able to constrain the space of admissible solutions to a manageable size even with a few hundred data-points.

In materials science, most important quantum phenomena are dominated by the ground state solution of the Schrödinger equation. Obtaining the ground state solution for a system with many interacting electrons remains one of the most challenging problems in theoretical physics. A traditional approach is to build up a trial function with multiple parameters, namely an ansatz, to represent the ground state solution, and the parameters in the ansatz are determined by satisfying the physical constraints. Studies of deep learning models using the physics-guided ansatz as the learning target have been performed [14, 21], but the applicability of the same ansatz to radically different systems is unclear.

The Ising model has also been studied recently in CoPhy-PGNN [12]. As is the case in that work, we leverage both unlabeled data and physics loss for better extrapolation. But unlike their neural network, which is also the basis of our baseline architecture, our proposed PG-MoE is scalable for large physical systems. CoPhy-PGNN trains a neural network to predict the entire ground state wavefunction for an $N=4$ system. With just $2^4 = 16$ coefficients in the output, it is a much smaller problem as compared to the $N=10$ system with $2^{10} = 1024$ output values in the wavefunction. With our baseline comparison, we show that PG-MoE can achieve similar test performance as the architecture in CoPhy-PGNN [12] while being smaller by more than two orders of magnitude. Additionally, the PG loss presented in CoPhy-PGNN applies to any eigenvector whereas our new loss is true for only the vector we seek to predict.

Problem Decomposition. Extreme classification tasks in domains like computer vision and NLP have seen a scaling issue similar to Quantum Mechanics. The challenge essentially originates from the necessity to train a deep model to predict millions of classes, which result in a memory and computation explosion in the last output layer. MACH (Merged-Average Classification via Hashing) [23] employs a hashing based divide-and-conquer algorithm to map $K$ classes into $B$ buckets, where $B \ll K$, and learns a classifier for each bucket. Another method of scaling large neural networks is to use a mixture-of-experts (MoE) [11, 18, 31], which involves training multiple expert networks or learners to handle a subset of the entire training set. A gating network is trained to distribute the inputs and combine the corresponding outputs to produce a final output. Although our idea of problem decomposition share the same underlying principle of divide-and-conquer, our approach is different in two respects: (1) we use physics to design the input sub-spaces, and (2) the gating component in our proposed PG-MoE need not be learned and involves a computationally cheap operation (Equation 2).

9 CONCLUSIONS AND FUTURE WORK

We have demonstrated the use of physics to decompose a complex machine learning problem into smaller problems, which avoids the exponential growth in the neural network’s output layer, which is prevalent in high-dimensional eigenvalue problems. The approach leads to learning multiple parameter-efficient expert networks, one for each component problem sub-space, to yield a Physics-Guided Mixture-of-Experts (PG-MoE). The PG-MoE in our case study is ~150x smaller but has similar extrapolation ability as the model trained on the complex problem. Along with decomposition, we also employed fundamental quantum mechanics in designing a loss function that included the variational method, which better guides the learning of the eigenvector we seek to predict. Notably, while we show the efficacy of our approach for the Ising chain model for 10 spins, our solution is scalable for much larger physical systems. Our work has useful connotations in other scientific applications like quantum spin problems in two-dimensions, interacting fermionic system, and physical systems governed by partial differential equations which possess matrix representation.

For future work, an important topic is to characterize an upper bound for the scalability of the PG-MoE approach, which may be studied by increasing the total number of spins as well as promoting the system to two-dimensions. We will also focus on improving the stability and sensitivity of PG-MoE training. Another intriguing direction is to incorporate traditional theoretical approaches like the quantum Monte Carlo method to train the PG-MoE for large systems. Because labeled data is not easily obtained in these cases, the advantages of PG-MoE for efficient neural network architecture and PG loss for extrapolation would get further tested.

REFERENCES

[1] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. 2018. Quantum Boltzmann Machine. Phys. Rev. X 8 (May 2018), 021050. Issue 2. https://doi.org/10.1103/PhysRevX.8.021050

[2] Sina Amini, Saidi, Ehsan Haghighat, Trevor Campbell, Anoush Poursartip, and Reza Vaziri. 2021. Physics-informed neural network for modelling the thermo-chemical curing process of composite-tool systems during manufacture. Computer Methods in Applied Mechanics and Engineering 384 (Oct 2021), 113595. https://doi.org/10.1016/j.cma.2021.113595

[3] Neil W. Ashcroft and David N. Mermin. 1976. Solid State Physics. Holt-Saunders.

[4] Tom Beucler, Stephan Rasp, Michael Pritchard, and Pierre Gentine. 2019. Achieving conservation of energy in neural network emulators for climate modeling. arXiv preprint arXiv:1906.06622 (2019).

[5] Oz F. de Alcantara Bonfim, B. Boechat, and J. Florencio. 2019. Ground-state properties of the one-dimensional transverse Ising model in a longitudinal magnetic field. Physical Review E 99, 1 (Jan 2019). https://doi.org/10.1103/physreve.99.012122

[6] Léon Bottou, Frank I. Curtis, and Jorge Nocedal. 2018. Optimization methods for large-scale machine learning. SIAM Review 60, 2 (2018), 223–311.

[7] Shengze Cai, Zhiping Mao, Zicheng Wang, Mingyang Jin, and George E. Karniadakis. 2021. Physics-informed neural networks (PINNs) for fluid mechanics: A review. arXiv:2105.09506 [physics.flu-dyn]

[8] Rich Caruana, Steve Lawrence, and C. Giles. 2000. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. Advances in neural information processing systems 13 (2000).

[9] Francesco D’Angelo and Lucas Böttcher. 2020. Learning the Ising model with generative neural networks. Physical Review Research 2, 2 (2020), 023266.

[10] Dong-Ling Deng, Xiaoqeng Li, and S. Das Sarma. 2017. Quantum Entanglement in Neural Network States. Phys. Rev. X 7 (May 2017), 021021. Issue 2. https://doi.org/10.1103/PhysRevX.7.021021

[11] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2013. Learning factored representations in a deep mixture of experts. arXiv preprint arXiv:1312.4334 (2013).

[12] Mohammad Elhamod, Jie Bu, Christopher Singh, Matthew Redell, Abantika Ghosh, Viktor Podolskiy, Wei-Cheng Lee, and Anuj Karpatne. 2021. CoPhy-PGNN: Learning Physics-guided Neural Networks with Competing Loss Functions for Solving Eigenvalue Problems. arXiv:2007.01420 [cs.LG]
