

Learning Physics-guided Neural Networks with Competing Physics Loss: A Summary of Results in Solving Eigenvalue Problems

Mohammad Elhamod^{1*}, Jie Bu^{1*}, Christopher Singh², Matthew Redell², Abantika Ghosh³, Viktor Podolskiy³, Wei-Cheng Lee², Anuj Karpatne¹

¹Department of Computer Science, Virginia Tech,

²Department of Physics, Binghamton University,

³Department of Physics and Applied Physics, University of Massachusetts Lowell,

*Equal contribution,

{elhamod, jayroxis, karpatne}@vt.edu, {csingh5, mredell1, wlee}@binghamton.edu, {abantika, viktor_podolskiy}@uml.edu

Abstract

Existing work in Physics-guided Neural Networks (PGNNs) have demonstrated the efficacy of adding single PG loss functions in the neural network objectives, using constant trade-off parameters, to ensure better generalizability. However, in the presence of multiple physics loss functions with competing gradient directions, there is a need to *adaptively* tune the contribution of competing PG loss functions during the course of training to arrive at generalizable solutions. We demonstrate the presence of competing PG losses in the generic neural network problem of solving for the lowest (or highest) eigenvector of a physics-based eigenvalue equation, common to many scientific problems. We present a novel approach to handle competing PG losses and demonstrate its efficacy in learning generalizable solutions in two motivating applications of quantum mechanics and electromagnetic propagation.

1 Introduction

With the increasing impact of deep learning methods in diverse scientific disciplines (Appenzeller 2017; Graham-Rowe et al. 2008), there is a growing realization in the scientific community to harness the power of artificial neural networks (ANNs) without ignoring the rich supervision available in the form of physics knowledge in several scientific problems (Karpatne et al. 2017a; Willard et al. 2020). One of the promising lines of research in this direction is to modify the objective function of neural networks by adding loss functions that measure the violations of ANN outputs with physical equations, termed as *physics-guided (PG) loss functions* (Karpatne et al. 2017b; Stewart and Ermon 2017). By anchoring ANN models to be consistent with physics, PG loss functions have been shown to impart generalizability even in the paucity of training data across several scientific problems (Jia et al. 2019; Karpatne et al. 2017c; Raissi, Perdikaris, and Karniadakis 2019; de Bezenac, Pajot, and Gallinari 2019). We refer to the class of neural networks that are trained using PG loss functions as physics-guided neural networks (PGNNs).

While some existing work in PGNN have attempted to learn neural networks by solely minimizing PG loss (and thus being label-free) (Raissi, Perdikaris, and Karniadakis 2019; Stewart and Ermon 2017), others have used both PG loss and data label loss using appropriate trade-off hyper-parameters (Karpatne et al. 2017c; Jia et al. 2019). However, what is even more challenging is when there are multiple physics equations with *competing* PG loss functions that need to be minimized together, where each PG loss may show multiple local minima. In such situations, simple addition of PG losses in the objective function with constant trade-off hyper-parameters may result in the learning of non-generalizable solutions. This may seem counter-intuitive since the addition of PG loss is generally assumed to offer generalizability in the PGNN literature (Karpatne et al. 2017c; de Bezenac, Pajot, and Gallinari 2019; Shin, Darbon, and Karniadakis 2020). This motivates us to ask the question: *is it possible to adaptively balance the importance of competing PG loss functions at different stages of neural network learning to arrive at generalizable solutions?*

In this work, we introduce a novel framework of *CoPhy*-PGNN, which is an abbreviation for *Competing Physics Physics-Guided Neural Networks*, to handle competing PG loss functions in neural network training. We specifically consider the domain of scientific problems where physics knowledge are represented as eigenvalue equations and we are required to solve for the highest or lowest eigen-solution. This representation is common to many types of physics such as the Schrödinger equation in the domain of quantum mechanics and Maxwell's equations in the domain of electromagnetic propagation. In these applications, solving eigenvalue equations using exact numerical techniques (e.g., diagonalization methods) can be computationally expensive especially for large physical systems. On the other hand, PGNN models, once trained, can be applied on testing scenarios to predict their eigen-solutions in drastically smaller running times. We empirically demonstrate the efficacy of our *CoPhy*-PGNN solution on two diverse applications in quantum mechanics and electromagnetic propagation, highlighting the generalizability of our proposed approach to many physics problems.

2 Background

2.1 Overview of Physics Problems:

The physics of the problem is available in the form of an eigen-value equation of the form: $\hat{A}\mathbf{y} = b\mathbf{y}$, where, for a given input matrix \hat{A} , b is an eigenvalue and \mathbf{y} is the corresponding eigenvector. We are interested in solving the lowest or highest eigen-solution of this equation in our target problems. Here, we provide a brief overview of the two target applications.

Quantum Mechanics: In this application, the goal is to predict the ground-state wave function of an Ising chain model with $n = 4$ particles. This problem can be described by the Schrödinger equation $\mathbf{H}\hat{\Psi} = \hat{E}\hat{\Psi}$, where \hat{E} , the energy level, is the eigenvalue; $\hat{\Psi}$, the wave function, is the eigenvector, and \mathbf{H} , the Hamiltonian, is the matrix. Since the ground-state wave function corresponds to the lowest energy level, we are interested in finding the lowest eigen-solution of this eigen-value equation. To be able to execute a detailed analysis, we choose a small problem scale ($n = 4$) for this application.

Electromagnetic Propagation: To illustrate our model’s scalability to large systems, we consider another application involving the propagation of the electromagnetic waves in periodically stratified layer stacks. The description of this propagation can be reduced to the eigenvalue problem $\hat{A}\vec{h}_m = k_z^{m2}\vec{h}_m$ where k_z^{m2} , the propagation constant of the electromagnetic modes along the layers, is the eigenvalue; and \vec{h}_m , the coefficients of the Fourier transform of the spatial profile of the electromagnetic field, is the eigenvector. It is important to note for this application that these quantities are complex valued, and that we are interested in the largest eigenvalue rather than the smallest.

2.2 Related work in PGNN:

PGNN has found successful applications in several disciplines including fluid dynamics (Wang, Wu, and Xiao 2017, 2016; Wang et al. 2017), climate science (de Bezenac, Pajot, and Gallinari 2019), and lake modeling (Karpatne et al. 2017c; Jia et al. 2019; Daw et al. 2020). However, to the best of our knowledge, PGNN formulations have not been explored yet for our target applications of solving eigenvalue equations in the field of quantum mechanics and electromagnetic propagation. Existing work in PGNN can be broadly divided into two categories. The first category involves label-free learning by only minimizing PG loss without using any labeled data. For example, Physics-informed neural networks (PINNs) and its variants (Raissi, Perdikaris, and Karniadakis 2019, 2017a,b) have been recently developed to solve PDEs by solely minimizing PG loss functions, for simple canonical problems such as Burger’s equation. Since these methods are label-free, they do not explore the interplay between PG loss and label loss. We consider an analogue of PINN for our target application as a baseline in our experiments.

The second category of methods incorporate PG loss as additional terms in the objective function along with label loss, using constant trade-off hyper-parameters. This includes work in basic Physics-guided Neural Networks (PGNNs) (Karpatne et al. 2017c; Jia et al. 2019) for the target application of lake temperature modeling. We use an analogue of this basic PGNN as a baseline in our experiments.

While some recent works have investigated the effects of PG loss on generalization performance (Shin, Darbon, and Karniadakis 2020) and the importance of normalizing the scale of hyper-parameters corresponding to PG loss terms (Wang, Teng, and Perdikaris 2020), they do not study the effects of competing physics losses which is the focus of this paper. Our work is related to the field of multi-task learning (MTL) (Caruana 1993), as the minimization of physics losses and label loss can be viewed as multiple shared tasks. For example, alternating minimization techniques in MTL (Kang, Grauman, and Sha 2011) in MTL can be used to alternate between minimizing different PG loss and label loss terms over different mini-batches. We consider this as a baseline approach in our experiments.

3 Methodology

3.1 Problem statement:

From an ML perspective, we are given a collection of training pairs, $\mathcal{D}_{Tr} := \{\hat{A}_i, (\mathbf{y}_i, b_i)\}_{i=1}^N$, where (\mathbf{y}_i, b_i) is generated by diagonalization solvers. We consider the problem of learning an ANN model, $(\hat{\mathbf{y}}, \hat{b}) = f_{NN}(\hat{A}, \theta)$, that can predict (\mathbf{y}, b) for any input matrix, \hat{A} , where θ are the learnable parameters of ANN. We are also given a set of unlabeled examples, $\mathcal{D}_U := \{\hat{A}_i\}_{i=1}^M$, which will be used for testing. We consider a simple feed-forward architecture of f_{NN} in all our formulations.

3.2 Designing physics-guided loss functions:

A naïve approach for learning f_{NN} is to minimize the mean sum of squared errors (MSE) of predictions on the training set, referred to as the Train-MSE. However, instead of solely relying on Train-MSE, we consider the following PG loss terms to guide the learning of f_{NN} to generalizable solutions:

Characteristic Loss: A fundamental equation we want to satisfy in our predictions, $(\hat{\mathbf{y}}, \hat{b})$, for any input \hat{A} is the eigenvalue equation, $\hat{A}\hat{\mathbf{y}} = \hat{b}\hat{\mathbf{y}}$. Hence, we consider minimizing the following equation:

$$C\text{-Loss}(\theta) := \sum_i \frac{\|\hat{A}_i\hat{\mathbf{y}}_i - \hat{b}_i\hat{\mathbf{y}}_i\|^2}{\hat{\mathbf{y}}^\top \hat{\mathbf{y}}}, \quad (1)$$

where the denominator term ensures that $\hat{\mathbf{y}}$ resides on a unit hyper-sphere with $\|\hat{\mathbf{y}}\| = 1$, thus avoiding scaling issues. Note that by construction, $C\text{-Loss}$ only depends on the predictions of f_{NN} and does not rely on true labels, (\mathbf{y}, b) . Hence, $C\text{-Loss}$ can be evaluated even on the unlabeled test data, \mathcal{D}_U .

Spectrum Loss: Note that there are many non-interesting solutions of $\hat{A}\hat{\mathbf{y}} = \hat{b}\hat{\mathbf{y}}$ that can appear as “local minima” in the optimization landscape of C -Loss. For example, for every input $\hat{A}_i \in \mathcal{D}_U$, there are d possible eigen-solutions (where d is the length of $\hat{\mathbf{y}}$), each of which will result in a perfectly low value of C -Loss = 0, thus acting as a local minima. However, we are only interested in a specific eigenvalue—usually the smallest or the largest—for every \hat{A}_i . Therefore, we consider minimizing another PG loss term that ensures the predicted \hat{b} at every sample is the desired one. In the case of the quantum mechanics application, we use the following loss to find the smallest eigen-solution:

$$S\text{-Loss}(\theta) := \sum_i \exp(\hat{b}_i) \quad (2)$$

The use of exp function ensures that E-Loss is always positive, even when predicted eigenvalues are negative (which is the case for all energy states, especially the ground-state). As for the electromagnetic propagation application, we simply direct the optimization towards the largest eigenvalue by replacing \hat{b}_i with $-\text{Re}(\hat{b}_i)$, where Re extracts the real part of the complex eigenvalue. Since in both cases, the exp function is being applied over negative quantities, S -Loss has smoothly varying gradients.

3.3 Adaptive tuning of PG loss weights:

A simple strategy for incorporating PG loss terms in the learning objective of f_{NN} is to add them to Train-MSE using trade-off weight parameters, λ_C and λ_S , for C -Loss and S -Loss, respectively. Conventionally, such trade-off weights are kept constant to a certain value across all epochs of gradient descent. This inherently assumes that the importance of PG loss terms in guiding the learning of f_{NN} towards a generalizable solution is constant across all stages (or epochs) of gradient descent, and they are in agreement with each other. However, in practice, we empirically find that C -Loss, S -Loss, and Train-MSE compete with each other and have varying importance at different stages (or epochs) of ANN learning. Hence, we consider the following ways of adaptively tuning the trade-off weights of C -Loss and S -Loss, λ_C and λ_S as a function of the epoch number t .

Annealing λ_S : The first observation we make is that S -Loss plays a critical role in the initial stages of learning. Having a large value of λ_S in the beginning few epochs is thus helpful to avoid the selection of local minima and instead converge towards a generalizable solution. Hence, we consider performing a simulated annealing of λ_S that takes on a high value in the beginning epochs, that slowly decays to 0 after sufficiently many epochs. Specifically, we consider the following annealing procedure for λ_S :

$$\lambda_S(t) = \lambda_{S0} \times (1 - \alpha_S)^{\lfloor t/T \rfloor}, \quad (3)$$

where, λ_{S0} is a hyper-parameter denoting the starting value of λ_S at epoch 0, $\alpha_S < 1$ is a hyper-parameter that controls the rate of annealing, and T is a scaling hyper-parameter.

Cold Starting λ_C : The second observation we make is on the effect of C -Loss on the convergence of gradient descent towards a generalizable solution. Note that C -Loss suffers from a large number of local minima and hence is susceptible to favoring the learning of non-generalizable solutions. Hence, in the beginning epochs, it is important to keep C -Loss turned off. Once we have crossed a sufficient number of epochs and have already zoomed into a region in the parameter space in close vicinity to a generalizable solution, we can safely turn on C -Loss so that it can help refine θ to converge to the generalizable solution. Essentially, we “cold starting” λ_C as given by the following procedure:

$$\lambda_C(t) = \lambda_{C0} \times \text{sigmoid}(\alpha_C \times (t - T_a)), \quad (4)$$

where, λ_{C0} is a hyper-parameter denoting the constant value of λ_C after a sufficient number of epochs, α_C is a hyper-parameter that dictates the rate of growth of the sigmoid function, and T_a is a hyper-parameter that controls the cut-off number of epochs after which λ_C is activated from a cold start of 0.

Overall Learning Objective: Combining all of the innovations described above in designing and incorporating PG loss functions, we consider the following overall learning objective:

$$E(t) = \text{Train-Loss} + \lambda_C(t) C\text{-Loss} + \lambda_S(t) S\text{-Loss}$$

Note that Train-Loss is only computed over \mathcal{D}_{Tr} , whereas the PG loss terms, C -Loss and S -Loss, are computed over \mathcal{D}_{Tr} as well as the set of unlabeled samples, \mathcal{D}_U . We refer to our proposed model trained using the above learning objective as *CoPhy*-PGNN, which is an abbreviation for *Competing Physics* PGNN.

4 Evaluation setup

Data in Quantum Physics: We considered $n = 4$ spin systems of Ising chain models for predicting their ground-state wave-function under varying influences of two controlling parameters: B_x and B_z , which represent the strength of external magnetic field along the X axis (parallel to the direction of Ising chain), and Z axis (perpendicular to the direction of the Ising chain), respectively. The Hamiltonian matrix \mathbf{H} for these systems is then given as:

$$\mathbf{H} = - \sum_{i=0}^{n-1} \sigma_i^z \sigma_{i+1}^z - B_x \sum_{i=0}^{n-1} \sigma_i^x - B_z \sum_{i=0}^{n-1} \sigma_i^z, \quad (5)$$

where $\sigma^{x,y,z}$ are Pauli operators and ring boundary conditions are imposed. Note that the size of \mathbf{H} is $d = 2^n = 16$. We set B_z to be equal to 0.01 to break the ground state degeneracy, while B_x was sampled from a uniform distribution from the interval $[0, 2]$.

Note that when $B_x < 1$, the system is said to be in a ferromagnetic phase, since all the spins prefer to either point upward or downward collectively. However, when $B_x > 1$, the system transitions to paramagnetic phase, where both upward and downward spins are equally possible. Because the ground-state wave-function behaves differently in the two

regions, the system actually exhibits different physical properties. Hence, in order to test for the generalizability of ANN models when training and test distributions are different, we generate training data only from the region deep inside the ferromagnetic phase for $B_x < 0.5$, while the test data is generated from a much wider range $0 < B_x < 2$, covering both ferromagnetic and paramagnetic phases. In particular, the training set comprises of $N = 100,000$ points with B_x uniformly sampled from 0 to 0.5, while the test set comprises of $M = 20,000$ points with B_x uniformly sampled from 0 to 2. For validation, we used sub-sampling on the training set to obtain a validation set of 2000 samples. We performed 10 random runs of uniform sampling over N , to show the mean and variance of the performance metrics of comparative ANN models, where at every run, a different random initialization of the ANN models is also used. Unless otherwise stated, the results in any experiment are presented over training size $N = 2000$.

Data in Electromagnetic Propagation: We considered a periodically stratified layer stack of 10 layers of equal length per period. The refractive index n of each layer was randomly assigned an integer value between 1 and 4. Hence, the permittivity $\epsilon = n^2$ can take values from $\{1, 4, 9, 16\}$. Note that the majority of eigenvalue solvers rely on iterative algorithms and are therefore not easily deployable in GPU environments. To demonstrate the scalability of our approach we generate $N = 2000$ realizations of the layered structure. For each example, we also generate the associated \hat{A} of size 401×401 complex values, making the scale of this problem about 2500 times larger than that of the quantum mechanics problem. The combination of the challenging scale of this eigen-decomposition and the scarcity of training data makes this problem interesting from scalability and generalizability perspective. To demonstrate extrapolation ability, we take a training size $|\mathcal{D}_{Tr}| = 370$ realizations that has a refractive index of only 1 in its first layer. On the other hand, we take a test set of size $|\mathcal{D}_U| = 1630$ with the first layer’s refractive index unconstrained (i.e. any value from the set $\{1,2,3,4\}$).

Baseline Methods: Since there does not exist any related work in PGNN that has been explored for our target applications, we construct analogue versions of PINN-*analogue* (Raissi, Perdikaris, and Karniadakis 2019) and PGNN-*analogue* (Karpatne et al. 2017c) adapted to our problem using their major features. We describe these baselines along with others in the following:

1. Black-box NN (or NN): This refers to the “black-box” ANN model trained just using Train-Loss without any PG loss terms.
2. PGNN-*analogue*: The analogue version of PGNN (Karpatne et al. 2017c) for our problem where the hyper-parameters corresponding to S -Loss and C -Loss are set to a constant value.
3. PINN-*analogue*: The analogue version of PINN (Raissi, Perdikaris, and Karniadakis 2019) for our problem that performs label-free learning only using PG loss terms

| Models | MSE ($\times 10^2$) | Cosine Similarity |
|--|-----------------------------------|--------------------------------------|
| <i>CoPhy</i>-PGNN (proposed) | 0.35 ± 0.12 | $99.50 \pm 0.12\%$ |
| Black-box NN | 1.06 ± 0.16 | $95.32 \pm 0.58\%$ |
| PINN- <i>analogue</i> | 6.27 ± 6.94 | $87.37 \pm 12.87\%$ |
| PGNN- <i>analogue</i> | 0.91 ± 1.90 | $97.97 \pm 4.89\%$ |
| MTL-PGNN | 6.33 ± 2.69 | $84.26 \pm 6.33\%$ |
| <i>CoPhy</i> -PGNN (only- \mathcal{D}_{Tr}) | 1.82 ± 0.36 | $93.61 \pm 0.91\%$ |
| <i>CoPhy</i> -PGNN (w/o S -Loss) | 10.97 ± 0.71 | $76.27 \pm 0.80\%$ |
| <i>CoPhy</i> -PGNN (Label-free) | 9.97 ± 4.42 | $63.97 \pm 16.20\%$ |

Table 1: Test-MSE and Cosine Similarity of comparative ANN models on training size $N = 1000$ on the quantum physics application.

with constant weights. Note that the PG loss terms are not defined as PDEs in our problem.

4. MTL-PGNN: Multi-task Learning (MTL) variant of PGNN where PG loss terms are optimized alternatively (Kang, Grauman, and Sha 2011) by randomly selecting one from all the loss terms for each mini-batch in every epoch.

We also consider the following ablation models:

1. *CoPhy*-PGNN (only- \mathcal{D}_{Tr}): This is an ablation model where the PG loss terms are only trained over the training set, \mathcal{D}_{Tr} . Comparing our results with this model will help in evaluating the importance of using unlabeled samples \mathcal{D}_U in the computation of PG loss.
2. *CoPhy*-PGNN (w/o S -Loss): This is another ablation model where we only consider C -Loss in the learning objective, while discarding S -Loss.
3. *CoPhy*-PGNN (Label-free): This ablation model drops Train-MSE from the learning objective and hence performs label-free (LF) learning only using PG loss terms.

Evaluation Metrics: We use two evaluation metrics: (a) Test MSE, and (b) Cosine Similarity between our predicted eigenvector, $\hat{\mathbf{y}}$, and the ground-truth, \mathbf{y} , averaged across all test samples. We particularly chose the cosine similarity for multiple reasons. First, Euclidean distances are not very meaningful in high-dimensional spaces of wave-functions, such as the ones we are considering in our analyses. Second, an ideal cosine similarity of 1 provides an intuitive baseline to evaluate goodness of results. But most importantly, in the electromagnetic propagation application, it is crucial to compare not just Fourier coefficients of the expansion (which is what the neural net produces) but rather the actual profile of the magnetic field in the real space. The accuracy of this prediction can be tested by calculating the overlap integral between the exact and the predicted profiles. That integral, due to orthogonality of Fourier expansion, reduces to the cosine similarity. This facilitates testing whether our predicted vectors are valid eigenvectors from a physical standpoint.

5 Results and analysis

5.1 Quantum Physics Application:

Table 1 provides a summary of the comparison of *CoPhy*-PGNN with baseline methods on the quantum physics ap-

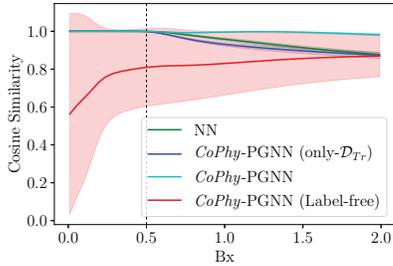


Figure 1: Cosine Similarity on test samples as a function of B_x . The dashed line represents the boundary between the interval used for training (left) and testing (right).

plication. We can see that our proposed model shows significantly better performance in terms of both Test-MSE and Cosine Similarity. In fact, the cosine similarity of our proposed model is almost 1, indicating almost perfect fit with test labels. (Note that even a small drop in cosine similarity can lead to cascading errors in the estimation of other physical properties derived from the ground-state wavefunction.) An interesting observation from Table 1 is that *CoPhy*-PGNN (Label-free) actually performs even worse than black-box NN. This shows that solely relying on PG loss without considering Train-MSE is fraught with challenges in arriving at a generalizable solution. Indeed, using a small number of labeled examples to compute Train-MSE provides a significant nudge to ANN learning to arrive at more accurate solutions. Another interesting observation is that *CoPhy*-PGNN (only- \mathcal{D}_{Tr}) again performs even worse than Black-box NN. This demonstrates that it is important to use unlabeled samples in \mathcal{D}_U , which are representative of the test set, to compute the PG loss. Furthermore, notice that *CoPhy*-PGNN (w/o S -Loss) actually performs worst across all models, possibly due to the highly non-convex nature of C -Loss function that can easily lead to local minima when used without S -Loss. This sheds light on another important aspect of PGNN that is often over-looked, which is that it does not suffice to simply add a PG-Loss term in the objective function in order to achieve generalizable solutions. In fact, an improper use of PG Loss can result in worse performance than a black-box model.

Evaluating generalization power: Instead of computing the average cosine similarity across all test samples, Figure 1 analyzes the trends in cosine similarity over test samples with different values of B_x , for four comparative models. Note that none of these models have observed any labeled data during training outside the interval of $B_x \in [0, 0.5]$. Hence, by testing for the cosine similarity over test samples with $B_x > 0.5$, we are directly testing for the ability of ANN models to generalize outside the data distributions it has been trained upon. Evidently, all label-aware models perform well on the interval of $B_x \in [0, 0.5]$. However, except for *CoPhy*-PGNN, all baseline models degrade significantly outside that interval, proving their lack of generalizability. Moreover, the label-free, *CoPhy*-PGNN (Label-free), model is highly erratic, and performs poorly across the

board.

Analysis of loss landscapes: We visualize the landscape of different loss functions w.r.t. ANN model parameters. In particular, we use the code in (Bernardi 2019) to plot a 2D view of the landscape of different loss functions, namely Train-MSE, Test-MSE, and PG-Loss (sum of C -Loss and S -Loss), in the neighborhood of a model solution, as shown in Figure 2. The model’s parameters are treated with filter normalization as described in (Li et al. 2018), and hence, the coordinate values of the axes are unit-less. Also, the model solutions are represented by blue dots. As can be seen, all label-aware models have found a minimum in Train-MSE landscape. However, when the test-MSE loss surface is plotted, it is clear that while the *CoPhy*-PGNN model is still at a minimum, the other baseline models are not. This is a strong indication that using the PG loss with unlabeled data can lead to better extrapolation; it allows the model to generalize beyond in-distribution data. We can see that without using labels, *CoPhy*-PGNN (Label-free) fails to reach a good minimum of Test-MSE, even though it arrives at a minimum of PG Loss.

5.2 Electromagnetic Propagation Application:

For this application, the size of \hat{A} is 401×401 , making it a daunting task for an eigensolver in terms of computation time. As a result, a grid search hyper-parameter tuning of ANN models is prohibitively expensive. This is due to the large number of epochs needed to optimize a model for a problem of this scale. Nonetheless, we were still able to optimize a model to do fairly well by manually adjusting the hyper-parameters and architecture of *CoPhy*-PGNN to yield acceptable results on the validation set. We emphasize, however, that a more exhaustive tuning could lead to better results that surpass the ones we obtained. Figure 3 shows that *CoPhy*-PGNN is still able to better extrapolate than a Black-box NN on testing scenarios with permittivity greater than 1. In fact, we have observed that as Black-box NN solely optimizes Train-MSE, its cosine similarity measure deteriorates on the test set. This is in contrast to *CoPhy*-PGNN’s ability to maintain a cosine similarity close to 1 even though its validation loss is comparable to Black-box NN’s.

While training our model still takes a significant amount of time (about 12 hours), its effectiveness with respect to testing speed is demonstrated in Table 2. We can see that our approach is at least an order of magnitude faster during testing than any numerical eigensolver. This highlights the promise in using neural networks to solve physics-based eigen-value problems, since, once trained, they can be used to produce eigen-solutions on test points much faster than numerical methods. Further, while *CoPhy*-PGNN shows higher error than numerical solvers, note that the cosine similarity of our model’s predictions with ground-truth is close to 0.8, thus admitting physical usability.

6 Conclusions and future work

This work proposed novel strategies to address the problem of competing physics loss functions in PGNN. For the general problem of solving eigen-value equations, we designed

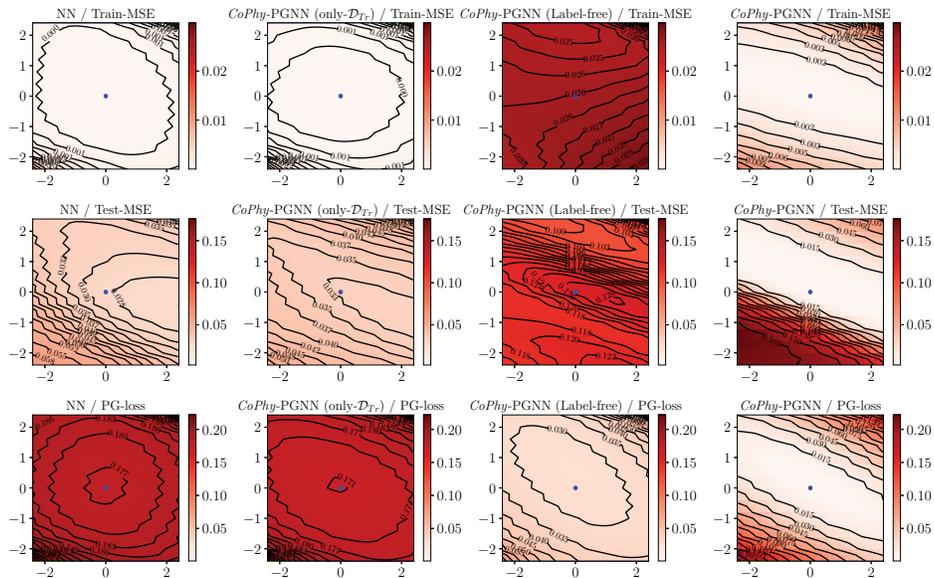


Figure 2: A comprehensive comparison between *CoPhy*-PGNN and different baseline models. The 1st and 2nd columns show that without using unlabeled data, the model does not generalize well. On the other hand, the 3rd column shows that without labeled data, the model fails to reach a good minimum. Only the last column, our proposed model, shows a good fit across both labeled and unlabeled data. The best performing model is also the model that best optimizes the PG loss.

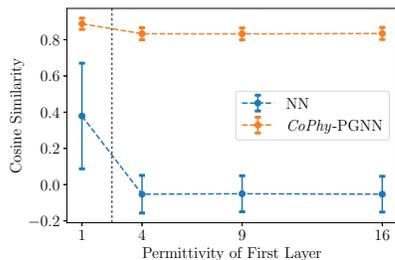


Figure 3: Cosine similarity of *CoPhy*-PGNN compared to Black-box NN for the electromagnetic propagation application. The dashed line represents the boundary between the interval used for training (left) and testing (right).

| Solver | average time (seconds) | average $ Ay - by $ |
|--------------------------|------------------------|-------------------------|
| <i>CoPhy</i> -PGNN | 0.0430 | 1.878×10^2 |
| numpy.linalg.eig | 93.743 | 7.714×10^{-6} |
| Matlab | 0.196 | 8.747×10^{-12} |
| torch.eig | 16.565 | 6.821×10^{-13} |
| scipy.linalg.eig | 106.223 | 7.538×10^{-4} |
| scipy.sparse.linalg.eigs | 8.893 | 4.418×10^{-3} |

Table 2: Comparison of speed and accuracy between *CoPhy*-PGNN and other numerical eigensolvers. Note that Matlab calculates the eigenvalue of interest (i.e. the largest), while the other eigensolvers, except for our proposed method, calculate all the eigenvalues of the given matrix. This explains why Matlab has relatively faster execution time.

a PGNN model *CoPhy*-PGNN and demonstrated its efficacy in two target applications in quantum mechanics and electromagnetic propagation. From our results, we found that: 1) PG loss helps to extrapolate and gives the model better generalizability; and 2) Using labeled data along with PG loss results in more stable PGNN models. Moreover, we visualized the loss landscape to give a better understanding of how the combination of both labeled data loss and PG loss leads to better generalization performance. We have also demonstrated the generalizability of our *CoPhy*-PGNN to multiple application domains with varying types of physics loss functions, as well as its scalability to large systems. Future work can focus on reducing the training time of our model so as to perform extensive hyper-parameter tuning to reach a better global minima. Finally, while this work empirically demonstrated the value of *CoPhy*-PGNN in combating with competing PG loss terms, future work can focus on theoretical analyses of our approach.

References

Appenzeller, T. 2017. The scientists’ apprentice. *Science* 357(6346): 16–17.

Bernardi, M. D. 2019. loss-landscapes. URL <https://github.com/marcellodebernardi/loss-landscapes/>.

Caruana, R. 1993. Multitask Learning: A Knowledge-Based Source of Inductive Bias. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning, ICML’93*, 41–48. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1558603077.

Daw, A.; Thomas, R. Q.; Carey, C. C.; Read, J. S.; Appling, A. P.; and Karpapne, A. 2020. Physics-Guided Architecture

- (PGA) of Neural Networks for Quantifying Uncertainty in Lake Temperature Modeling. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, 532–540. SIAM.
- de Bezenac, E.; Pajot, A.; and Gallinari, P. 2019. Deep learning for physical processes: Incorporating prior scientific knowledge. *Journal of Statistical Mechanics: Theory and Experiment* 2019(12): 124009.
- Graham-Rowe, D.; Goldston, D.; Doctorow, C.; Waldrop, M.; Lynch, C.; Frankel, F.; Reid, R.; Nelson, S.; Howe, D.; and Rhee, S. 2008. Big data: science in the petabyte era. *Nature* 455(7209): 8–9.
- Jia, X.; Willard, J.; Karpatne, A.; Read, J.; Zwart, J.; Steinbach, M.; and Kumar, V. 2019. Physics Guided RNNs for Modeling Dynamical Systems: A Case Study in Simulating Lake Temperature Profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, 558–566. SIAM.
- Kang, Z.; Grauman, K.; and Sha, F. 2011. Learning with Whom to Share in Multi-Task Feature Learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11*, 521–528. Madison, WI, USA: Omnipress. ISBN 9781450306195.
- Karpatne, A.; Atluri, G.; Faghmous, J. H.; Steinbach, M.; Banerjee, A.; Ganguly, A.; Shekhar, S.; Samatova, N.; and Kumar, V. 2017a. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering* 29(10): 2318–2331.
- Karpatne, A.; Atluri, G.; Faghmous, J. H.; Steinbach, M.; Banerjee, A.; Ganguly, A.; Shekhar, S.; Samatova, N.; and Kumar, V. 2017b. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering* 29(10): 2318–2331.
- Karpatne, A.; Watkins, W.; Read, J.; and Kumar, V. 2017c. Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling. *arXiv preprint arXiv:1710.11431*.
- Li, H.; Xu, Z.; Taylor, G.; Studer, C.; and Goldstein, T. 2018. Visualizing the Loss Landscape of Neural Nets. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 6389–6399. Curran Associates, Inc. URL <http://papers.nips.cc/paper/7875-visualizing-the-loss-landscape-of-neural-nets.pdf>.
- Raissi, M.; Perdikaris, P.; and Karniadakis, G. 2017a. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. *arXiv preprint arXiv:1711.10561*.
- Raissi, M.; Perdikaris, P.; and Karniadakis, G. E. 2017b. Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations. *arXiv preprint arXiv:1711.10566*.
- Raissi, M.; Perdikaris, P.; and Karniadakis, G. E. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378: 686–707.
- Shin, Y.; Darbon, J.; and Karniadakis, G. E. 2020. On the convergence and generalization of physics informed neural networks. *arXiv preprint arXiv:2004.01806*.
- Stewart, R.; and Ermon, S. 2017. Label-free supervision of neural networks with physics and domain knowledge. In *AAAI*.
- Wang, J.-X.; Wu, J.; Ling, J.; Iaccarino, G.; and Xiao, H. 2017. A Comprehensive Physics-Informed Machine Learning Framework for Predictive Turbulence Modeling. *arXiv preprint arXiv:1701.07102*.
- Wang, J.-X.; Wu, J.-L.; and Xiao, H. 2016. Physics-Informed Machine Learning for Predictive Turbulence Modeling: Using Data to Improve RANS Modeled Reynolds Stresses. *arXiv preprint arXiv:1606.07987*.
- Wang, J.-X.; Wu, J.-L.; and Xiao, H. 2017. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Physical Review Fluids* 2(3): 034603.
- Wang, S.; Teng, Y.; and Perdikaris, P. 2020. Understanding and mitigating gradient pathologies in physics-informed neural networks. *arXiv preprint arXiv:2001.04536*.
- Willard, J.; Jia, X.; Xu, S.; Steinbach, M.; and Kumar, V. 2020. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*.