Deep Learning Solutions of Schrödinger Equation and Kronig-Penney Model

Oğuzhan Güngördü
Electrical-Electronics Eng.& Physics
Koc University
ogungordu18@ku.edu.tr

I. Introduction

Neural networks, a class of machine learning models inspired by the structure and function of the human brain, have found significant applications in the field of physics. These artificial neural networks are composed of interconnected nodes, or neurons, that process and transmit information. In physics, neural networks are employed to solve complex problems, make predictions, and approximate solutions to intricate equations. Their ability to learn and generalize from large datasets enables them to capture underlying patterns and relationships in physical systems. Neural networks have been successfully applied in various areas of physics, including quantum mechanics, condensed matter physics, astrophysics, and particle physics, among others [1]. They have proven to be valuable tools for tasks such as image and signal processing, data analysis, pattern recognition, and modelling complex physical phenomena. By leveraging the power of neural networks, physicists can gain deeper insights, accelerate computations, and unlock new avenues for discovery in the realm of physics.

In the realm of quantum mechanics, the accurate solution of mathematical models plays a pivotal role in understanding the behaviour of particles and their interactions. This work delves into two distinct aspects of quantum systems, encompassing both analytical and computational approaches. The first part focuses on solving the 2D Schrödinger equation for infinite wells (IWs) and simple harmonic oscillators (SHOs), employing well-established deep learning technique to derive precise solutions for these fundamental systems. These solutions provide valuable insights into the behaviour and ground state energy of particles confined within these potentials. The second part of this work takes an approach by utilizing deep learning methodologies to tackle the solution of the Kronig-Penney model. By leveraging the power of neural networks, this novel approach aims to overcome the limitations of traditional numerical methods and approximate the energy bands for periodic potentials. This exploration of deep learning techniques not only expands our understanding of the Kronig-Penney model but also highlights the potential of nonlinear processing and pattern recognition capabilities of neural networks in solving intricate quantum mechanical problems. By combining analytical solutions with cutting-edge computational techniques, this work offers a comprehensive analysis of quantum systems and demonstrates the potential of deep learning in advancing our understanding of quantum phenomena for complicated many-body systems.

II. Schrodinger Equation

With access to precise solutions of the Schrödinger equation for one-electron systems, most chemistry principles could be derived from fundamental principles. However, obtaining exact wavefunctions for complex chemical systems is difficult due to their NP-hard computational nature. Nevertheless, approximations can be achieved through polynomial-scaling algorithms. The main challenge faced by these algorithms lies in selecting a wavefunction approximation, known as an Ansatz, which involves a trade-off between efficiency and accuracy. Neural networks have demonstrated impressive capability as accurate and practical function approximators and hold promise as a concise Ansatz for spin systems' wavefunctions. However, problems in electronic structure demand wavefunctions that adhere to Fermi-Dirac statistics [2]. The results obtained through deep learning methods demonstrate superior performance compared to traditional numerical approaches, particularly when using lower grid numbers. This highlights the potential of deep learning as a powerful tool for accelerating the computation of quantum mechanical systems while maintaining high levels of accuracy. By harnessing the capabilities of neural networks, this work aims to bridge the gap between analytical and numerical solutions, providing an alternative avenue that combines the best of both worlds.

A. *Analytical and Numerical Solutions*

$$\hat{H}\psi \equiv (\hat{T} + \hat{V})\psi = \varepsilon\psi \qquad (1)$$

For each potential V (IW and SHO potentials) utilized in this study were specifically generated with a suitable dynamic range and length scale that would yield ground-state energies falling within a physically relevant range. These potentials were represented on a square domain ranging from -20 to 20 atomic units (a.u.) and discretized on a 64x64 grid. As the potential for the SHOs and IWs possess an analytical solution, we employed this solution as a reference to validate the accuracy of our solver. Note that analytical solutions are labelled as ground-truth for the training process either. Numerical solution of Schrödinger eqn. (1) is implemented by [3] which calculates finite differences using vertical, horizontal, and crosswise spatial finite differences to approximate spatial derivatives. Analytical solutions of the (1) for the ground-state energies can be seen in the following equations for IW and SHO respectively. Note that k_x and k_y represent curvature of the potential on two primary axes, while L_x and L_y represent width of the well in two dimensions.

$$\varepsilon_0 = \frac{1}{2}\pi^2\hbar^2(L_x^{-2} + L_y^{-2})$$

$$\varepsilon_0 = \frac{\hbar}{2}(\sqrt{k_x} + \sqrt{k_y})$$

B. *Generating Dataset and Training Details*

Since potentials are defined using variables of k_x, k_y, L_x and L_y, these parameters are generated using random function of NumPy to generate training and validation dataset. Training datasets consisted of 100,000 training examples and training was run for 100 epochs with batch size 1000 on T4 GPU core. Note that c_x and c_y are the center coordinates of the potentials and they are generated uniformly without any problem because they do not affect the ground state energies. Since the ground state energies of infinite well (IW) potentials depend on the inverse square parameters, generating these variables uniformly may not result in a uniform distribution of ground state energies. This nonuniform distribution can introduce bias in the training procedure when using a dataset with nonuniform energy distribution because randomly generating $L_x$and L_y independently leads to a distribution of energies highly biased toward low energy values. To address the issue of nonuniform distribution in ground state energies caused by the inverse square IW potential, one approach is to generate the ground state energies and one of the parameters uniformly. Then, the other parameter can be calculated using the following formulas to achieve a more uniform distribution of energies:

$$L_y = 1/\sqrt{\frac{2E}{\pi^2} - \frac{1}{L_x^2}}$$

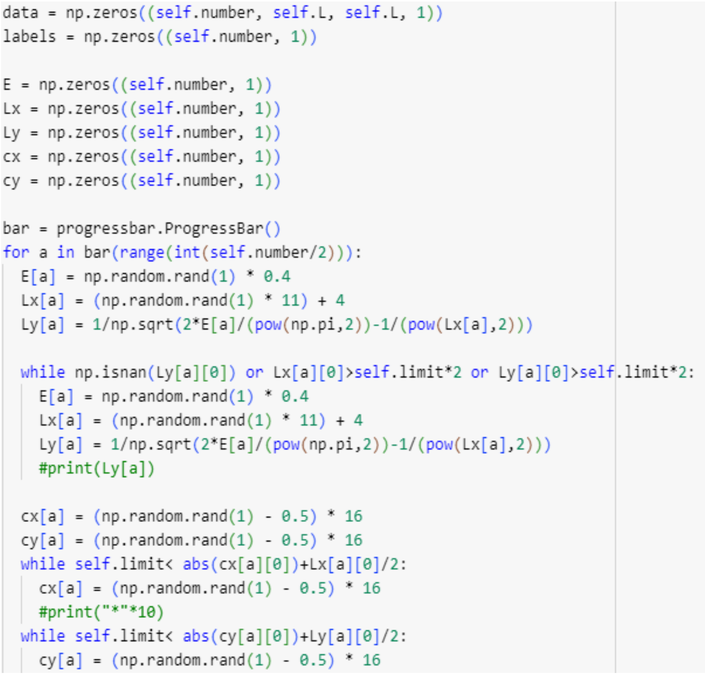
Overall, we generate k_x, k_y, c_x and c_y uniformly for SHO, and generate potential map according to the following formula and bounds:

$$V(x,y) = \frac{1}{2}\left(k_x(x - c_x)^2 + k_y(y - c_y)^2\right)$$

The potential is truncated at 20.0 Ha, (i.e., if V > 20, V = 20)

Parameter		Lower bound	Upper bound
k_x	spring constant	0.0	0.16
k_y	spring constant	0.0	0.16
c_x	center position	-8.0	8.0
c_y	center position	-8.0	8.0

On the other hand, we generate only c_x and c_y uniformly for IW potentials. To ensure a more uniform distribution of ground state energies, the following procedure is implemented. First, the energy E is randomly generated uniformly within the interval 0 to 0.4 Hartree. Then, the parameter L_x, representing the width of the well, is randomly generated within the interval 4.0 to 15.0. To determine the value of L_y, which will produce the desired energy E given L, upper formula is employed. Not all combinations of $L_x$and E yield valid solutions for L_y, so attempts are made until a valid solution is found. To prevent one dimension of the well from consistently being larger, the values of L_x and $L_y$are swapped with a 50% probability. This process ensures a relatively even distribution of energy by allowing for the exploration of different combinations of L_x and L_y, resulting in a more balanced dataset distribution. Note that not all combinations of c_x, c_y, L_x and L_y are located within the -20 a.u. to 20 a.u. limits. Therefore, we implement a space limit check to control nonzero potential is in the finite space by implementing the following algorithm:



```
data = np.zeros((self.number, self.L, self.L, 1))
labels = np.zeros((self.number, 1))

E = np.zeros((self.number, 1))
Lx = np.zeros((self.number, 1))
Ly = np.zeros((self.number, 1))
cx = np.zeros((self.number, 1))
cy = np.zeros((self.number, 1))

bar = progressbar.ProgressBar()
for a in bar(range(int(self.number/2))):
  E[a] = np.random.rand(1) * 0.4
  Lx[a] = (np.random.rand(1) * 11) + 4
  Ly[a] = 1/np.sqrt(2*E[a]/(pow(np.pi,2))-1/(pow(Lx[a],2)))

  while np.isnan(Ly[a][0]) or Lx[a][0]>self.limit*2 or Ly[a][0]>self.limit*2:
    E[a] = np.random.rand(1) * 0.4
    Lx[a] = (np.random.rand(1) * 11) + 4
    Ly[a] = 1/np.sqrt(2*E[a]/(pow(np.pi,2))-1/(pow(Lx[a],2)))
    #print(Ly[a])

  cx[a] = (np.random.rand(1) - 0.5) * 16
  cy[a] = (np.random.rand(1) - 0.5) * 16
  while self.limit< abs(cx[a][0])+Lx[a][0]/2:
    cx[a] = (np.random.rand(1) - 0.5) * 16
    #print("*"*10)
  while self.limit< abs(cy[a][0])+Ly[a][0]/2:
    cy[a] = (np.random.rand(1) - 0.5) * 16
```

Fig. 1 Boundary limit check algorithm for IW potentials

After implementing the formula that we define for 2D potentials, an example of IW and SHO potentials can be seen in Figure 2.

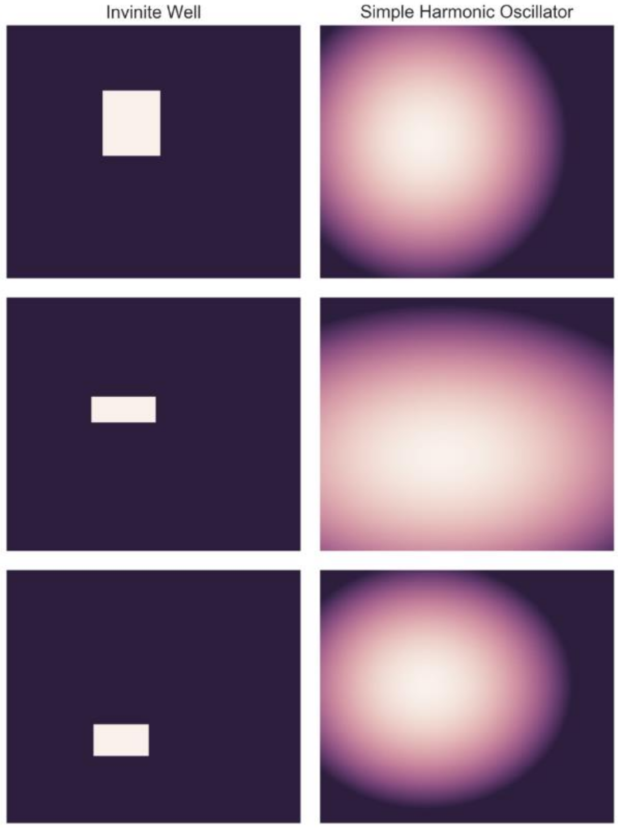


Fig. 2 Example of IW and SHO potentials on grid

C. Model Architecture

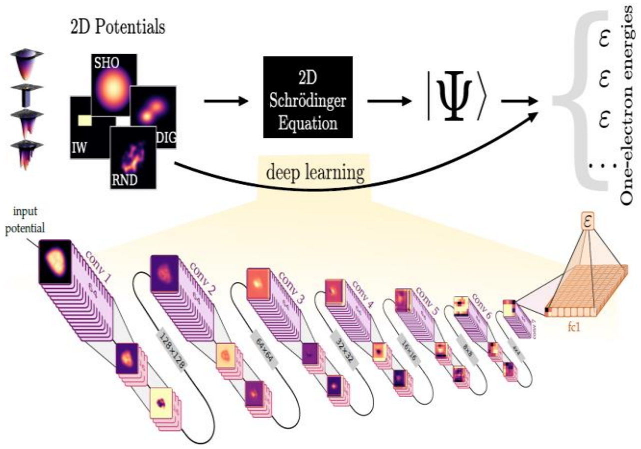


Fig. 3 Network Architecture that solves energy eigenstates.

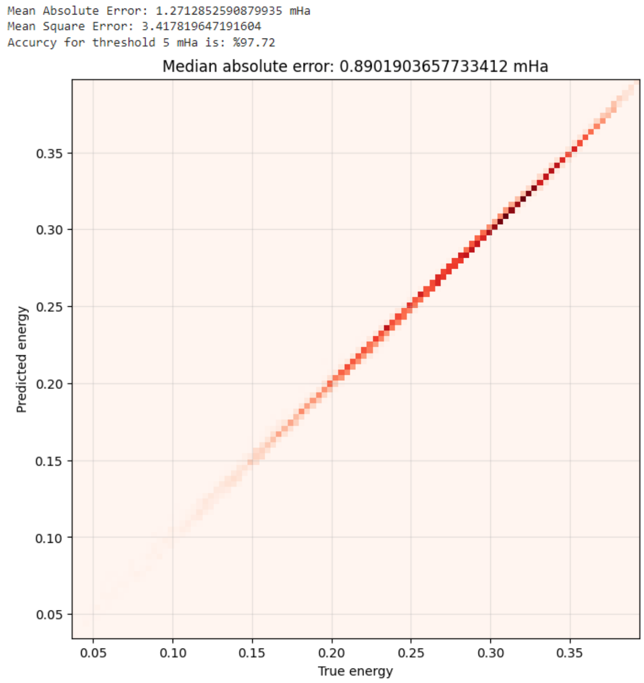
As depicted in Figure 3, the neural network directly predicts the ground-state energy based on the 2D potential. In other words, the input to the network is a 2D potential represented as a single-channel image, and the output is the ground-state energy associated with that specific 2D potential.

Our model incorporates two types of convolutional layers referred to as "reducing" and "non-reducing". The reducing layers consist of seven convolutions with filter sizes of 3x3 pixels. Each reducing layer employs 64 filters and a stride of 2x2, effectively decreasing the image resolution by a factor of two at each step. Between each pair of reducing convolutional layers, we have inserted two convolutional layers (a total of 12) with 16 filters of size 4x4. These filters operate with a unit stride, preserving the image resolution. The purpose of these additional layers is to introduce more trainable parameters into the network. All convolutional layers employ the rectified linear unit (ReLU) activation function.

The output of the final convolutional layer is passed to a fully-connected layer with a width of 1024, also utilizing ReLU activation. This layer, in turn, connects to a final fully-connected layer with a single output. This output represents the prediction of our deep neural network (DNN). It is used to calculate the mean-squared error (L2 loss) between the true label and the predicted label, serving as the loss function in our model.

D. Results

The performance of the simple harmonic oscillator, which is known for its simplicity, was exceptionally impressive which can be seen in Figure 4. The trained model demonstrated remarkable accuracy in predicting the ground state energies, achieving a median absolute error (MedAE) of 0.89 mHa, mean absolute error (MAE) of 1.27 mHa, and accuracy of 97.72% for ±5 mHa.

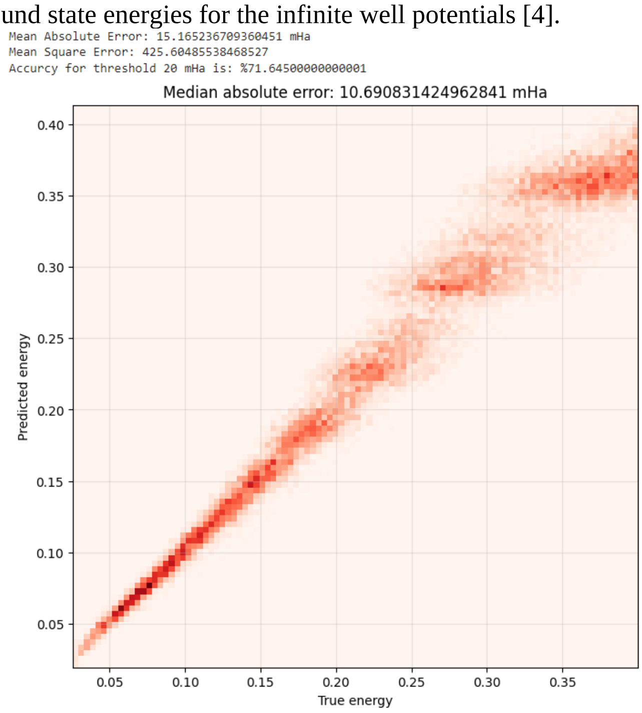


```
Mean Absolute Error: 1.2712852590879935 mHa
Mean Square Error: 3.417819647191604
Accurcy for threshold 5 mHa is: %97.72
```

Fig. 4 Histograms of true vs. predicted energies for SHO

On the other hand, the performance of the infinite well potentials were moderate in Figure 5, with a MedAE of 10.69

mHa, MAE of 15.17 mHa, and accuracy of 71.65% for ±20 mHa. This result is notably lower than the performance achieved with the simple harmonic oscillator potentials, despite both types of potentials being analytically dependent on two simple parameters. The relatively poorer performance can be attributed to the sharp discontinuity associated with the infinite well potentials, as well as the sparse information present in the binary-valued potentials. These factors likely contribute to the challenges faced in accurately predicting the ground state energies for the infinite well potentials [4].



```
Mean Absolute Error: 15.165236709360451 mHa
Mean Square Error: 425.60485538468527
Accurcy for threshold 20 mHa is: %71.6450000000001
```

Fig. 5 Histograms of true vs. predicted energies for IW

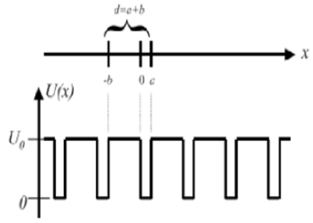
Note that average elapsed time to predict ground state energy of the 64x64 grid space potential is 1.95 seconds for numerical solution, while average elapsed time for deep learning architecture is 0.23 seconds with higher accuracy. While increasing the grid spaces improves accuracy by providing better approximation of space derivatives through finite differences, it significantly increases memory usage due to the computational costs involved. However, by leveraging the capability of GPUs, the deep learning model can be efficiently executed, resulting in faster inference times and reduced memory usage.

III. Kronig-Penney Model

The solution of the Kronig-Penney model using deep learning techniques represents a significant advancement in understanding and analyzing fundamental models in physics. Traditionally, solving the exact equation of the Kronig-Penney model, which describes the behaviour of electrons in a periodic potential, has proven to be a simplified task due to its complexity of defining the Dirac delta function. However, with the advent of deep learning, a new avenue has opened for tackling such problems. Deep learning algorithms, particularly neural networks, have demonstrated remarkable capabilities in approximating complex functions and patterns without solving them directly. By leveraging these powerful tools, researchers can now explore alternative methods to solve the Kronig-Penney model or any physics models by employing nonlinear processing and pattern recognition capabilities of neural networks. This approach offers the potential to overcome the limitations of traditional analytical methods and provides a fresh perspective on solving fundamental models in physics. Moreover, the use of deep learning in this context not only allows for efficient and accurate solutions but also sheds light on the broader concept of utilizing nonlinear processing in neural networks to tackle other challenging problems in the field of physics.

A. Numerical Solution

The numerical solution implemented in [5] is used to analyze the energy levels predicted by the Kronig-Penney model of electron band structure for a periodic potential. The result of [5] are plots of the E-K bands of the periodic potential. Periodic potential is approximated with square function to get an analytical solution. To define repeating unit cells, system variables are defined by width of the well (a), width of the spacing in between the potential wells (b), depth of potential energy (U_0), and maximum relative energy over x-axis $(\frac{E}{U_0} = \zeta)$. After solving the Schrödinger's equation using the Bloch Theorem [6] which is the periodic potential for boundary conditions, we get a two separate equality for $0 < \zeta < 1$ and $\zeta > 1$ as in Figure 6.



The curves are obtained by solving the following equation:

$$\cos k(a+b) = \frac{1-2\zeta}{2\sqrt{\zeta(1-\zeta)}}\sin\left(\alpha_0 a\sqrt{\zeta}\right)\sinh\left(\alpha_0 b\sqrt{1-\zeta}\right) + \cos\left(\alpha_0 a\sqrt{\zeta}\right)\cosh\left(\alpha_0 b\sqrt{1-\zeta}\right)$$

$$for\, 0 < \zeta < 1$$

$$\cos k(a+b) = \frac{1-2\zeta}{2\sqrt{\zeta(\zeta-1)}}\sin\left(\alpha_0 a\sqrt{\zeta}\right)\sin\left(\alpha_0 b\sqrt{\zeta-1}\right) + \cos\left(\alpha_0 a\sqrt{\zeta}\right)\cos\left(\alpha_0 b\sqrt{\zeta-1}\right)$$

$$for\, 1 < \zeta$$

Where,

$$\alpha_0 = \sqrt{\frac{2mU_0}{h^2}}$$

$$\zeta = \frac{E}{U_0}$$

Fig. 6 Solution of Kronig-Penney Model for periodic potentials

The left-hand side (LHS) of the equation is a function of k, while right-hand side (RHS) is function of only energy. To reconstruct E-K diagram, we should solve the function from the RHS by using the equality where LHS is in the range of -1 and 1 due to oscillation. Note that range of variable ζ is defined within the range of 0 to specified maximum relative energy using 20.000 samples. That is, overlapped regions are reconstructed to calculate the energy band gaps. Therefore, Kronig-Penney model is solved using two steps. First, LHS and RHS of the solution of the Schrödinger's equation will be calculated. Then, valid solutions are reconstructed to get the E-K energy bands. Figure 7 shows example the solution of the Kronig-Penney model for system variables $\mathrm{a} = \mathrm{b} = 2.7 \text{ Angstroms}, \mathrm{U}_0 = 5 \text{ eV}, \text{and } \zeta = 6.$

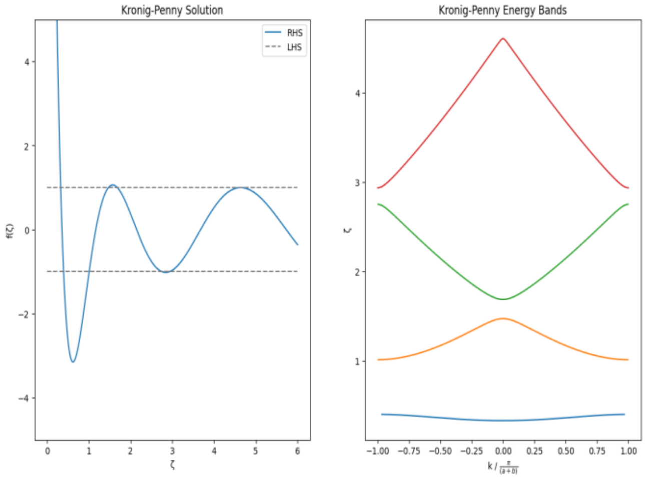


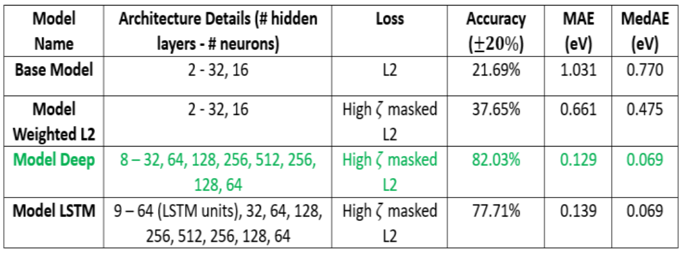
Fig. 7 Example of Numerical Solution for Kronig-Penney Model

B. Ablation Study and Results

In the ablation study, we investigate the impact of various model architectures and loss functions on the performance of our network in predicting the right-hand side (RHS) of the Kronig-Penney solution. The Kronig-Penney solution consists of 20,000 data points at the output, while our network takes 4 system variables as an input. However, we deliberately exclude the first 100 points from the predicted output. These initial points often contain large values that can result in infinities and pose challenges for normalization. Furthermore, these points do not contribute significantly to the oscillatory behaviour and information related to energy bands present in the remaining part of the solution. By excluding these points, we ensure that the network focuses on capturing the essential information and provides more accurate predictions for the main portion of the solution which are the last 19,900 points. Therefore, our network takes 4 system variables as an input, and predicts the last 19,900 points of the RHS of the Kronig-Penney model using neurons of the neural network. After prediction of the RHS, we reconstruct the energy bandgaps using the appropriate regions that gives the solution as always. Note that performance of models is evaluated by comparing accuracy, mean absolute error (MAE), and median absolute error (MedAE). For evaluating different models, we examine the solution of the RHS of the Kronig-Penney, as well as the predicted energy bands and energy gaps for a specific example system. By comparing the performance of various models, we can assess their effectiveness in capturing the behaviour of the RHS and accurately predicting the energy bands and gaps. This analysis allows us to gain insights into the strengths and limitations of each model and identify the most suitable approach for accurately characterizing the system under investigation. Note that the energy band gap is determined by calculating the difference between the minimum value of the third band (conduction band) and the maximum value of the second band (valence band). This calculation provides a measure of the energy separation between these two important bands, which is a crucial characteristic of the material system under consideration. By accurately predicting the energy bands and their respective extrema, we can derive insights into the nature of electronic states and the behaviour of electrons within the material. The training set consists of randomly generated parameter values within specific ranges. The a and b variables range from 1.0 to 4.0 Angstrom, while U_0 ranges from 2.0 to 8.0 eV. Additionally, the ζ parameter falls between 5.0 and 6.5. This diverse range of parameter values ensures that the model learns from a wide variety of inputs during training. Note that the training procedure for all architectures involved using a dataset of 100,000 samples, with a batch size of 32, and training for a total of 1000 epochs. Also, a learning rate scheduler is implemented to reduce the learning rate by a factor of 0.5 if there is no improvement in validation loss for 30 epochs, with a minimum learning rate of 1e-7.

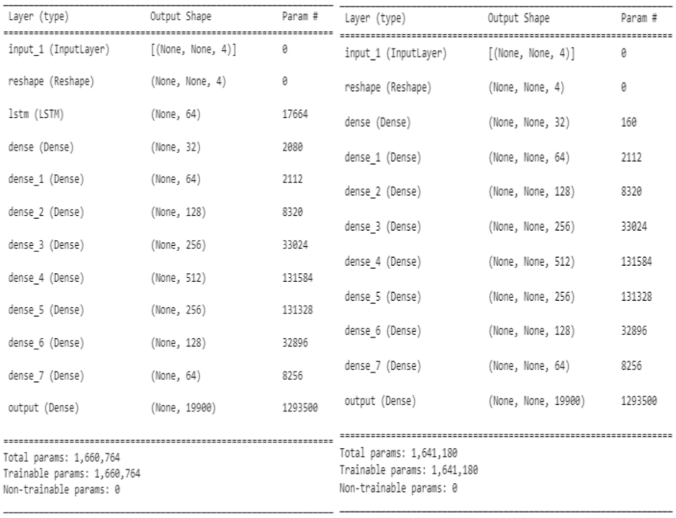
In our ablation study, we explore the performance of four different architectures (details can be seen in Figure 8) for our task on the evaluation dataset which is different than the training dataset. Initially, "Base Model" exhibits poor performance with an accuracy of only 21.69%, primarily due to its inability to predict higher relative energy values effectively. To address this limitation, we introduce a modified loss function called the "high ζ masked L2", which assigns more importance to high relative energy values. Training "Model Weighted L2" with this adjusted loss function leads to an improved accuracy of 37.65%, although it remains relatively modest. To better capture the oscillatory patterns inherent in our task, we incorporate a deeper architecture in "Model Deep". This architecture consists of 8 hidden layers structured in a decoder and encoder fashion. The increased depth allows the model to learn oscillations more effectively, resulting in a significantly improved accuracy of 82.03%.

Recognizing the oscillatory nature of the task and the sequential dependency of the data, we introduce Long Short-Term Memory (LSTM) units into "Model LSTM". LSTM units are specifically designed for modelling time series data, making them suitable for capturing the complex temporal dependencies present in our oscillatory function. While "Model LSTM" shows comparable performance to "Model Deep", it does not outperform it in terms of accuracy. Note that best results are achieved by using "Model Deep" in terms of accuracy, MAE, and MedAE which can be seen in Figure 8. These findings highlight the importance of architectural choices in achieving better prediction performance. The inclusion of LSTM units, along with the deeper architecture, demonstrates the ability to capture oscillations and temporal dependencies, leading to improved accuracy. Further exploration and fine-tuning of architectural variations may yield even better results in future investigations. Model architectures of "Model Deep" and "Model LSTM" can be seen in Figure 9.



Model Name	Architecture Details (# hidden layers - # neurons)	Loss	Accuracy ($\pm 20\%$)	MAE (eV)	MedAE (eV)
Base Model	2 - 32, 16	L2	21.69%	1.031	0.770
Model Weighted L2	2 - 32, 16	High ζ masked L2	37.65%	0.661	0.475
Model Deep	8 – 32, 64, 128, 256, 512, 256, 128, 64	High ζ masked L2	82.03%	0.129	0.069
Model LSTM	9 – 64 (LSTM units), 32, 64, 128, 256, 512, 256, 128, 64	High ζ masked L2	77.71%	0.139	0.069

Fig. 8 Ablation Study Results based on Model Architectures and Loss Functions



```
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
input_1 (InputLayer)        [(None, None, 4)]         0

reshape (Reshape)           (None, None, 4)           0

lstm (LSTM)                 (None, 64)                17664

dense (Dense)               (None, 32)                2080

dense_1 (Dense)             (None, 64)                2112

dense_2 (Dense)             (None, 128)               8320

dense_3 (Dense)             (None, 256)               33024

dense_4 (Dense)             (None, 512)               131584

dense_5 (Dense)             (None, 256)               131328

dense_6 (Dense)             (None, 128)               32896

dense_7 (Dense)             (None, 64)                8256

output (Dense)              (None, 19900)             1293500

=================================================================
Total params: 1,660,764
Trainable params: 1,660,764
Non-trainable params: 0
_________________________________________________________________
```

```
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
input_1 (InputLayer)        [(None, None, 4)]         0

reshape (Reshape)           (None, None, 4)           0

dense (Dense)               (None, None, 32)          160

dense_1 (Dense)             (None, None, 64)          2112

dense_2 (Dense)             (None, None, 128)         8320

dense_3 (Dense)             (None, None, 256)         33024

dense_4 (Dense)             (None, None, 512)         131584

dense_5 (Dense)             (None, None, 256)         131328

dense_6 (Dense)             (None, None, 128)         32896

dense_7 (Dense)             (None, None, 64)          8256

output (Dense)              (None, None, 19900)       1293500

=================================================================
Total params: 1,641,180
Trainable params: 1,641,180
Non-trainable params: 0
_________________________________________________________________
```

Fig. 9 Network Architecture of Model LSTM (left) and Model Deep (right)

Performance of the four models is also evaluated on example that has the following system variables: $a = 2.477$, $b = 3.494$, $U_0 = 6.550$, and $\zeta = 6.252$. The results of the predicted RHS of KP model are depicted in Figure 10. "Base Model" fails to accurately predict the oscillations at higher relative energies. "Model Weighted L2" improves upon this by incorporating the masked loss but still falls short in capturing the complete behaviour. In contrast, both "Model Deep" and "Model LSTM" demonstrate nearly accurate predictions of the RHS, closely resembling the ground truth.

The accurate prediction of the RHS by "Model Deep" and "Model LSTM" also translates into accurate predictions of the energy bands, as shown in Figure 11. "Base Model" and "Model Weighted L2" exhibit significant errors in predicting the energy bands, while "Model Deep" and "Model LSTM" only miss one band. Notably, the first energy band gap is 0.979 eV, and the predictions by "Model Deep" and "Model LSTM" yield 0.789 and 0.776 eV, respectively. Although the prediction of "Model Deep" falls within a 20% range, it is not considered promising due to the poor performance in low energy band gaps, as depicted in Figure 12. This discrepancy may arise from the insufficient data generated for systems with low energy band gaps. However, our models show promising performance for higher energy band gaps, as evident in Figure 12. Note that green line is the accuracy line $(x = y)$, and red line is the fitting line of the predictions in Figure 12.

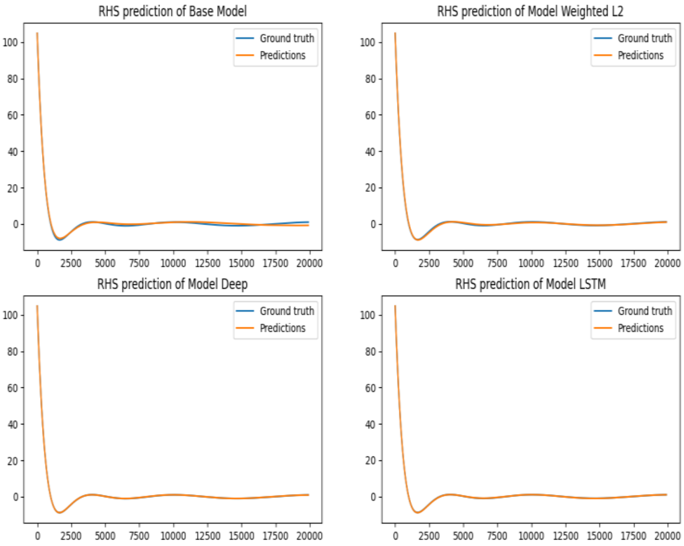


Fig. 10 RHS Prediction of KP using different architectures.

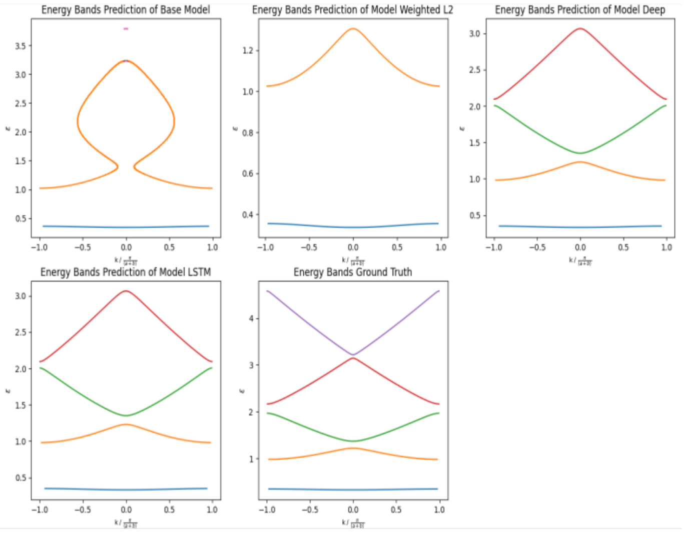


Fig. 11 Energy Bands Prediction using different architectures.

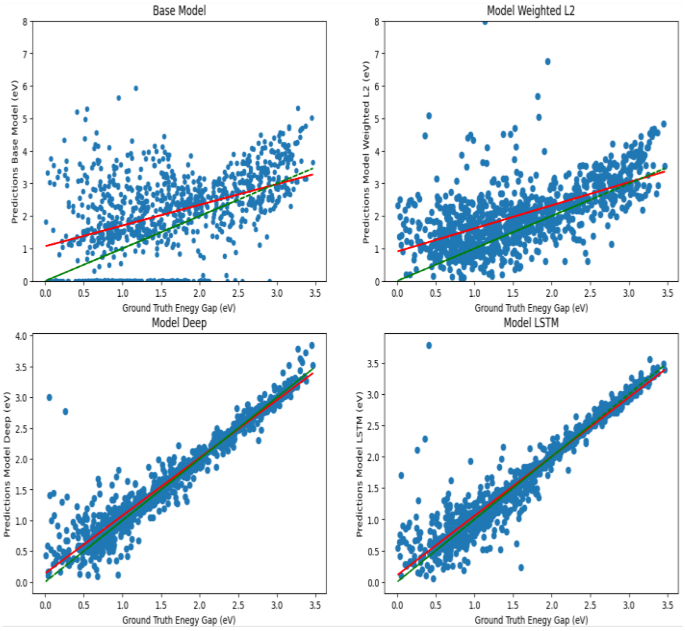


Fig. 12 Histograms of ground-truth vs predictions of energy band-gaps using different architecture.

IV. Conclusions

This study first demonstrates the implementation of the methodology presented in [4] by incorporating additional checks for generating appropriate IW potentials. Our model exhibits comparable performance to the results reported in [4], validating the effectiveness of using deep learning to solve the Schrödinger equation. Through accurate predictions of ground state energy and the ability to speed up the computational process, deep learning offers a promising approach in the realm of quantum physics.

Second, a deep learning architecture to solve the Kronig-Penney model is implemented, specifically focusing on predicting energy bands and energy gaps. In the context of our work, it is worth noting that our study represents the first attempt to solve the Kronig-Penney model using deep learning techniques. Our results demonstrated that using a deeper architecture with a customized weighted L2 loss improved the effectiveness of the model in predicting energy bands, also resulting in higher accuracy for energy band gaps. It should be noted that although the system variables were generated uniformly, the training dataset was non-uniform in terms of energy band gaps, which led to poorer performance in low energy band gaps. To address this issue, generating a uniform training dataset in terms of energy band gaps will be beneficial in improving model performance.

Overall, implementing neural networks provided comparable accuracy while surpassing numerical solutions in terms of speed and computational cost. Additionally, by leveraging the nonlinear processing capability of neural networks, we could capture the direct relationship between the desired output and input data, eliminating the need to solve the exact functions involved, such as the variation principal procedure. Although the variational principle, a fundamental concept in quantum mechanics, provides a powerful framework for solving quantum systems, predicting the exact function using the variational principle can be challenging due to the complexity and nonlinearity of the underlying equations. The variational principle involves finding the minimum energy state by varying a trial wave function, and solving this optimization problem analytically can be computationally demanding, especially for complex systems. Therefore, deep neural networks can simplify the process by approximating the desired function through neuron-wise structures for solving fundamental physics problems.

V. References

[1] Tanaka, A., Tomiya, A., & Hashimoto, K. (2021). Deep Learning and Physics. Mathematical Physics Studies. Springer Singapore. DOI: 10.1007/978-981-33-6108-9.

[2] Pfau, David & Spencer, James & Matthews, Alexander & Foulkes, Matthew. (2020). Ab initio solution of the many-electron Schrödinger equation with deep neural networks. Physical Review Research. 2. 10.1103/PhysRevResearch.2.033429.

[3] https://github.com/jmcelve2/cheshire

[4] K. Mills, M. Spanner, and I. Tamblyn, "Deep learning and the Schrödinger equation," Physical Review A, vol. 96, no. 4, Oct.2017.[Online].Available:https://doi.org/10.1103/physreva.96.042113. (doi: 10.1103/physreva.96.042113)

[5] https://www.youtube.com/watch?v=oetHHDC3dds

[6] *Ashcroft, Neil; Mermin, N. David (1976). Solid State Physics. New York: Holt, Rinehart and Winston. ISBN 978-0-03-083993-1.*