Week 1: 03/02/2025

During 2024 Trimester 3, I did an internship with the Quantum Nanotechnology Group at my university. I worked on solving the Schrödinger equation using Crank-Nicholson method. This experience made me interested in combining machine learning with quantum mechanics.
After discussing my proposal with my supervisor, we decided to do my master's project on using Physics-Informed Neural Networks (PINNs) to solve the many-electron time-dependent Schrödinger equation.

Key Papers Reviewed:

- Shah et al. (2022): Applied PINNs to 1D TDSE, focusing on loss function design and training challenges.
- Hao et al. (2024): Introduced structure-preserving PINNs, improving long-time stability in PDE solutions.

Next Steps:

- Started a detailed review of how many-electron problems are solved
- Choose a simple benchmark problem to start with
- Explore techniques such as causal training to make the model scalable

Week 2: 10/02/2025

This week, I started by modelling a single electron of mass m trapped in a one-dimensional time-independent harmonic potential. I set the initial condition as the ground state wavefunction and applied Dirichlet boundary conditions. My goal was to use a PINN to evolve the wavefunction over time.

During testing, the model correctly predicted the initial condition, but as time progressed, the wavefunction deviated from the expected evolution, suggesting issues in training stability, loss function weighting, or hyperparameter choices.

In our weekly meeting, my supervisor suggested that I should try recreating the model from Shah et al. (2022) to gain better insights into loss function design, training strategies, and hyperparameter tuning. This paper successfully used PINNs to solve the 1D time-dependent Schrödinger equation, making it a useful reference for improving my approach.

Week 3: 17/02/2025

This week, I focused on two main tasks:

1. Recreating the 2022 Shah et al. model for solving the 1D time-dependent Schrödinger equation using PINNs.
2. Preparing and delivering a presentation on Physics-Informed Neural Networks (PINNs) and loss function equations to my supervisor and his subordinates.

The key difference in the paper's approach was the use of atomic units, where constants like mass m, Planck's constant $\hbar$, and frequency ω are set to 1. Additionally the paper uses a superposition of the ground and first excited states as the initial condition.

Despite implementing these modifications, I still encountered the same issue as last week:

- The model correctly predicts the initial condition.
- However, as time progresses, the Mean Squared Error (MSE) increases significantly, causing the predicted wavefunction to diverge from the expected solution.

Next Steps

- Analyze the loss function and training dynamics more closely to identify possible causes of high MSE.
- Experiment with different architectures, activation functions, and optimisation techniques to improve stability.
- Discuss potential hyperparameter tuning strategies in the next meeting.

Week 4: 24/02/2025

This week, I successfully recreated the baseline model from the 2022 Shah et al. paper for solving the 1D time-dependent Schrödinger equation using PINNs.

Initially, the model was prioritizing the initial condition loss and boundary condition loss over the physics loss. This prevented the network from properly learning the Schrödinger equation.
To fix this, I increased the weight assigned to the physics loss in the total loss function. This adjustment ensured that the PINN focused more on satisfying the governing equation, leading to an improved training outcome.

After tuning the loss weights, the model correctly predicted the probability density graph with a MSE of 10^{-5}, which is acceptable according to the results in the reference paper. This confirms that the PINN is now learning the correct wavefunction evolution.

Next Steps:

- Test the model with separate initial conditions, using:
    1. Ground state only
    2. First excited state only
- Compare the accuracy and convergence for each case.

Week 5: 03/03/2025

This week, I recreated the previous model using two separate initial conditions:

- Ground state only
- First excited state only

Both models successfully learned the time evolution, and I am analyzing the differences in their training stability, loss behavior, and accuracy. This will help understand how different initial conditions impact the convergence and generalization of the PINN.

Alongside these experiments, I also continued refining the first model I had been working on. Now that I have insights from the Shah et al. (2022) model, I am applying similar loss balancing techniques and hyperparameter tuning to improve its accuracy.

This week, I also started drafting my Research Proposal report, which is due next week.

Next Steps

- Complete the research proposal draft and refine it for submission.
- Continue analyzing the results of different initial conditions in the model.
- Implement further optimizations in my original model based on new insights.

Week 6: 10/03/2025

This week, I completed the solution of the single-electron in a 1D harmonic potential model using physical units instead of atomic units. I changed the model architecture to work with,

- Space: nanometers (nm)
- Time: picoseconds (ps)
- Energy: milli electron volts (meV)

This required adjusting the Schrödinger equation, constants used and scaling the inputs, x and t accordingly. After these changes, PINN successfully learned the time evolution of the wavefunction.

Next Steps:

- I will start working on the time dependent moving quantum dot model
- An analytical solution must be generated using the Crank-Nicholson method

Week 7: 17/03/2025

This week, I created an analytical solution for the moving quantum dot model using the Crank-Nicholson method. This will act as a reference t my machine learning model.
I also tried to train the PINN model for this problem. But the model failed to capture wavefunction evolution after the dot started moving.
I realised that the problem may be due to poor point sampling in regions where the wavefunction is non-zero. Using a simple uniform distribution of collocation points does not give model the ability to capture non-adiabatic patterns.

Next Steps:

- Try dynamic point allocation, concentrating more points near the moving quantum dot
- Also, try increasing the overall number of points while keeping the points distribution uniform
- Compare both approaches

Week 8: 24/03/2025

This week, first I tried to increase the number of training points while keeping the distribution uniform. But, the model still struggled to track the moving quantum dot. The improvement was very limited.
Next, I changed the point sampling method to a normal distribution centred around the position of the quantum dot at a given instance of time. This significantly improved the model performance. But the model still hasn’t captures all the details.

Next Steps:

- I should try reducing the time domain to see if it increases model accuracy

Week 9: 31/03/2025

This week, I start by trying the previous week's model, but this time time domain reduced to 20ps from 50ps. I couldn't experience much improvement. Later I also experimented with dynamic loss weight allocation and dynamic t_max during training. But results were either same or worse.
Later I tried to train the model as multiple models. But my supervisors didn't agree to the idea.

Next Steps:

- Implement the model with splitting into multiple models

Week 10: 07/04/2025

This week, I successfully solved the moving quantum dot problem. To improve model performance, I increased the model complexity by adding more layers. This allowed PINN to capture more detailed features of the problem. But the training took longer time.
After analysing the result, we suspected there may be discontinuities in the derivatives of the moving quantum dot, which may be causing the training instability.

Next Steps:

- Reduce the training time by optimising the model architecture
- Create a new loss function to handle the discontinuity

Week 11: 14/04/2025

This week, I reduced the training time by almost half by increasing the point density around t1 and t2. These changes made the training faster and stable.
I also designed a new loss function to potentially handle the discontinuity. However, there was no improvement in the model training stability. Then, I realised that the discontinuity happens at the first derivative of the position of the centre of quantum dot. Hence, the suspected discontinuity have no impact in the model training or performance.

Next Steps:

- Try the model with different v_{QD}
- Make the model more generalisable so that I can include v_{QD} and omega as inputs