# EE798Z Course Project: Machine Learning for Quantum Control

Abhimanyu Singh
Department of Physics, Indian Institute of Technology Kanpur

November 24, 2024

**Abstract**

In this report, I review various applications of machine learning in developing quantum systems and focusing on quantum control, dynamics of an open quantum system, and noise spectroscopy. I discuss standard methods implemented using machine learning to solve these problems. I also worked out an optimal control policy for a two-level system with noise and imperfect measurements using reinforcement learning for two different learning scenarios, data-based and model-based.I also talks about the limitation of the model

## 1 Introduction

Quantum technologies are entering a phase where an advantage over their classical counterparts will soon be visible. However, building these quantum technologies—whether they are sensors, computers, or other devices—is incredibly challenging due to their extreme sensitivity to environmental noise. Additionally, simulating these technologies is resource-intensive, as the dimensions of the Hilbert space increase exponentially with the number of particles. As Feynman believed, classical computers are fundamentally limited in simulating quantum phenomena because of the unique nature of quantum mechanics. While it is possible to approximately simulate the dynamics of an open quantum system with a few quantum particles, developing these quantum technologies remains complex. A critical question is how to decide the optimal parameters for our device to achieve the best performance for a given application. Various researchers have addressed this challenge using advanced algorithms and Monte Carlo simulations on state-of-the-art supercomputers to determine control parameters that govern the evolution of open quantum systems.

In this project, I explored quantum control using machine learning, a nascent field that has gained significant traction in the past 2-3 years. This report documents my exploration throughout the semester across various topics, revealing why machine learning is emphasized as a powerful tool—when applied within its limits and approximations. The report is organized into three main sections: A. Using deep learning for noise spectroscopy, B. Simulating the dynamics of an open quantum system with machine learning, C. Quantum control for building quantum technologies, and

To clarify, except for section C, the first two sections do not contain my original work. As in the earlier parts of my projects, things were not failing in place, So I thought of just documenting what I was doing. Still, when my project fell into place later in the semester, I decided to retain my exploration part in the report. Hence, it can be excluded from the evaluation purposes. I've included my comments and reviews based on mutual information across different papers and codes I explored as a reference for potential future work to solve exciting ML problems in near future

### A. Using Deep Learning for Noise Spectroscopy

Quantum systems are inherently open and inevitably interact with their surrounding environment. They are susceptible to gain and losses and decoherence due to the environment. To mitigate these, we must characterize the environment completely and its interaction with the system under a model.

We focus on two of the works here using machine learning[1,2]. The first one is used to classify the main feature of the spectral density of the environment (which is modelled as a coupled harmonic oscillator to the system, and spectral density is the coupling constants $g_n$ and their frequencies $w_n$). They solved it using a simple 3-layer NN (250-80-3); the input is the Fourier transform coefficients of the expectation measurements in Pauli basis with respect to time, while the output is the class of spectral density, namely ohmic, sub-ohmic, or non-ohmic, using pure dephasing and amplitude damping channel model of decoherence.

The second work goes further by applying machine learning to demonstrate that neural networks can significantly enhance the accuracy of noise spectroscopy. This is achieved by reconstructing the power spectral density characterizing an ensemble of carbon impurities surrounding a diamond's nitrogen-vacancy (NV) centre. The NV electronic spin is controlled with a dynamical decoupling (DD) sequence (specifically, a Carr-Purcell (CP) sequence) to measure its dephasing, thereby characterizing the noise spectral density (NSD) of the nuclear spin bath, i.e., $S(\omega; s_0, A, \sigma)$.

$$S(\omega) = s_0 + A \exp\left(-\frac{(\omega - \omega_c)^2}{2\sigma^2}\right)$$

Their study concludes that NNs trained with data obtained for $N = 16$ reconstruct the NSD more accurately than the best estimate provided by the classical method for $N = 48$. It outperforms the state-of-the-art older methods.

This ends my exploration of how to characterize the environment for an open quantum system. But why characterize when you can directly compute the exact wavefunctions or energy levels with some approximation, even in an open quantum system?

## B. Dynamics of Open Quantum Systems Using Machine Learning

I started out by using the idea of noise spectrum to dynamics and control of the open quantum system, which led me to this paper[11] This article proposes a machine learning framework to characterise and predict the dynamics of open quantum systems, moving beyond traditional Quantum Noise Spectroscopy (QNS) methods. The framework utilises a combination of "whitebox" layers, implementing known quantum mechanics formulas like time-ordered evolution, and "blackbox" layers (e.g., neural networks) to learn the complex relationship between control pulses and measurement outcomes. This allows for the prediction of the VO operator(similar to Lindbladian but can also model non-markovian dynamics too), which considers the influence of noise on the system, without depending on assumptions about the noise or control signals. The framework's performance was validated using simulated datasets of a noisy qubit, demonstrating high accuracy in predicting measurement outcomes. While the article focuses on classical noise, the method can be extended to handle non-Markovian dynamics and various control capabilities. The authors suggest future research could explore optimising the algorithm, particularly for noisy, multi-axis datasets. Additionally, the article highlights the potential for integrating the framework with conventional QNS(Quantum noise spectroscopy) methods like the Alvarez-Suter(measure coherence and use it to refer the power spectrum distribution) method for noise spectrum estimation.The neural network described in the paper uses gated recurrent units (GRUs).

After this, I came across three papers doing the same thing in separate groups[4,5,6] .

They are using neural networks encoded quantum many-body states, termed neural-network quantum states (NQS), which have emerged as a promising tool for the variational treatment of quantum many-body problems. Using gradient-based optimization, the ground-state or time-evolved quantum wavefunctions can be efficiently approximated as solutions to the many-body Schrödinger equation. However, these works build upon neural network density operators (NDO), which parameterize the mixed density operator to describe open quantum systems capable of computing their dynamics. Here, the dimension of the Hilbert space grows as $2^{2^n}$ rather than $2^n$.

There are various approaches used, and I will go through the first one, which is to analytically trace out additional bath degrees of freedom in a purified restricted Boltzmann machine (RBM), yielding an ansatz function that always fulfils all the properties of a physical density matrix. This approach utilizes a variational Monte Carlo algorithm, which relies on the variational principle of quantum mechanics.

$$E = \frac{\langle \psi_{\text{trial}} | H | \psi_{\text{trial}} \rangle}{\langle \psi_{\text{trial}} | \psi_{\text{trial}} \rangle} \geq E_0$$

They try to solve the quantum master equation in the form below :

$$\dot{\rho} = -i[H, \rho] + \sum_j \frac{\gamma_j}{2} \left( 2c_j \rho c_j^\dagger - c_j^\dagger c_j \rho - \rho c_j^\dagger c_j \right) \tag{1}$$

where $\rho$ is the density matrix of the system, $H$ its Hamiltonian, and the $\gamma_j$ and $c_j$ the dissipation rates and jump operators of its dissipation. The index $j$ runs over all dissipation channels. They use a parametrization of the density matrix in terms of complex-valued Restricted Boltzmann Machines (RBM) with the architecture ias shown in image below.

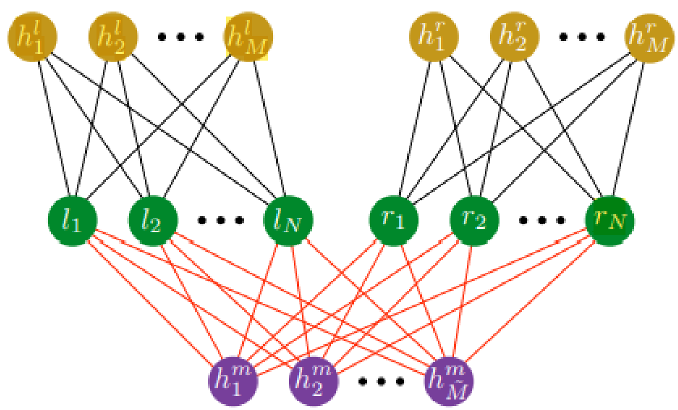



Figure 1: FIG. 1. Sketch of the employed neural network. Visible layer in green and hidden layers in light brown and purple. There is a hidden layer for the row indices $l_i$ and the column indices $r_i$ of $\rho$ (i = 1, 2, . . . , N) with hidden neurons $h_j^l$ and $h_j^r$ (j = 1, . . . , M). A further hidden layer with neurons $h_k^m$ (k = 1, . . . , ) is responsible for the mixing, This image is from Neural-Network Approach to Dissipative Quantum Many-Body Dynamics Michael J. Hartmann, Giuseppe Carleo.

The neurons $h^{(l)}$, $h^{(r)}$, and $h^{(m)}$ are responsible for mediating correlations among, respectively, the column degrees of freedom of the density matrix, the row degrees of freedom, and mixed correlations between the two.

$$\rho_{l_i, r_j} = \exp\left[ \sum_{j=1}^{N} \left( a_j l_j + a_j^* r_j \right) \right] \times \prod_{k=1}^{M} \mathcal{X}_k \times \prod_{p=1}^{\tilde{M}} \mathcal{Y}_p \tag{2}$$

$$\mathcal{X}_k = \cosh\left( b_k + \sum_{j=1}^{N} W_{k,j} l_j \right) \cosh\left( b_k^* + \sum_{j=1}^{N} W_{k,j}^* r_j \right) \tag{3}$$

$$\mathcal{Y}_p = \cosh\left( c_p + c_p^* + \sum_{j=1}^{N} \left( U_{p,j} l_j + U_{p,j}^* r_j \right) \right) \tag{4}$$

where the vector indices $\vec{l} = (l_1, l_2, \dots)$ and $\vec{r} = (r_1, r_2, \dots)$ contain the left ($l_j$) and right ($r_j$) indices for all lattice sites $j$, and the variational parameters are the complex-valued weights $W_{k,j}$, $U_{p,j}$ and biases $a_j$, $b_k$, and $c_p$.

This is how the parametrise the system; optimisation is done via some methods, their can we various optimisation methods. They also give a method to compute the expectations of operators. and verifying their model by applying it to spin-lattice systems and comparing it with RK methods, does pretty well, Their codes were not public so that they couldn't play around with it. But they use a library called NetKet. Didn't run much code on my system, but I implemented some of their notebooks in Colab; works pretty well and is nicely documented on GitHub. This code can be used to solve any open system with, Lindbladian of the form 1.

There was another paper[3] , which by putting some soft constraints instead of hard constraints, were able to better using a CNN, They rearranged the left and right Hilbert space of the spin configurations, which enabled a simple convolutional network architecture to capture the non-equilibrium steady state (NESS) efficiently.

Now, I would like to discuss a very important work as they have developed novel methods and algorithms to retrieve the best-fit Lindbladian to a quantum channel. They have implemented for Google Quantum computer, Cirq. The key strengths of their method is that it can be applied to a single (or small number of) tomographic snapshot(s), is completely assumption-free regarding the structure of the analysed operator, and does not rely on any prior knowledge of the environment or the noise model. At the core of the method is a convex optimisation programme that searches for the closest Lindblad generator within a given distance from the matrix logarithm of the input. This approach successfully deals with imprecise tomographic data, extracting Markovian dynamics within any desired regime of tolerance, and can be used to look for time-independent and time-dependent Markovian dynamics consistent with a series of tomographic snapshots. If no Markovian channel is found, the scheme provides a welldefined quantitative measure of non-Markovianity in terms of the minimal addition of white noise required to "wash out" memory effects and render the evolution Markovian. The inputs to the function are as follows:

Mdata: The tomographic snapshot, representing a measured or simulated quantum state or process.,epsilon (error tolerance parameter): This sets the acceptable error limit for eigenvalue degeneracy checks and distance computations,random samples: Number of random bases to check, used in cases of degenerate eigenvalues to ensure robustness.precision: The level of numerical precision when checking if eigenvalues are degenerate. dI: A large initial value representing the "worst-case" scenario for distances, used as an initial maximum when finding minimal distances,finding_mu (Boolean value): This determines whether the code calculates the non-Markovianity parameter, denoted by $\mu$. If set to True, the code evaluates $\mu$; if set to False, it simply finds the best-fit Lindbladian without computing $\mu$. Running the code with finding_mu = True is significantly slower than with finding_mu = False.

There are three outputs from the function: - Result: This indicates if a Lindbladian was successfully found. The output is True if a Lindbladian was found and False otherwise. - Lindbladian: If a Lindbladian is found, this variable contains the operator; otherwise, it defaults to np.zeros(1). - Distance: The meaning of this output depends on whether a Lindbladian was found. If a Lindbladian was found, Distance represents the distance between $e^{\text{Lindbladian}}$ and the tomographic snapshot (Mdata). If no Lindbladian was found and finding_mu is set to True, Distance represents the non-Markovianity parameter, $\mu$. ,

### C. Control of Quantum Systems using machine learning

Quantum control is essential for designing actual quantum technologies or experiments, which require resilience as it is a complex problem; Quantum systems differ from our familiar classical systems in that we cannot measure the former without adding a probabilistic perturbation to them. Due to this fundamental noise, controlling quantum systems is generally more complex than controlling classical systems. It requires us to know about the environment, how measurements are happening, and How to change the sequence or value of transformation parameters over time in an environment full of noise. As all of the quantum phenomena are being tried to model as quantum circuits and gates, with the help of quantum computers, we could solve these problems. So, designing these circuits was done earlier using quantum control theory, but the problem with every quantum mechanical thing is the exponential scaling of the dimensions. Using Reinforcement learning, I implemented control protocols focussing on generating a specific quantum state in a two-dimensional Hilbert space.

This section is divided into three parts first section gives a brief overview of the Quantum control theory [10] In the second part, I give details about the noise model and discuss about the RL scheme used and the results obtained from a two-level system. Using the Results and codes of these two papers.[8,9] In the third part, I discuss the potential application and different approaches that can be used to improve its robustness, and the limitation of the model

## 1.Quantum control theory

Quantum control theory employs various methodologies to steer quantum systems toward desired states or behaviours. So their area various ways many from classical control theory:

1. **Open-Loop Control:** Predefined control sequences applied without feedback.
   - *Methodology:* Develop a system model, solve optimization problems (e.g., GRAPE, CRAB), and design control pulses.
   - *Applications:* High-precision quantum gates, dynamical decoupling.
2. **Closed-Loop Control (Feedback Control):** Real-time feedback modifies control actions based on measurements.
   - *Methodology:* Measure system properties, update control actions using feedback laws, and implement measurement-based or coherent feedback control techniques.
   - *Applications:* Noise suppression, error correction, state stabilization.
3. **Robust Control:** Ensures performance under model or environmental uncertainties.
   - *Methodology:* Design control laws accounting for potential errors using tools like Lyapunov functions or H-infinity control.
   - *Applications:* Managing environmental noise, uncertainties in initial states.
4. **Optimal Control Theory:** Finds efficient control paths by solving optimization problems.
   - *Methodology:* Use Pontryagin's Maximum Principle to minimize cost functions (e.g., energy, time) while maximizing fidelity.
   - *Applications:* Fast state transfer, high-fidelity quantum gates.

## 2.Control Model

The system model involves a finite-dimensional quantum system subjected to Markovian noise, generalized measurements, and unitary control actions. The goal is to minimize the distance between the actual state and a desired target state after a finite time horizon.

**Noise Models:** These models describe how the state evolves with interaction with the environments and thy are represented by Kraus, I discuss the Kraus maps for the noise model I have used:

**Depolarizing Channel:** This channel represents the loss of information where the qubit transitions to a thoroughly mixed state with probability $p$. The Kraus operators are:

$$K_0 = \sqrt{1-p}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \quad K_1 = \sqrt{\frac{p}{3}}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, \quad K_2 = \sqrt{\frac{p}{3}}\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}, \quad K_3 = \sqrt{\frac{p}{3}}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$$

**Amplitude Damping Channel:** This channel models energy dissipation, such as spontaneous emission, where the excited state decays to the ground state. The Kraus operators are:

$$K_0 = \begin{bmatrix}1 & 0\\ 0 & \sqrt{1-\gamma}\end{bmatrix}, \quad K_1 = \begin{bmatrix}0 & \sqrt{\gamma}\\ 0 & 0\end{bmatrix}$$

**Bit Flip Channel:** This channel represents a noise model where the qubit flips its state ($|0\rangle \leftrightarrow |1\rangle$) with probability $p$. The Kraus operators are:

$$K_0 = \sqrt{1-p}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \quad K_1 = \sqrt{p}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$$

**Measurments:** Measurements are not perfectly projective. The operators are such that they satisfy the completeness condition. For two-dimensional systems, They are:

$$M_0 = \begin{pmatrix}\sqrt{1-\epsilon} & 0\\ 0 & \sqrt{\epsilon}\end{pmatrix}, \quad M_1 = \begin{pmatrix}\sqrt{\epsilon} & 0\\ 0 & \sqrt{1-\epsilon}\end{pmatrix}.$$

**Quantum control:** In general, control is implemented by unitary transformation with control Hamiltonian parametrised by beta. Here, we are considering the simple model, where it;s actions are not coupled to the system; only the parameter gets modified based on the feedback.

The operator and Hamiltonian are:

$$U_\beta(t) = e^{-iH_c(\beta_t)}, \quad H_c(\beta_t) = i\left[\beta_t\left(a - a^\dagger\right)\right].$$

We need to get the beta value, as measurement outcomes form a discrete stochastic process, beta is also becoming stochastic. As a cost function, I have used pseudo distance based on fidelity.

$$d(\rho, \rho_{\text{target}}) = 1 - F(\rho, \rho_{\text{target}}), \quad F(\rho, \rho_{\text{target}}) = \text{tr}\left[\sqrt{\sqrt{\rho_{\text{target}}}\rho\sqrt{\rho_{\text{target}}}}\right]^2.$$

The target state matrix $\rho_{\text{target}}$ is:

$$\rho_{\text{target}} = \begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix}.$$

**Reinforcement Learning** I have used reinforcement learning, as we have to make sequential decisions based on a reward, fidelity and timesteps. So, it turns out to be a promising approach, but other deep learning techniques, such as PINNS, have been used. These are the details of the RL:

- **RL Environment:** The environment consists of a quantum system with potential noise, coupled with a measurement apparatus. The environment receives control parameters and provides either measurement outcomes or filtered state estimations to the agent.

- **Agent's Task:** The agent's task is to determine the optimal control law that drives the quantum state close to a target state. The agent achieves this by optimising a policy, which determines the control parameters based on the available information.

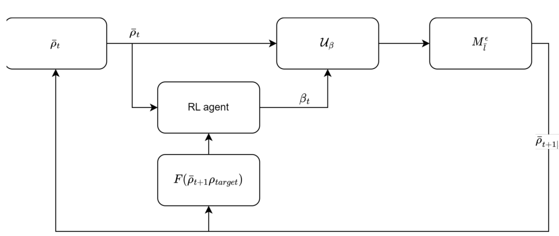
- **Reward Function:** Two reward functions are considered. When the agent has access to the quantum system's estimated state, fidelity is used as the reward function, measuring the closeness of the current state to the target state. When only measurement outcomes are available, the reward function is based on whether the outcome aligns with the desired target state.

- **Policy Gradient:** I have implemented the **Stable Baselines3** framework in which a neural network is used to compute the policy $\pi_\theta$, with the multidimensional parameter $\theta$ encompassing all the network's weights and biases. The neural network takes the current state $(s_t)$ as an input vector and produces the action probabilities.and used openai gym[12]

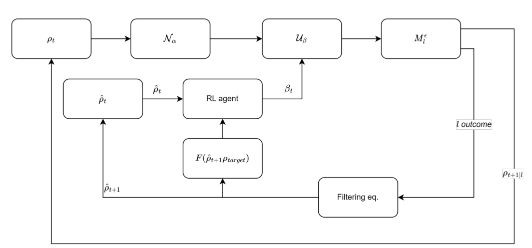
As our experimental constraints differ from time to time, Sometimes we have fully observable system, partially observable etc So, I have tried implementing the model in three different scenarios(following Guatto and Sivak's work), out of which only two could work. So these are three different learning scenario:

- **Model-Based Learning Scenario (MBs):**
  In this scenario, the RL agent is trained using a model of the quantum system without noise. During training, the agent receives the nominal state of the quantum system and learns to select control parameters that maximise the fidelity with the target state. During validation, noise is introduced, and the agent uses the filtered state, an estimation based on the observed noisy measurement outcomes and the nominal dynamics. This scenario assesses the agent's ability to generalise knowledge to noisy conditions.

- **Data-Based Learning Scenario (DBs):**
  The RL agent is trained using the filtered dynamics, which incorporates noisy measurements and updates the state estimate using a filtering equation. The agent receives the filtered state as input and learns to choose control parameters that again maximise fidelity with the target state. During validation, the same filtered dynamics are used to evaluate the agent's performance under the conditions on which it was trained.

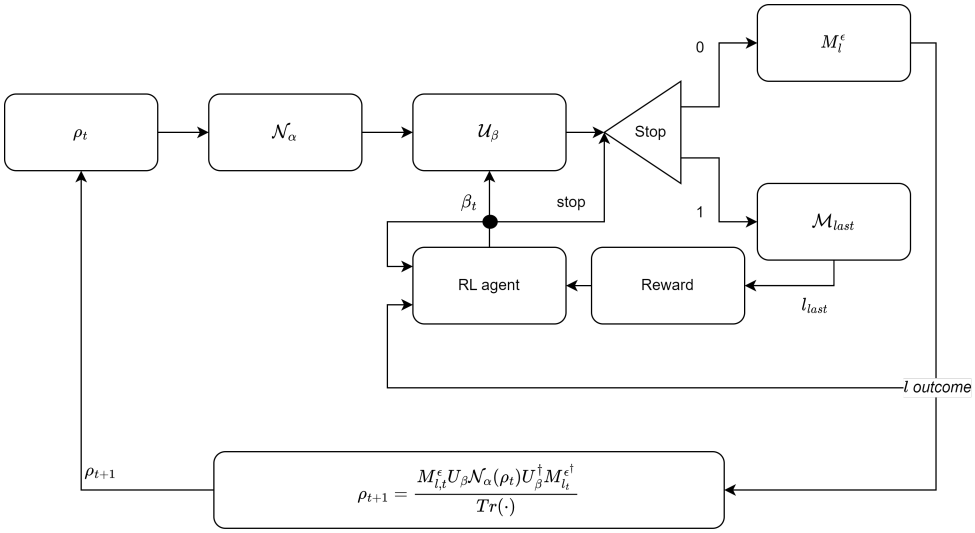



**Model-Based Learning Scenario**




**Data-Based Learning Scenario**

- **Model-Free Scenario (QOMDP):**
  This scenario uses a Quantum Observable Markov Decision Process (QOMDP) where the agent does not have access to the quantum state or any estimation of it. Instead, the agent receives the measurement outcome and previous control action as input, and its task is to select the next control action and decide whether to stop the episode. The reward function depends on whether the final measurement outcome aligns with the desired target state. This approach removes the reliance on a model of the quantum system, but its performance depends heavily on the quality of the measurements.




figureModel-Free Learning Scenario (QOMDP)

# 2 Results of Model-Based and Data-Based Scenarios

The results of the Model-Based and Data-Based scenarios are presented below, showing the impact of various parameters on fidelity, reward, and timesteps.

## 2.1 Model-Based Scenario

The following plots depict the fidelity, reward, and timesteps as a function of noise ($\epsilon$) and parameter ($\alpha$) for the Model-Based scenario:

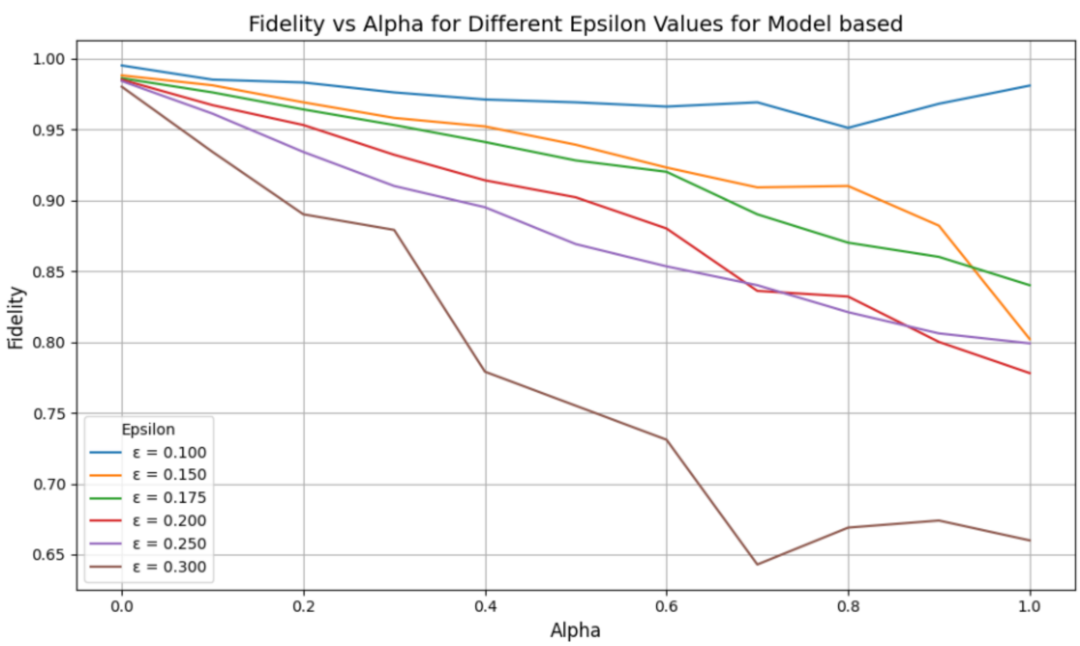



fig Fidelity vs Noise (Model-Based)

## 2.2 Data-Based Scenario

The following plots depict the fidelity, reward, and timesteps as a function of noise ($\epsilon$) and parameter ($\alpha$) for the Data-Based scenario:

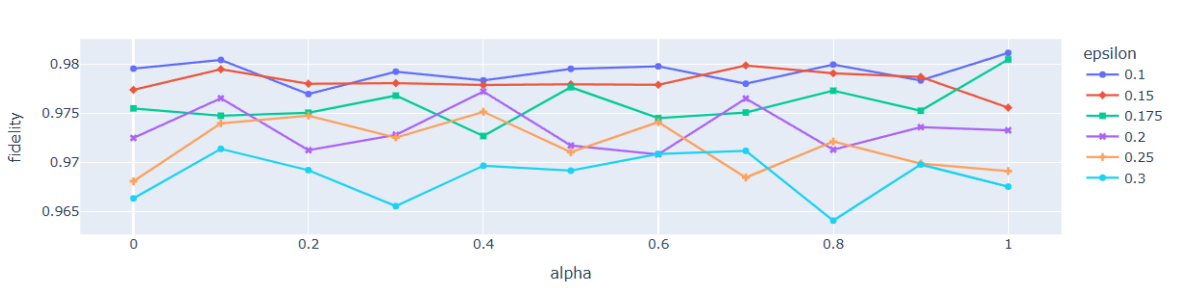



fig: Fidelity vs Noise (Data-Based)

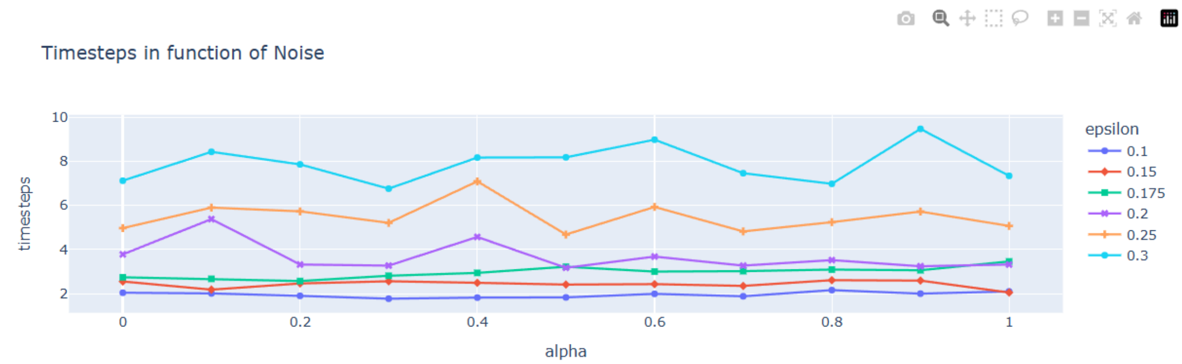



fig: Time steps vs Noise (Data-Based)

# 3 Discussion

It is clear from the plots that the DBM model outperforms the learning in model-based scenarios, even in boisterous environments. So, there is a tradeoff between measurement inaccuracy and noise tolerance. These schemes can easily outperform classical control methods when noise or inaccuracies exist in the measurements. However, these models also have their limitations and can be very resource-intensive, so proper resource budgeting must be done before using any of the methods.DBS and MBS has one of the problems is that they have model bias; we have assumed a model, even when the environment of he open system cannot be modelled exactly or is computably heavy, but here we can QODMP scenarios which don't have any model bias, which helps us get rid of f this limitation. If these models are used with proper hard and soft constraints can be used to Quantum control for a lot of complicated quantum technologies.

Also, using the previous works of characterising the environment of the quantum system or inferring the quantum channel can help us use the DBS and MBS in more efficient work.

This work is on a two-level system with ideal settings, yet it isn't easy to do, with my machine. But this can be even extended to quantum computers too; Google claims to be doing it.Also new approaches can also be used like PINNS[13]

# 4 Conclusion

In this report, I have tried documenting my exploration of different problems in quantum mechanics using machine learning. Sections A and B are the compilation of my exploration, While section C discussed the approach and results of implementing quantum control for two-level open quantum systems in different learning scenarios, And found them making pretty decent even with high noise and imperfect measurements. Apologies for any goof-ups and mistakes, as it mostly is a first or second draft; I wanted to write more but was constrained by document length and time.

# 5 References


1. Spectral Density Classification For Environment Spectroscopy: Jessica Barr, Giorgio Zicari, Alessandro Ferraro, Mauro Paternostro. https://arxiv.org/pdf/2308.00831

2. Deep learning enhanced noise spectroscopy of a spin qubit environment: Stefano Martina, Santiago Hernández-Gómez, Stefano Gherardini, Filippo Caruso, Nicole Fabbri. https://arxiv.org/pdf/2301.0

3. Deep neural networks as variational solutions for correlated open quantum systems Johannes Mellak, Enrico Arrigoni, and Wolfgang von der Lindenhttps: //arxiv.org/pdf/2401.14179

4. Fitting quantum noise models to tomography data Emilio Onorati, Tamara Kohler, and Toby S. Cubitt, Quantum 7, 1197 (2023).

5. Neural-Network Approach to Dissipative Quantum Many-Body Dynamics Michael J. Hartmann Giuseppe Carleo :https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.122.250502

6. A. Nagy and V. Savona, preceding Letter, Variational Quantum Monte Carlo with Neural Network Ansatz for Open Quantum Systems, Phys. Rev. Lett. 122, 250501 (2019).

7. F. Vicentini, A. Biella, N. Regnault, and C. Ciuti, following Letter, Variational Neural Network Ansatz for Steady States in Open Quantum Systems, Phys. Rev. Lett. 122, 250503 (2019).

8. Model-Free Quantum Control with Reinforcement Learning; Phys. Rev. X 12, 011059 (2022)V. V. Sivak ,1,* A. Eickbusch ,1 H. Liu,1 B. Royer ,2 I. Tsioutsios,1 and M. H. Devoret1,† code https://github.com/v-sivak/quantum-control-rl

9. Improving robustness of quantum feedback control with reinforcement learning Manuel Guatto(some figures are from the paper too) : https://arxiv.org/pdf/2401.17190

   code: https://github.com/ManuelGuatto/RL4RobustQC

10. Quantum control theory and applications: A survey https://arxiv.org/pdf/0910.2350

11. Characterization and control of open quantum systems beyond quantum noise spectroscopy:https://www.natu 020-00332-8

12. https://gymnasium.farama.org/

13. Physics-informed neural networks for quantum control:https://arxiv.org/abs/2206.06287