

Project Proposal

Andry Paez

1 The Search for the Elusive Particle: Higgs Boson Detection via Deep Learning

1.1 Background

In 2012, the discovery of the Higgs boson marked a major breakthrough in particle physics. This project aims to build a deep learning model to detect the Higgs boson in datasets from particle collisions, such as those available on CERN's Open Data Portal. Deep learning can improve the accuracy of particle detection, making it an innovative method in particle physics research.

1.2 Approach

Data Source: The project will use the Higgs boson dataset from CERN, which includes features like invariant mass, missing energy, and transverse momentum.

Algorithm: A neural network classifier will be implemented using Python libraries such as TensorFlow or PyTorch.

Goals: The goal is to train the deep learning model to classify events as either signal (Higgs boson) or background, aiming for high classification accuracy.

Tools: Python packages such as SciPy, Pandas, and Matplotlib for data processing and visualization, and TensorFlow or PyTorch for building the model.

Challenges: Working with large datasets, limited deep learning experience, and tuning hyperparameters to ensure good generalization.

1.3 Citations

1. CERN Open Data Portal, available at <https://opendata.cern.ch/>.
2. P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature Communications*, vol. 5, p. 4308, 2014. Available at: <https://doi.org/10.1038/ncomms5308>.

2 Quantum Weirdness: Exploring Quantum Error Correction

2.1 Background

Quantum error correction is crucial for maintaining qubit coherence in quantum computers. This project will simulate simple quantum error-correcting codes, like the Shor or Steane code, to protect quantum information from noise and decoherence. Understanding quantum error correction is key to making quantum computers more reliable.

2.2 Approach

Data Source: This project does not need external data but will simulate qubits and noise.

Algorithm: Using libraries like Qiskit or Cirq, I will simulate quantum gates and error correction codes.

Goals: The objective is to encode qubits, simulate noise, and demonstrate the effectiveness of the Shor code for correcting errors.

Tools: Python libraries such as Qiskit, NumPy, and Matplotlib.

Challenges: Quantum mechanics concepts, simulating realistic noise, and understanding quantum gates.

2.3 Citations

1. J. Roffe, "Quantum Error Correction: An Introductory Guide," *Contemporary Journal*, 2019.
2. Qiskit Documentation, available at <https://qiskit.org/documentation/>.

3 Tunnel Vision: A Quantum Leap Through Barriers

3.1 Background

Quantum tunneling occurs when particles pass through barriers, even when their energy is lower than the barrier height. It's essential in phenomena like alpha decay and semiconductor operations. This project aims to simulate quantum tunneling using the time-dependent Schrödinger equation in one dimension, allowing for visualization and analysis of how a particle behaves when it encounters a potential barrier.

3.2 Project Approach

Schrödinger Equation: I will use finite difference methods to solve the time-dependent Schrödinger equation, simulating the wave packet propagation.

Potential Barrier: Various potential barriers (rectangular, Gaussian, etc.) will be simulated to see how the barrier's shape and height affect tunneling probability.

Numerical Stability: The Crank-Nicolson scheme will be employed to ensure numerical stability.

Tools: Python libraries like NumPy, SciPy, and Matplotlib will be used, along with QuTiP for solving more complex setups.

3.3 Citations

1. Louis S., "Quantum Tunneling Notebook," *GitHub*, available at <https://github.com/louishrm/Quantum-Tunneling/blob/main/QM%20Tunnelling.ipynb>.

4 Monte Carlo Madness: A Stochastic Approach to Quantum Systems

4.1 Background

Monte Carlo methods are widely used in quantum mechanics for calculating complex integrals, particularly in the path integral formulation. This project will apply Monte Carlo simulations to study simple quantum systems, such as a particle in a box or a harmonic oscillator, and use random sampling to compute observable properties like energy or probability distributions.

4.2 Project Approach

Monte Carlo Integration: Basic Monte Carlo methods will be used to approximate integrals in quantum systems, such as calculating the energy of a quantum particle.

Path Integral Simulation: By simulating different particle trajectories and their weights, this project will explore how the quantum state evolves.

Comparison with Analytical Results: The results will be compared with analytical solutions to verify the accuracy of the simulation.

Tools: Python libraries like NumPy, Matplotlib, and potentially PyMC for more advanced statistical sampling.

Citations

1. Quantum Monte Carlo Methods. Physics 142 Lab, UC San Diego, 2016, available at https://courses.physics.ucsd.edu/2016/Spring/physics142/Labs/Lab5/QM_MonteCarlo.pdf.
2. Thijssen, J., “Quantum Monte Carlo Methods,” in *Computational Physics*, Cambridge University Press, 2007, pp. 372-422.