National College of Ireland

Configuration Manual

MSc Research Project
MSc in Artificial Intelligence

Bhagyam Babu
Student ID: x23279745

School of Computing
National College of Ireland

Supervisor: Prof. Mohit Garg

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Bhagyam Babu
Student ID:	X23279745
Programme:	MSc in Artificial Intelligence **Year:** 2024(Sept)
Module:	MSc Practicum Part 2
Lecturer:	Prof. Mohit Garg
Submission Due Date:	16-September-2025
Project Title:	The Carbon Cycle Reimagined: Advanced CO2 Capture, Quantum-Catalysed Conversion, and Energy-Efficient Processing
Word Count:	745 **Page Count:** 4

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Bhagyam Babu

Date: 11-September-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☑
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☑
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☑

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Bhagyam Babu
Student ID: x23279745

1 System Overview

This manual document explains the configuration and setup of the CO2 Conversion and Capture System, which integrates quantum machine learning for catalyst analysis, MOF screening for CO2 capture, and energy optimisation.

2 Environment Setup

Prerequisites

Python 3.8 or higher
pip (Python package installer)

pip3 install numpy==2.3.2
pip3 install pandas==2.3.1
pip3 install scikit-learn==1.7.0
pip3 install matplotlib==3.10.3
pip3 install seaborn==0.13.2
pip3 install scipy==1.16.0
pip3 install joblib==1.5.1
pip3 install qiskit==1.4.3
pip3 install qiskit-machine-learning==0.8.3
pip3 install qiskit-aer==0.17.1
pip3 install qiskit-ibm-runtime==0.40.1
pip3 install ase==3.25.0
pip3 install pymatgen==2025.6.14
pip3 install pymatgen-analysis-diffusion==2024.7.15

3 Dataset Creation

CO2 Capture Dataset

Source

File: ASR_data_SI_20250204.csv (located in processed_data/ directory)
Format: CSV file containing Metal-Organic Framework (MOF) data
URL:https://zenodo.org/records/14510695

```
import pandas as pd
df = pd.read_csv("processed_data/ASR_data_SI_20250204.csv")
```

Features

Primary Features:

ASA (m^2/g) - Accessible Surface Area
PV (cm^3/g) - Pore Volume
LCD (Å) - Largest Cavity Diameter
PLD (Å) - Pore Limiting Diameter

Density (g/cm³)
Derived Features:
SA_PV_ratio = ASA / PV
PV_Density_ratio = PV / Density
LCD_PLD_ratio = LCD / PLD
Norm_ASA = ASA / Density
log_ASA = log(ASA + 1)
sqrt_PV = √PV
Has_OMS_bin - Binary indicator for Open Metal Sites (1 if present, 0 otherwise)

Data Processing

Cleaning:
Converted string values to numeric
Handled missing values by dropping rows with missing critical features

Feature Engineering:
Created ratio-based features
Applied logarithmic and square root transformations
Encoded categorical variables

```
import numpy as np
df['log_ASA'] = np.log1p(df['ASA (m2/g)'])
df['sqrt_PV'] = np.sqrt(df['PV (cm3/g)'])
```

```
numeric_columns = ['ASA (m2/g)', 'PV (cm3/g)', 'LCD ()', 'PLD ()', 'Density (g
    /cm3)']
for col in numeric_columns:
    df[col] = pd.to_numeric(df[col], errors='coerce')
```

CO2 Conversion Dataset

Source

Format: Randomly generated 747 XYZ files containing molecular structures.fair-chem rejected my request to download the dataset as the detail provided was deemed insufficient to verify my status as a student at NCI.

Features

Atomic Structure Features:
Atomic coordinates
Element types
Bond distances and angles

Electronic Structure Features:
Electronegativity
d-band center
Crystal structure information

Metal-Specific Properties :
Electronegativity
d-band center
Crystal structure type (e.g., fcc)

Data Processing

Loading:
Used XYZ files using ASE (Atomic Simulation Environment)
Processed each file to extract molecular features

Feature Extraction:

Calculates coordination geometry
Extracted electronic structure descriptors
Included temperature and pressure parameters

Label Generation:

Created synthetic binding energy labels based on structural features
Used a proxy function to calculate binding energies

Feature Scaling:

Used StandardScaler for normalizing features
Ensured consistent feature ranges for ML models

Validation:

Checkd for missing or invalid data Skiped files with processing errors Maintained consistent feature ordering.

This dataset structure supports both classical machine learning and quantum-enhanced approaches for catalyst analysis and MOF screening.

4 Code Structure

Core Integration Scripts (Root Directory)

catalyst_energy_bridge.py - Main integration script connecting CO2 capture, conversion, and energy modules.
energy_integration_connector.py - Handles communication and data flow between different components.
run_energy_test.py - Script used to run tests across the energy integration pipeline.
test_bridge_integration.py - Integration test script to validate component interactions.

CO2 Conversion Module

CO2_Conversion/quantum_ml_pipeline.py - Implements the quantum machine learning pipeline for catalyst structure analysis.
CO2_Conversion/catalyst_generator.py - Script to generate catalyst configurations for the quantum pipeline.
CO2_Conversion/catalyst_dataset/ - Directory containing XYZ files representing various metal-based catalyst structures.
CO2_Conversion/pycache/ - Auto-generated compiled Python files.
CO2_Conversion/qiskit_env/ - Custom virtual environment configured for running Qiskit and quantum simulations.

CO2 Capture Module

CO2_Capture/screen_co2_mofs.py - Main script for screening metal-organic frameworks (MOFs) for checking CO2 capture efficiency.
CO2_Capture/processed_data/ - Dataset folder
ASR_data_SI_20250204.csv – Source dataset used for training and evaluation.
top10_predicted_mofs.csv – Output list of top MOFs selected for performance

Energy Integration Module

Energy_Integration/optimizer.py - Defines optimization algorithms for energy efficiency.
Energy_Integration/storage_systems.py - Implementation of different energy storage methods.
Energy_Integration/energy_calculator.py - Performs energy-related computations.
Energy_Integration/renewable_energy.py - Integrates renewable energy sources.

Output Files and Visualizations

quantum_ml_results_20250804_004607.png - Visual output from the quantum ML catalyst evaluation pipeline.
CO2_Capture/actual_vs_predicted.png - Model evaluation plot: actual vs predicted surface area.
CO2_Capture/feature_importance.png - Shows feature contributions in the Random Forest model.
CO2_Capture/residual_plot.png - Residual analysis of model predictions.

5 Commands to run the training, testing and evaluation

files

1. CO2 Capture (MOF Screening)
cd /Users/bhagyam.babu/Documents/COLLEGE/NCI/practicum2/Project/CO2_Capture
Run the MOF screening script
python3 screen_co2_mofs.py
Expected Output:
Trained Random Forest model saved as rf_surface_area_model.joblib
Predictions saved as rf_surface_area_predictions.csv
Top 10 MOFs saved as top10_predicted_mofs.csv
Visualization files:actual_vs_predicted.png,feature_importance.png,residual_plot.png

2. CO2 Conversion (Quantum ML Pipeline)
Training and Evaluation

cd /Users/bhagyam.babu/Documents/COLLEGE/NCI/practicum2/Project
source venv/bin/activate # On macOS/Linux
Run the quantum ML pipeline (training and evaluation)
python3 CO2_Conversion/quantum_ml_pipeline.py
Expected Output:
Training progress and evaluation metrics
Visualization files saved as quantum_ml_results_*.png

3. Energy Integration
cd /Users/bhagyam.babu/Documents/COLLEGE/NCI/practicum2/Project
Run the energy integration test
python3 run_energy_test.py
python3 test_bridge_integration.py
python3 catalyst_energy_bridge.py
python3 energy_integration_connector.py

energy_integration_connector.py -Coordinates energy management key components
CO2EnergyManager-class that orchestrates all energy operations
This imports the below py files.
EnergyCalculator-for energy requirement calculations
RenewableEnergy-for power generation
StorageSystems-for energy storage
EnergyOptimizer-for decision making

6 Hardware Specification

Memory	64GB Ram
CPU Cores	14 total (10 performance + 4 efficiency)
Storage	Apple SSD