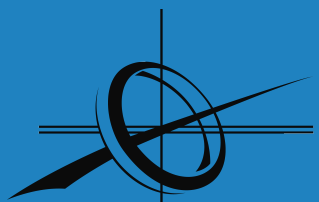
POLITÉCNICA



Universidad Politécnica de Madrid

ETSI SISTEMAS INFORMÁTICOS

Exploring the Quantum Frontiers of Generative Adversarial Networks in Chemical Catalysis

Master's Thesis

Master in Quantum Computing Technology

Author:
Oscar Novo Diaz

Tutors:
Alejandro Borrallo
Carlos Kuchkovsky

30th September 2023

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Exploring the Quantum Frontiers of Generative Adversarial Networks in Chemical Catalysis

Master's Thesis

Master in Quantum Computing Technology

Academic Year 2022-2023

Author:
Oscar Novo Diaz

Tutors:
Alejandro Borrallo
Carlos Kuchkovsky

Acknowledgments

I would first and foremost like to express my heartfelt gratitude to my tutors, Alejandro Borrallo and Carlos Kuchkovsky, for their invaluable assistance and guidance throughout this project. Their patience and willingness to lend an ear during our numerous discussions have been a cornerstone in the successful completion of this research.

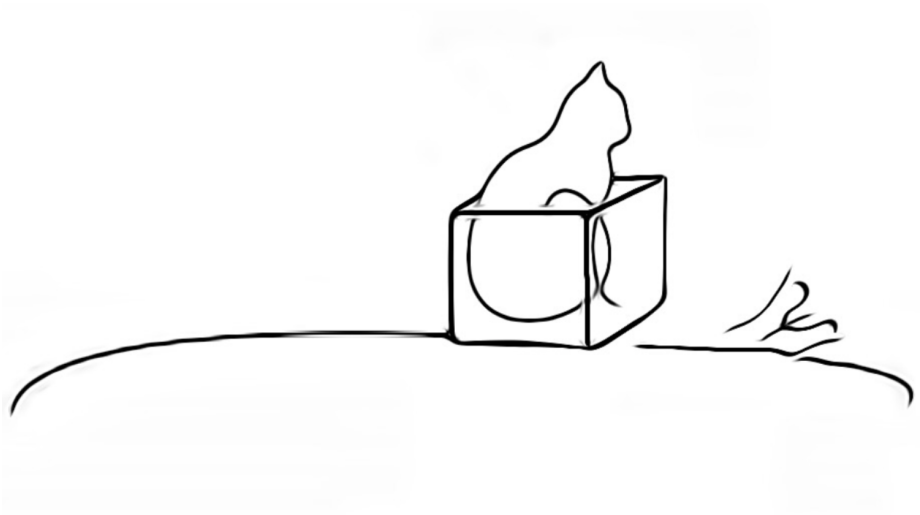
I extend my sincere thanks to all the professors involved in the Quantum Computing program who have imparted knowledge on quantum theories in a manner that was both engaging and profound. Your dedication has fostered a rich learning environment that has been nothing short of inspiring.

A special note of thanks goes out to my esteemed colleagues, with whom I have had the privilege to undertake this journey. Your camaraderie and collaborative spirit have greatly enriched this experience, making it not only a learning curve but also a period of personal growth and cherished memories.

Lastly, I cannot forget the unwavering support and encouragement I have received from my family and friends. Your faith in my abilities and your constant morale boosts have been a source of strength, enabling me to persevere even in the face of challenges. In various ways, big and small, you all have contributed to shaping this journey, aiding me in weaving this tapestry of achievement with threads of encouragement, faith, and love.

Thank you for being the constants in my ever-evolving quantum equation of personal and academic growth.

And I could not conclude these words of appreciation without expressing my sincerest gratitude to all the cats who, in their unique manner, have taken up the role of teaching humans the nuances of the quantum world in an elegant manner. Your feline wisdom and grace have subtly taught humans the fascinating world of quantum mechanics amidst its complexity.



Agradecimientos

En primer lugar, me gustaría expresar mi más sincero agradecimiento a mis tutores, Alejandro Borrallo y Carlos Kuchkovsky, por su inestimable ayuda y orientación a lo largo de este proyecto. Su paciencia y disposición a escucharme durante nuestras numerosas discusiones ha sido una piedra angular en la exitosa culminación de esta proyecto.

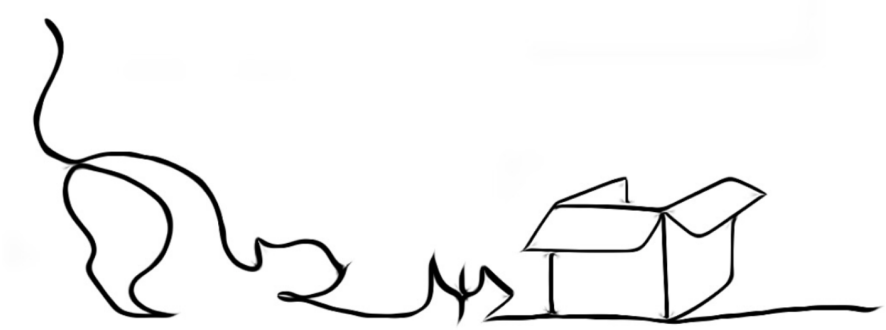
Quiero también extender mi más sincera gratitud a todos los profesores de este master. Su dedicación y motivación a la hora de impartir las asignaturas ha sido extremadamente inspiradora. Gracias por compartir tanta sabiduría de una manera tan cautivadora y profunda.

También quería agradecer a todos mis estimados colegas del master, con quienes he tenido el privilegio de empezar este maravilloso viaje. Todos ellos han enriquezido enormemente esta experiencia, convirtiéndola en un período de crecimiento personal y infinidad de recuerdos valiosos.

Por último, no puedo olvidar el inquebrantable apoyo que he recibido de mi familia y amigos. Su fe incondicional en mi ha sido esa fuente infinita de fuerza e inspiración en todos los aspectos de mi vida. De una forma u otra, cada uno de ellos ha contribuido a dar forma a este maravilloso viaje, ayudándome a tejer este tapiz de logros con hilos de aliento, fe y amor.

Gracias a todos vosotros por ser el pilar inmutable en todas las facetas de mi vida.

Y no podría concluir estas palabras de agradecimiento sin expresar mi más sincera gratitud a todos los gatos que, con la elegancia felina que los caracteriza, nos han enseñado a los humanos el fascinante mundo de la mecánica cuántica de una forma sutil y concisa.



Abstract

Quantum computing and generative Artificial Intelligence (AI) are two burgeoning fields that hold the promise of fundamentally revolutionizing the computational and data science landscapes. While quantum computing is renowned for enabling faster and more efficient calculations through the principles of superposition and entanglement, generative AI models, particularly Generative Adversarial Networks (GANs), are at the forefront of creating novel data and information, paving the way for unprecedented advancements in various domains.

The synthesis of these two fields can potentially spearhead a paradigm shift in the realm of generative AI, fostering the development of quantum-inspired generative models that meld the computational prowess of quantum computing with the data generation capabilities of AI. This thesis delineates the current state-of-the-art in both fields and explores their burgeoning intersection, with a keen focus on the infusion of quantum principles in enhancing generative AI models.

A critical aspect of this exploration is an in-depth study into the application of GANs in the field of chemical catalysis, a domain where the generation of new data can significantly accelerate the discovery and optimization of catalysts. In particular, this work investigates the potential of using GANs to generate novel chemical catalysis processes, emphasizing the creation and optimization of Copper (Cu) binary catalysts. By leveraging the capabilities of quantum computing, these models can potentially facilitate the discovery of more efficient and effective catalytic processes, opening new opportunities for research and development in the field.

The overall aim of this thesis is to unravel the potential advantages and challenges of integrating quantum computing in generative AI, and to pave a road map for future directions in this exciting interdisciplinary domain. Through a meticulous analysis of specific generative AI models, this work endeavors to decipher how quantum principles can augment their capabilities, thereby delineating a promising trajectory for the future of quantum-enhanced generative AI.

List of Acronyms

AI Artificial Intelligence.

CGAN Conditional Generative Adversarial Network.

CGCNN Crystal Graph Convolutional Neural Networks.

Cu Copper.

DFT Density Functional Theory.

FID Frechet Inception Distance.

GANs Generative Adversarial Networks.

GCN Graph Convolutional Network.

GNNs Graph Neural Networks.

IS2RE Initial Structure to Relaxed Energy.

KL Kullback-Leibler.

ML Machine Learning.

OCP Open Catalyst Project.

PQC Parametrized Quantum Circuit.

QGAN-HG Quantum Generative Adversarial Network with a Hybrid Generator.

QM9 Quantum Machines 9.

TOF Turnover Frequency.

VAEs Variational Autoencoders.

VQC Variational Quantum Circuit.

WGAN-GP Wasserstein GAN with Gradient Penalty.

Contents

List of Figures

Chapter 1

Introduction

The journey of catalysis, deeply entrenched in the annals of human history, serves as a testament to human ingenuity and the relentless pursuit of advancement. From its humble origins in ancient civilizations to its modern applications in various industries, catalysis has been a linchpin in fostering human development and technological progress.

In the nascent stages of civilization, catalysis found its initial applications in the production of fermented products such as beer and bread. These processes relied on the catalytic properties of yeast to facilitate the fermentation process, thereby transforming raw materials into consumable products [3]. This marked the inception of humanity's relationship with catalysis, a bond that would only strengthen and diversify over the centuries.

As societies progressed, so did the complexity and sophistication of catalytic processes. The industrial revolution marked a significant milestone in the evolution of catalysis. During this period, catalysis became a vital tool in the chemical industry, facilitating the synthesis of a myriad of chemical products, ranging from dyes to explosives, thereby revolutionizing the manufacturing landscape.

In the 20th century, the Haber-Bosch process emerged as a groundbreaking development in the field of catalysis. This process, which facilitated the mass production of ammonia, a critical component in fertilizers, fundamentally altered the agricultural landscape. By enabling large-scale ammonia production, the Haber-Bosch process played a pivotal role in boosting agricultural productivity, thereby contributing to the green revolution, also known as the Third Agricultural Revolution [4].

However, the monumental achievements of the Haber-Bosch process come with significant environmental repercussions. The process is characterized by a high energy consumption, necessitating large quantities of fossil fuels. Consequently, this has led to substantial carbon dioxide emissions, contributing to

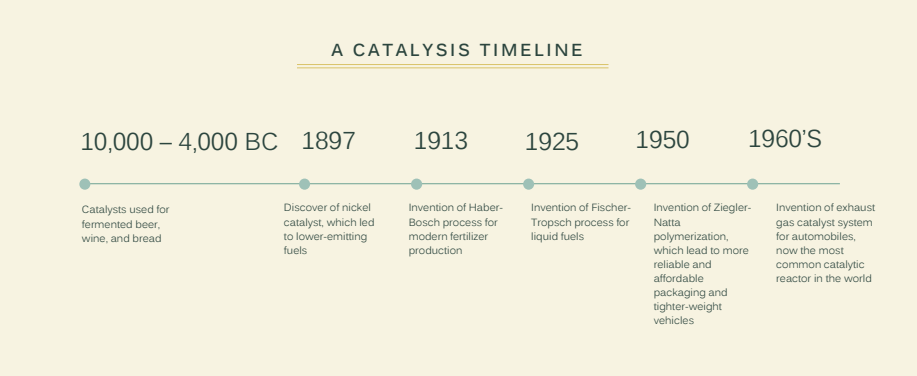


Figure 1.1: Catalysis Through the Ages: A Timeline of Advancements.

about 1% of global CO_2 emissions. This environmental footprint underscores the urgent need for more sustainable and energy-efficient catalytic processes.

In the contemporary era, catalysis continues to find widespread applications across various industries, including the pharmaceutical, petrochemical, and environmental sectors. The development of catalysts that facilitate environmentally friendly processes, such as green catalysis, represents a promising avenue for mitigating the environmental impact of industrial activities.

Looking ahead, the field of catalysis stands at a critical juncture. As the global community grapples with the pressing challenges of climate change and environmental degradation, the role of catalysis in fostering sustainable development has never been more critical. Through the integration of innovative technologies such as artificial intelligence and quantum computing, the field of catalysis holds the promise of spearheading a new era of sustainable and environmentally friendly processes, thereby playing a central role in shaping a greener and more sustainable future.

1.1 Motivation

In the dynamic and ever-evolving landscape of modern technology, the urgency to cultivate sustainable and energy-efficient catalytic processes has escalated significantly, propelled by the escalating concerns of climate change. Within this intricate framework, Generative AI stands as a shining pillar of innovation, emanating rays of promise across a wide array of domains, from fostering the flourishing arenas of artistic creation to leading revolutionary strides in scientific research. The anticipated seamless integration of generative AI with catalysis processes signifies a promising horizon, abundant with prospects for pioneering developments and discoveries in the field.

As we initiate this research journey, we are poised to explore previously untapped domains, intertwining the spheres of quantum computing and generative AI within catalysis processes. Quantum computing, renowned for its extraordinary computational capabilities, holds the promise of being a revolutionary entity, equipped to significantly diminish the computational barriers traditionally encountered in the scrutiny of catalytic processes. In the existing scenario, the utilization of adsorption energy of a molecular structure on a surface serves as a pivotal metric to delineate catalyst activity and facilitate catalyst design. This modeling paradigm encompasses the prediction of energy and forces across diverse configurations of adsorbate molecules at designated interfaces, with a core objective of enhancing the simulation efficacy for both inorganic and organic interfaces in catalysis. The primary computational challenges stem from the intensive Density Functional Theory (DFT) calculations necessitated to ascertain a structure's forces and energy, critical in predicting adsorption energies of minute molecules on potential catalysts. Traditional approaches, grounded deeply in the tenets of DFT, require an immense allocation of computational resources, to the order of 72,000 central processing unit (CPU)-years for an intricate analysis of configurations associated with a duo of catalyst and adsorbent structures[5].

Generative AI, distinguished by its capacity to generate novel data patterns and structures, unveils fresh perspectives in the realm of catalysis. Leveraging the potential of Generative AI enables researchers to conceptualize new catalysts

and refine existing processes, steering the field towards more sustainable and eco-friendly catalytic procedures. The amalgamation of Generative AI and quantum computing is set to instigate a fundamental shift in the field, nurturing an environment of innovation and sustainability.

1.2 Objective of the Thesis

Central to this research initiative is a fervent dedication to advancing the nuanced understanding of quantum mechanics and generative AI, intertwined with a critical exploration of computational chemistry. In a world fervently seeking solutions to mitigate the environmental ramifications of industrial activities, this research seeks to contribute and promote green and sustainable advancements, particularly in the domain of catalysis.

At the heart of this exploration lies the complex process of modeling the energy and forces of atomic systems, a fundamental challenge in the field of computational chemistry. Researchers in this field leverage quantum mechanical simulation tools, such as DFT, to predict adsorption energies of small molecules on potential catalysts - a critical determinant in assessing the efficacy of a catalyst. DFT, grounded in the principles of quantum mechanics, facilitates the simulation of atomic movements within specified scenarios, iteratively modifying the positions of atoms within the system until a state of minimum energy configuration, or relaxation, is achieved. This intricate process involves a detailed analysis of structures comprising both a surface and an adsorbate.

As we embark on this academic voyage, we aim to transcend the limitations of current methodologies, fostering a symbiotic relationship between generative AI and quantum mechanics to pioneer a wave of sustainable catalytic processes. By utilizing the potent capabilities of generative AI in tandem with the computational strength of quantum mechanics, this research aspires to catalyze a paradigm shift in the field, steering the scientific community towards a trajectory that embodies sustainability and energy efficiency. This aligns with the broader vision of paving the path towards the realization of a greener and more sustainable future.

Chapter 2

Fundamentals of Catalysis and Computational Chemistry

Catalysis, a cornerstone in the realm of chemical processes, serves as an instrumental tool in steering reactions to proceed at enhanced rates and often, more selectively. A **catalyst** is defined as *a substance introduced to a reaction to amplify the reaction rate without getting consumed in the process.* It accomplishes this by either lowering the transition state energy, which in turn reduces the activation energy, or altering the mechanism of the reaction, consequently modifying the nature and energy of the transition state. This section offers a panoramic view of catalysis, shedding light on the various types of catalysts and illustrating their profound impact through the lens of the Carbonic Anhydrase reaction, a fundamental biological reaction. Additionally, It delves into the catalysis chemical databases and provides a brief explanation of DFT (Density Functional Theory) and its significance in the field of catalysis.

2.1 Roles and Mechanisms of Catalysts

Catalysts operate on a molecular scale, modifying the progression of chemical reactions by reducing activation energy or altering the reaction mechanism. Key functionalities in catalysis encompass lowering the activation energy, modifying the reaction mechanism, providing a reactant regeneration cycle, and enabling reactions to reach equilibrium faster. Here's a deeper look into these functionalities:

1. **Lowering Activation Energy**: Activation energy is the minimum amount of energy required to initiate a chemical reaction. Catalysts lower this energy barrier, providing an alternative pathway with a lower activation energy. This is achieved by forming transient intermediate complexes with reactants, which require less energy to go on to form products. Mathematically, this can be represented as:

$$E_{\text{activation, without catalyst}} > E_{\text{activation, with catalyst}}$$

   Consequently, the reaction rate increases as a larger proportion of molecules possess the necessary energy to undergo the reaction.

2. **Altering Reaction Mechanism**: Apart from lowering activation energy, catalysts can alter the reaction mechanism, introducing new reaction pathways that might be more favorable than the uncatalyzed pathway. This might involve changing the sequence of elementary steps in the reaction, or even introducing new intermediates, thus changing the nature and energy of the transition state. This manipulation of the reaction mechanism can

sometimes lead to completely different products than would be produced in the uncatalyzed reaction.

3. **Reactant Regeneration Cycle**: In many reactions, catalysts undergo a cycle where they team up with reactants to form intermediates and then are regenerated at the end of the reaction cycle, a concept fundamentally applied in heterogeneous catalysis. This cycle ensures that the catalyst is available to catalyze subsequent reactions, thus not being consumed in the overall reaction process. For example, in the Haber process, nitrogen (N_2) and hydrogen (H_2) gases adhere to the surface of an iron catalyst, facilitating the synthesis of ammonia (NH_3), as represented by the reaction [4]:
$$N_2(g) + 3H_2(g) \rightarrow 2NH_3(g)$$
4. **Equilibrium Acceleration Effect**: While catalysts do not alter the thermodynamics of a reaction, they enable reactions to reach equilibrium faster by speeding up both the forward and reverse reactions equally, which can be critical in industrial processes where time is a significant factor.

Through these mechanisms, catalysts play an indispensable role in a vast array of chemical processes, fostering efficiency and sustainability. Moreover, catalysts play a pivotal role in environmental protection by facilitating reactions that help in reducing pollution. For instance, in automobile exhaust systems, catalytic converters facilitate reactions that convert harmful gases like carbon monoxide (CO) and nitrogen oxides (NO_x) to less harmful substances like carbon dioxide (CO_2) and nitrogen gas (N_2).

Catalysts manifest in various forms, each serving distinct roles in biochemical and chemical reactions. Here, we introduced the three primary types:

1. **Enzymes**: These biological catalysts are proteins that significantly speed up the rate of virtually all of the chemical reactions that take place within cells. They provide an environment within which reactions can take place more rapidly.
2. **Acid-Base Catalysts**: These catalysts function through the donation or acceptance of protons to expedite a reaction. In acid catalysis, a hydrogen ion (H^+) acts as the catalyst, whereas in base catalysis, an hydroxide ion (OH^-) takes up this role. In acid catalysis, the catalyst is usually a H^+ ion, represented in reactions as:
$$\text{HA} \rightarrow H^+ + A^-$$
This type of reaction shows the dissociation of the molecule HA into a hydrogen ion (H^+) and its corresponding anion (A^-).
3. **Heterogeneous or Surface Catalysts**: In contrast to homogeneous catalysts, which are uniformly distributed in the same phase as the reactants, heterogeneous catalysts are situated in a different phase, enabling them to operate at phase interfaces. Heterogeneous catalysts find extensive applications in various industrial processes. In base catalysis, the catalyst is an hydroxide ion (OH), showcased in reactions as:
$$\text{B} + H_2O \rightarrow BH^+ + OH^-$$

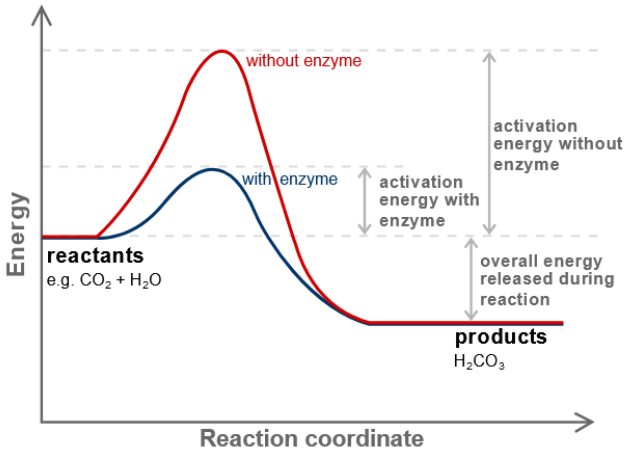


Figure 2.1: Enzyme Action Unveiled: Carbonic Anhydrase Catalyzes Carbon Dioxide Conversion (Source: Image from Wikimedia Commons, CC BY-SA 3.0)

A classic case of enzymatic catalysis in action is witnessed in the Carbonic Anhydrase reaction[6]. The chemical equation representing this reaction is as follows:

$$CO_2 + H_2O \rightleftharpoons H_2CO_3 \rightarrow H^+ + HCO_3^-$$

This enzyme is renowned as one of the fastest known enzymes, orchestrating between 10^4 and 10^6 reactions per second. This rate is extraordinarily higher compared to the uncatalyzed reaction, which proceeds at a rate of approximately 0.2 reactions per second, translating to a phenomenal 10^5 to 10^7 increase in rate.

Furthermore, the oxidation reaction fostered by this enzyme is thermodynamically favored at 25 degrees Celsius, enhancing its significance in physiological contexts, underscoring its profound importance in supporting life.

Catalysts wield a substantial influence on a plethora of domains, including environmental protection and industrial sectors. In industrial settings, catalysts are pivotal in facilitating reactions at interfaces in heterogeneous catalysis, requiring a deep comprehension of molecular interactions and adsorption energies on potential catalyst surfaces. The utilization of catalysts in these domains is guided by the overarching goal to augment the efficiency of inorganic and organic interfaces simulated for use in catalysis, offering a promising avenue for advancements in this field.

2.2 Exploration of Chemical Databases

In the field of computational chemistry, especially in catalysis research, various chemical databases are utilized to gather extensive data about different molecular structures and their properties. These databases serve as a valuable resource for researchers aiming to identify promising catalysts through com-

putational analyses. Below, we delineate several renowned chemical databases integral in the domain of catalysis.

2.2.1 The QM9 Chemical Database

The Quantum Machines 9 (QM9) dataset[7], a significant subset of the GDB-17 chemical database, plays a pivotal role in the domain of computational chemistry, particularly emphasizing its utility in drug design. This dataset encompasses a repository of 134k stable small organic molecules, each characterized by a composition of no more than nine non-hydrogen atoms which primarily include Carbon, Nitrogen, Oxygen, and Fluorine. These molecules were chosen from a larger collection within the GDB-17 database, which includes an astounding 166 billion organic molecules.

The dataset offers detailed descriptors, including but not limited to geometric structures, vibrational frequencies, and electronic properties calculated using DFT. Furthermore, the dataset provides a plethora of quantum mechanical molecular calculations. These calculations provide a comprehensive understanding of the molecules, paving the way for their potential application in various fields including catalysis.

2.2.2 The GDB-17 Chemical Database

The GDB-17 database stands as a monumental repository in the domain of computational chemistry, housing a staggering collection of approximately 166.4 billion molecules [8]. This extensive database encompasses molecules constituted of up to 17 atoms, incorporating elements such as Carbon (C), Nitrogen (N), Oxygen (O), Sulphur (S), and various halogens. A significant feature of this database is its coverage of a size range that encompasses many drugs and substances typically utilized as lead compounds in various applications.

Delving deeper into the technical intricacies of the database, the GDB-17 database is generated based on a systematic and rule-based enumeration approach. From a mathematical standpoint, the database can be perceived as a multidimensional graph, where each molecule occupies a unique position defined by its inherent properties and structural attributes. This mathematical model makes it easier to apply computational algorithms and software for studying and handling data. However, dealing with the enormous size of GDB-17 is a formidable task, and it currently limits its use in advanced virtual screening methods. In such cases, selecting a random subset from the GDB-17 database, which contains a range from hundreds of thousands to several million molecular structures, can produce statistically significant results that are representative enough to serve as a proxy for the complete database.

GDB-17 is commonly used as a base for generating specialized datasets, such as the QM9 dataset. These subsets are created through a detailed curation process aimed at including molecules relevant to specific research areas, such as drug development. The hierarchical structure of GDB-17 allows for more targeted research efforts, providing scientists with the opportunity to focus on subsets of molecules for specialized studies.

In summary, the GDB-17 database is widely utilized in computational chemistry for its extensive collection of molecular structures and associated property

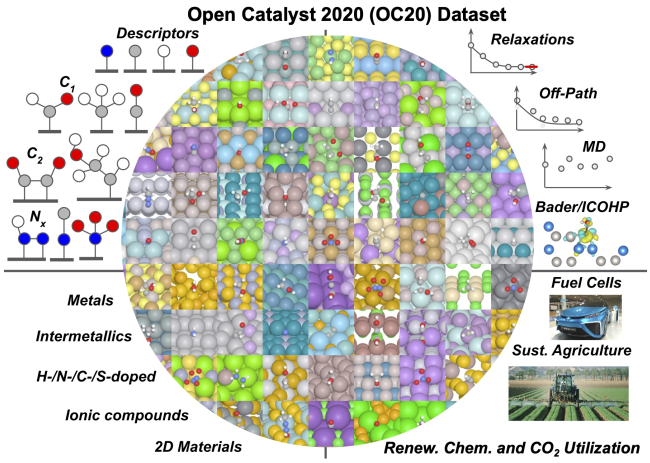


Figure 2.2: Open Catalyst 2020 (OC20) Dataset (Source: [1])

data. It serves as a resource for researchers across various scientific disciplines, aiding in computational studies and data-driven research.

2.2.3 The Open Catalyst Dataset

The Open Catalyst Project (OCP) [9], a pivotal resource in the domain of computational catalysis, has been meticulously developed to foster innovation and advancements in the field. This dataset combines two prominent datasets: OC20 [10] and OC22 [2], which serve as rich reservoirs of data pivotal for catalytic research.

The OC20 dataset laid the groundwork for the subsequent development of the OC22 dataset. This dataset houses a wealth of information pertaining to surface and adsorbate structures, providing researchers with a robust platform to explore and analyze catalytic processes at a granular level. It embodies a systematic compilation of data derived from DFT calculations, offering insights into the energetic and atomic properties of various structures.

Progressing further, the OC22 dataset builds upon the foundation laid by the OC20 dataset. The OC22 dataset, compared to its predecessor, includes training data from oxides. Oxides are critical for the development of oxygen reaction catalysts. To address this, OC22 includes 62,331 DFT relaxations (∼9,854,504 single point calculations) across a range of oxide materials, coverages, and adsorbates in its dataset.

These datasets altogether contain 1.3 million molecular relaxations with results from over 260 million DFT calculations. The data housed in both datasets can be utilized to develop predictive models, facilitating the simulation and analysis of catalytic processes with enhanced efficiency and accuracy. Furthermore, this dataset - and particularly the Open Catalyst Project (OCP) - aims to develop new Machine Learning (ML) methods and models to accelerate the catalyst simulation process for renewable energy technologies, opening avenues for the creation of AI models with the capability to make highly precise predictions regarding catalytic behaviors and their resulting outcomes.

The dataset includes code for training and evaluating models designed for

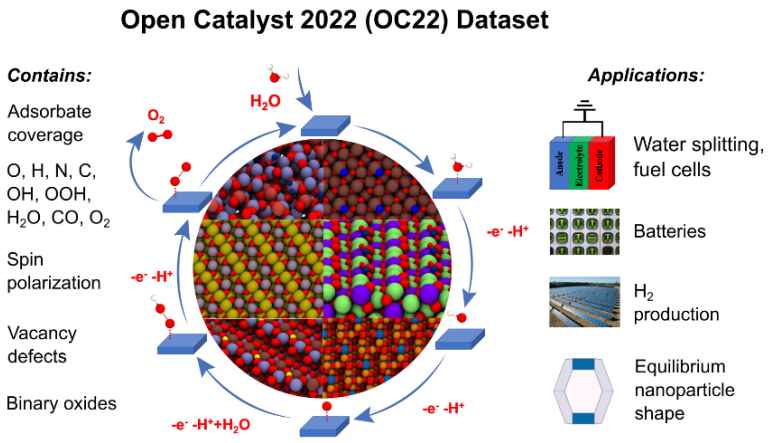


Figure 2.3: Overview of the OC22 Dataset Contents and Impact Areas, Emphasizing the Oxygen Generation Process in Water (Source: [2])

tasks where arbitrary chemical structures serve as input, enabling the prediction of energies, forces, and positions. It can serve as a foundational framework for research projects in this domain.

In conclusion, the Open Catalyst Project is used in the field of computational catalysis, offering detailed data for various research applications. Among its objectives is the aim to develop new machine learning methods and models that can expedite catalyst simulation in renewable energy technologies. The OCP's datasets are part of the available resources that researchers use for studies related to improving the efficiency and sustainability of catalytic processes.

2.3 Density Functional Theory (DFT)

In this section, the focus will be on introducing Density Functional Theory (DFT), a computational quantum mechanical modeling method used for analyzing the electronic structure of atoms, molecules, and materials. The fundamental principles behind DFT will be explained, along with a discussion of its challenges and synergies with AI. By the end of this section, readers should have a comprehensive understanding of how DFT operates as a valuable tool in both theoretical and practical contexts

2.3.1 DFT: Fundamentals

DFT is a computational technique extensively utilized in physics and chemistry to investigate the electronic properties of many-body systems, especially atoms, molecules, and condensed phases, with applications extending to the study of catalysts. The foundation of DFT lies in the Hohenberg-Kohn theorem [11], which establishes that the ground state properties of a many-electron system are uniquely determined by the electron density distribution. In simpler terms, it states that if you know the electron density, then you can determine all other ground state properties of the system, such as its energy. This theorem provides the theoretical justification for using electron density as the basic variable

in DFT calculations, as opposed to wave functions in traditional quantum mechanics.

The total energy functional in terms of the electron density - which is directly related to the Density Functional Theory (DFT) - can be represented as:

$$E[\rho(\vec{r})] = T[\rho(\vec{r})] + V_{ee}[\rho(\vec{r})] + V_{ext}[\rho(\vec{r})] + V_{xc}[\rho(\vec{r})]$$

where:

- $E[\rho(\vec{r})]$ is the total energy
- $T[\rho(\vec{r})]$ is the kinetic energy functional
- $V_{ee}[\rho(\vec{r})]$ is the electron-electron interaction energy
- $V_{ext}[\rho(\vec{r})]$ is the external potential energy
- $V_{xc}[\rho(\vec{r})]$ is the exchange-correlation energy
- $\rho(\vec{r})$ is the electron density at position $\vec{r}$

The total energy functional ($E[\rho(\vec{r})]$) is closely related to the Kohn-Sham equations, which are a central component of Kohn-Sham Density Functional Theory (KS-DFT). In practice, the Kohn-Sham equations are solved iteratively to find the electronic properties, where each electron is described by a wave function $\psi_i(\vec{r})$ satisfying the equation:

$$\left[-\frac{1}{2}\nabla^2 + V_{eff}(\vec{r})\right]\psi_i(\vec{r}) = \varepsilon_i\psi_i(\vec{r})$$

with the effective potential given by:

$$V_{eff}(\vec{r}) = V_{ext}(\vec{r}) + \int d\vec{r'}\frac{\rho(\vec{r'})}{|\vec{r} - \vec{r'}|} + V_{xc}(\vec{r})$$

The goal in DFT simulations is to find the ground state electron density $\rho(\vec{r})$ that minimizes the energy functional, providing a pathway to investigate the properties of the system, including the adsorption energies of molecules on potential catalysts, which is crucial for catalyst design and optimization.

While DFT stands as a powerful tool, it is accompanied by substantial computational demands. The process of relaxation, a pivotal component in DFT analysis, necessitates hundreds of hours of computation on multicore machines, representing a significant computational bottleneck [5]. The initial structure definition, preceding a relaxation run, further adds to the complexity and computational demands of this method.

To further illustrate, the computational bottlenecks primarily revolve around the calculations involving the determination of a structure's forces and energy, vital in predicting adsorption energies on potential catalysts. These calculations, grounded in DFT, require extensive computational resources, highlighting a significant challenge in the broader implementation of this method.

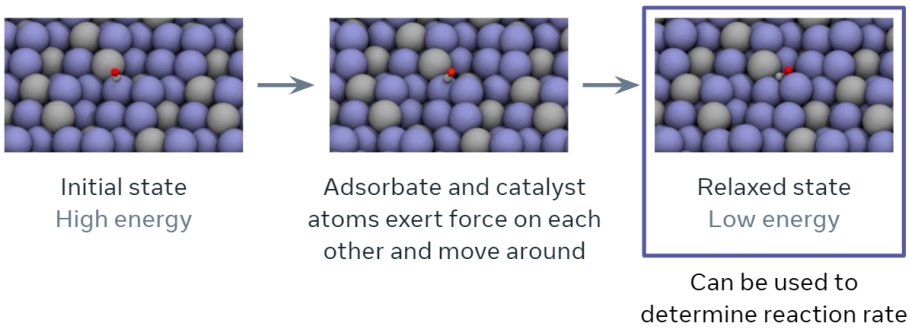


Figure 2.4: Molecular Transition from Initial State to Relaxed Configuration (Source: [1])

2.3.2 Relaxation Phenomena in Catalytic Processes

In the framework of catalysis processes, relaxation represents a critical phase, delineating the trajectory of atomic interactions towards a state of minimum energy. This phenomenon is rooted in the adjustments of atoms' nuclear positions, which are typically initiated with an approximate estimation. The essence of relaxation lies in the refinement of these initial estimates through a series of computational iterations, thereby honing the precision of energy calculations involved in catalysis processes.

During the process of relaxation, the molecular reaction undergoes a series of transformations, evolving towards a configuration where the energy of the adsorbate, interacting with the catalyst, reaches its minimum. This process is facilitated through the application of DFT, a potent tool that enables an in-depth analysis of the energy landscapes associated with different atomic configurations. The iterative nature of DFT analyses, which involves the continuous adjustment of atomic positions, fosters a dynamic environment where the interactions between atoms are continually evolving, gradually converging towards a state of equilibrium, or the lowest energy configuration.

Mathematically, the process can be described using a series of equations that encapsulate the dynamics of atomic interactions. The energy of the system, represented as a function of the positions of the atoms, forms the cornerstone of these analyses. Through the application of optimization algorithms, the system evolves, minimizing this energy function iteratively until a state of minimum energy is attained. This state represents a point in the configuration space where the forces acting on each atom balance out, fostering a state of stability and equilibrium.

Moreover, the relaxation process holds significant implications in the context of catalysis, influencing the efficacy and performance of catalysts. The precise determination of atomic positions serves as a vital parameter in assessing the interaction energies, thereby offering insights into the potential effectiveness of different catalysts. Consequently, relaxation forms a critical component in the design and analysis of catalysis processes, steering the trajectory of research and innovation in this domain towards new frontiers.

2.3.3 Establishing the Initial Structures in DFT Calculations

Before embarking on the relaxation process, it is pivotal to define an initial structure, a fundamental step that lays the groundwork for subsequent computational analyses in catalysis processes. As shown in Figure 2.4, the initial structure delineates the preliminary configuration of atoms within the system, serving as a starting point for the relaxation process. This configuration is often based on theoretical constructs or derived from experimental data, providing a tangible framework to initiate the complex computational processes involved in catalysis.

The definition of an initial structure is a meticulous process, requiring a profound understanding of the chemical and physical properties of the elements involved. Often, this involves a comprehensive analysis of molecular geometries, electronic structures, and potential energy surfaces. The objective is to establish a configuration that is not only theoretically sound but also aligns with empirical observations, thereby fostering a harmonious integration of theory and practice.

From a mathematical standpoint, the initial structure can be represented using a series of parameters that encapsulate the positional coordinates of atoms within the system. Among these, Cartesian coordinates provide the most straightforward representation. Defined within a three-dimensional space, these parameters offer a detailed representation of the atomic structure, facilitating a comprehensive analysis of the system's properties. Moreover, this mathematical representation serves as a foundation for the application of DFT analyses, enabling the detailed exploration of the energy landscapes associated with different atomic configurations.

Furthermore, the definition of an initial structure often involves the integration of quantum mechanics principles, offering a deeper insight into the atomic and subatomic interactions governing the system. Through the application of quantum mechanical simulation tools, researchers can delve into the intricate dynamics of atomic interactions, fostering a rich and detailed understanding of the processes underlying catalysis.

2.3.4 Overcoming Challenges in DFT

Although DFT is highly accurate, it is frequently criticized for its computational expense, especially when applied to large molecular systems or materials with complex structures. This section explores the primary computational bottlenecks in DFT calculations and discusses potential solutions to address these challenges.

The primary challenges associated with DFT include the following:

- **Computational Expense**

  At the core of DFT's computational expense lies the intricate calculations required to predict the adsorption energies of small molecules on potential catalysts. As part of this computational burden are the following:

  - **Initial Structure Determination**

Before initiating any calculations, a precise initial structure must be defined. This is often challenging due to the complex nature of many materials and molecules. Any inaccuracies at this stage can lead to erroneous results in subsequent calculations.

- **Relaxation Process**

  The relaxation process is another key contributor to computational expense. It involves iterative adjustment of atomic positions until the system reaches its minimum energy state or ground state. The complexity of the optimization algorithms used and the high dimensionality of the problem space add to the computational cost.

- **Convergence Issues**

  DFT calculations often face convergence issues, especially in systems with a large number of atoms or strong correlation effects. The SCF (Self-Consistent Field) iterations sometimes fail to converge, requiring the careful selection of mixing parameters and algorithms.

- **Scaling with System Size**

  The computational cost of DFT calculations scales steeply with system size, typically as $O(N^3)$, where N is the number of electrons. This makes it difficult to study large systems or perform long time-scale simulations.

The high computational costs and challenges outlined above have driven the exploration of alternative approaches to accelerate DFT calculations. A pivotal development in this regard has been the integration of AI with DFT.

2.3.5 AI Integration with DFT

The integration of AI with Density Functional Theory (DFT) signifies a methodological shift in computational chemistry. Utilizing the computational efficiency of machine learning algorithms, particularly neural networks, allows for a more expedient execution of DFT simulations, traditionally known for their computational intensity and prolonged runtimes.

Neural networks trained on the OC20 and OC22 datasets optimize their internal parameters via backpropagation algorithms. This results in reduced predictive error and models that approximate DFT-calculated molecular properties with high fidelity. These machine learning models undergo iterative optimization using gradient descent algorithms, fine-tuning their predictive capabilities and thereby offering a computationally efficient alternative to traditional DFT calculations.

The computational advantages of integrating AI into DFT simulations are significant. DFT calculations often require extensive computational resources due to the complexity of solving differential equations and integral transformations. The application of machine learning algorithms in this context provides a cost-effective and time-efficient alternative. This efficiency facilitates faster identification and analysis of potential catalysts, accelerating the rate of research and discovery in computational chemistry.

In summary, the synergy between AI algorithms and DFT methodologies offers a streamlined approach to molecular simulations. This integration has the potential to accelerate research across various scientific domains by providing a less resource-intensive yet accurate method for predicting molecular properties.

2.3.6 Conclusion

In conclusion, the landscape of catalysis research is on the brink of a significant transformation, propelled by the integration of cutting-edge advancements in quantum mechanics, artificial intelligence, and computational chemistry. As we have explored in this background section, the role of catalysts in enhancing the rate of chemical reactions is pivotal, with numerous applications spanning various domains including biochemical reactions, acid-base catalysis, and heterogeneous catalysis.

The development and utilization of comprehensive chemical datasets like QM9 and GDB-17 have facilitated a deeper understanding of the molecular properties and behaviors that govern catalytic processes. Furthermore, the introduction of the Open Catalyst Datasets have marked a significant milestone in the field, offering a rich repository of data that can aid in the development of novel catalysis technologies.

The integration of AI with DFT simulations represents a promising avenue for accelerating the pace of research in this domain. By leveraging the computational prowess of AI, it has become possible to significantly reduce the time and resources required to conduct DFT simulations, thereby facilitating the rapid discovery and analysis of new catalytic processes.

As we move forward, it becomes increasingly important to delve deeper into the intricate dynamics of catalytic processes, exploring new methodologies and approaches that can foster innovation and drive progress in the field. The integration of generative AI and quantum computing with traditional catalysis research holds immense potential, promising to usher in a new era of scientific discovery and innovation. Further details on this topic will be explored in the following chapters.

Chapter 3

Generative AI: Foundations and Trends

Generative AI stands as a pioneering field within the domain of artificial intelligence, offering a radical approach to data generation and analysis. Anchored in complex mathematical theories, it facilitates the development of algorithms capable of synthesizing data akin to the ones they have been trained on. This section endeavors to present an in-depth exploration of the mathematical foundations, recent breakthroughs, and prospective developments in the swiftly evolving sphere of generative AI. It aims to shed light on its core methodologies and applications across diverse scientific realms, including physics and engineering.

3.1 Introduction to Generative AI

Generative AI encompasses algorithms within the unsupervised and semi-supervised machine learning paradigms, which empower computers to generate novel data resembling the contents of a given dataset using existing resources such as text, image, audio and video files, or even coding scripts.

To grasp the concept underlying generative AI more fully, it is vital to delineate it from discriminative modeling. Discriminative models are primarily employed in classifying pre-existing data entities, for instance, categorize images of particles like electrons, muons, and neutrinos into their respective groups, a task predominantly found in supervised machine learning endeavors. On the contrary, generative models strive to comprehend the inherent structure of a dataset to fabricate analogous instances, like generating a realistic representation of subatomic particles such as electrons or neutrinos, a task mainly associated with unsupervised and semi-supervised machine learning initiatives.

Therefore, while discriminative algorithms concentrate on distinguishing specific cases by learning a decision boundary, generative algorithms aspire to decipher the intrinsic attributes of the data, facilitating its reproduction from the ground up. This distinction can be perceived as:

- Discriminative models ascertain the boundaries separating various classes.
- Generative models scrutinize the distribution patterns within individual classes.

Thus, as illustrated in Figure 3.1, discriminative algorithms are essentially concerned with correlating features to labels. In contrast, generative algorithms undertake a reverse approach. Rather than predicting a label based on certain features, they seek to forecast features predicated on a given label.

This brings us to the fascinating realm of generative AI. Generative AI, an extension of generative models, plays a pivotal role in artificial intelligence by

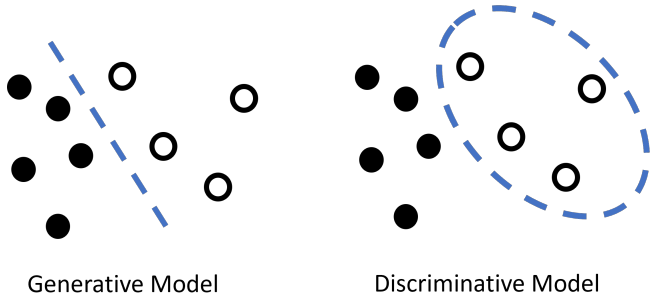


Figure 3.1: Generative and Discriminative Models - A visual representation showcasing the distinct approaches of generative and discriminative models.

harnessing the power of creativity in data generation. While generative models aim to understand and replicate data patterns, generative AI takes a step further in the field of machine learning. It excels at producing entirely new and realistic data instances that align with the underlying data distribution.

The expanded capacity of generative AI offers versatile utility, ranging from simulating subatomic particle interactions in high-energy physics to crafting lifelike images in computer vision. These generative AI models have played pivotal roles in reshaping diverse domains, providing innovative solutions to intricate challenges.

In the context of catalysis and drug discovery - a central topic of this thesis - these models have proven instrumental by accelerating research and development through the prediction of molecular properties and the generation of novel molecular structures. Similarly, in the realm of quantum mechanics, generative AI models are pioneering a deeper understanding and manipulation of quantum states, potentially revolutionizing quantum computing and catalyzing advancements in various scientific domains.

3.2 Detailed Overview of Generative AI Models

This section provides a deep dive into the mathematical frameworks and architectures that constitute the backbone of generative AI, unraveling the intricate methodologies and principles that govern these models and offering an insight into their functioning and potential applications.

3.2.1 Generative Adversarial Networks (GANs)

Introduced by Ian Goodfellow and his team in 2014[12], GANs have emerged as a pivotal tool in the field of generative AI. Operating on the principles of game theory, GANs involve the simultaneous training of two neural networks, a generator, and a discriminator, in a min-max game setting, fostering a dynamic environment of competition and cooperation.

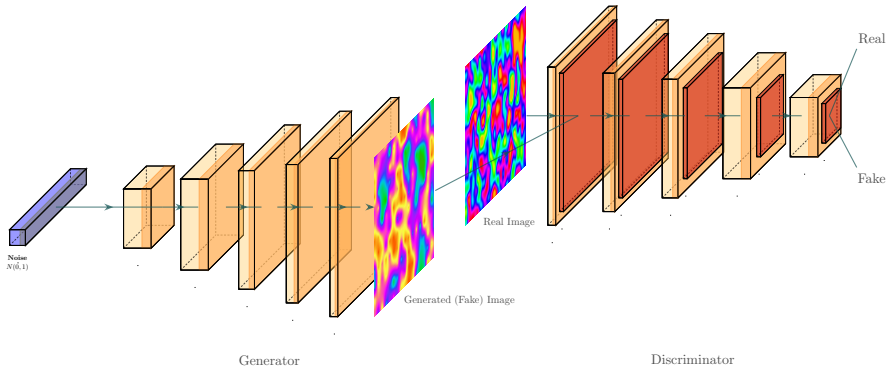


Figure 3.2: GAN Architecture: This image illustrates a Generative Adversarial Network (GAN) setup with a generator, a discriminator, and examples of generated (fake) and real images.

3.2.1.1 Architecture and Components

The GAN architecture is founded on a zero-sum game framework, involving two key components: the generator (G) and the discriminator (D). As illustrated in Figure 3.2, the generator aims to generate data that mimics the real data distribution, while the discriminator strives to distinguish between real and generated data. This adversarial process is mathematically encapsulated in the following min-max objective function:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (3.1)$$

The different elements of this function can be disseminated as follows:

- $\boldsymbol{\min_G \max_D}$**:** These operations represent the minimization and maximization processes, respectively. Within a GAN, the generator strives to minimize this objective function ($\min_G$), while the discriminator endeavors to maximize it ($\max_D$).
- $\boldsymbol{V(D,G)}$**:** This function encapsulates the core objective of the GAN, quantifying how effectively the generator (G) produces data resembling real data and how adeptly the discriminator (D) discriminates between real and generated data.
- $\boldsymbol{E}_{x\sim p_{\mathbf{data}}(x)}[\log D(x)]$: This component corresponds to the expected value computed over real data samples (x) drawn from the genuine data distribution (p_{data}). It assesses the logarithm of the discriminator's ability to correctly identify real data as genuine, essentially measuring the discriminator's skill in recognizing authentic data.
- $\boldsymbol{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$: This element represents the expected value calculated over random noise vectors (z) generated from a prior distribution (p_z). It computes the logarithm of the discriminator's probability of

erroneously classifying generated data as real. In essence, it quantifies the discriminator's proficiency in detecting fake data.

This objective function forms the core of the adversarial training process, with the generator seeking to minimize it, while the discriminator aims to maximize it. This iterative process continues until a Nash equilibrium is reached [13], where neither player can improve their score by changing their strategy.

The training dynamics involve a careful balance between the two networks, ensuring that neither network overpowers the other, which could result in mode collapse, where the generator generates limited varieties of samples.

3.2.1.2 Advancements and Innovations

Since its inception, the GAN framework has witnessed a series of innovations and advancements, fostering a rich ecosystem of diverse architectures and methodologies. Conditional GANs [14], for instance, incorporate additional information to guide the data generation process, allowing for more controlled and targeted data generation.

Cycle-GANs [15] represent another significant advancement, enabling translations between different data domains without the requirement of paired examples. This has opened new avenues for image-to-image translation, style transfer, and data augmentation.

Notable advancements also include the development of Wasserstein GANs (WGANs) [16]. Unlike traditional GANs, which use the Jensen-Shannon divergence to measure the dissimilarity between data distributions, WGANs utilize the Wasserstein distance (also known as the Earth Mover's distance). This change in distance metric offers several advantages, including a more stable training process.

Furthermore, researchers have developed various loss functions and training methodologies to stabilize the training process and enhance the performance of GANs, including techniques like gradient penalty [17] and spectral normalization [18].

Recent trends also indicate a growing interest in integrating quantum principles into GAN architectures, potentially paving the way for the next wave of innovations in the field of generative AI. This thesis aligns with these trends, focusing on the application of quantum principles in catalysis, contributing to the ongoing exploration of this convergence.

3.2.2 Autoencoders and Variational Autoencoders

Autoencoders[19] and their variant, Variational Autoencoders (VAEs) [20], are neural networks designed to reconstruct their input data through a process of compression and decompression. The architecture - shown in Figure 3.3 - comprises two components: an encoder function $f(x)$ and a decoder function $g(z)$, where x is the input data and z is the encoded latent representation. The mathematical objective of an autoencoder is to minimize the reconstruction error, typically represented as a loss function $L(x, g(f(x)))$.

In other words, the encoder's task is to acquire the skill of compressing the input data into an encoded representation, while the decoder is trained to reconstruct the original data from this encoded representation, aiming for the highest fidelity to the original input.

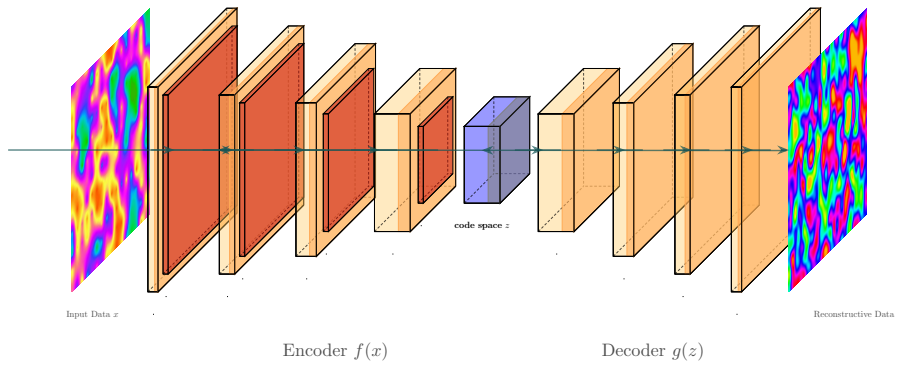


Figure 3.3: Autoencoder Architecture: This image provides a visual representation of an autoencoder, a neural network architecture used for unsupervised learning and dimensionality reduction. Autoencoders aim to encode and decode data, capturing essential features in the process.

VAEs extend the autoencoder framework by introducing a probabilistic layer that models the uncertainty inherent in the data generation process. The loss function in VAEs, which combines a reconstruction loss with a regularization term based on the Kullback-Leibler (KL) divergence, can be represented as:

$$\mathcal{L}(x) = -E_{z\sim q(z|x)}[\log p(x|z)] + \mathrm{KL}(q(z|x)||p(z)) \tag{3.2}$$

In this equation, the first term, $-E_{z\sim q(z|x)}[\log p(x|z)]$, assesses how effectively the VAE can reconstruct the input data x from different latent representations z sampled from the approximate posterior distribution $q(z|x)$. In simpler terms, it measures how well the VAE can recreate the original data from various possible internal representations. The second term, $\mathrm{KL}(q(z|x)||p(z))$, quantifies the difference between the approximate distribution of these representations learned by the VAE and a predefined prior distribution $p(z)$. This encourages the VAE to shape its latent space according to the chosen distribution, typically a standard normal distribution, ensuring a well-behaved and continuous latent space. During training, the VAE seeks to minimize this loss function, improving its ability to generate data resembling the original input while maintaining a structured latent space.

3.2.2.1 Applications and Developments

Autoencoders have found extensive applications in numerous domains such as dimensionality reduction, anomaly detection, and image denoising. Their ability to learn compact representations makes them invaluable tools in data compression and generation tasks.

VAEs, with their probabilistic foundations, have further broadened the scope of applications, enabling the generation of new data samples that are statistically similar to the training data. They are extensively used in image generation, drug discovery, and other domains where generating new data samples is essential.

In recent years, the field has witnessed the development of several advanced VAE architectures, including conditional VAEs [21], which allow for targeted

data generation by conditioning on additional information, and hierarchical VAEs [22], which introduce a multi-layered latent space to capture more complex data distributions.

Other notable advancements worth mentioning include hierarchical VAEs [23], which introduce a multilayered latent space to capture more complex data distributions, and the Beta-VAE [24], which offers a more balanced trade-off between reconstruction accuracy and latent space regularization.

3.2.3 Flow-based Generative Models

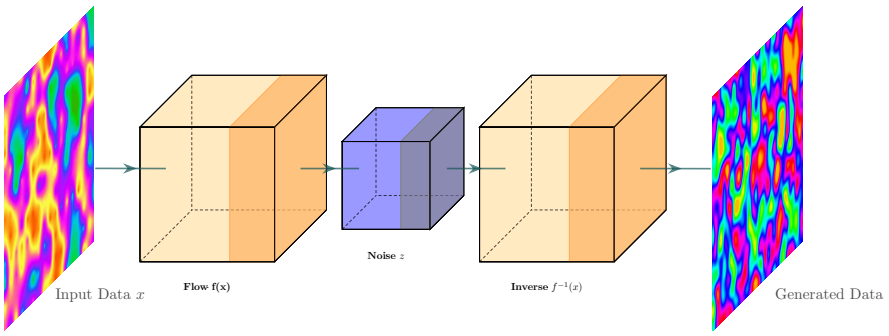


Figure 3.4: Flow-Based Architecture: Image illustrating a flow-based neural network, known for its capacity to model complex data distributions in applications like generative modeling and density estimation."

Flow-based generative models are a distinct category within the realm of generative modeling. Unlike popular approaches such as GANs and VAEs, these models are constructed through a series of invertible transformations. What sets them apart is their explicit learning and modeling of the probability density function of real data using normalizing flows. Normalizing flows, a potent statistical tool for density estimation, underpin the core principle of these models. They consist of a sequence of reversible transformations that map a simple initial distribution, often a Gaussian distribution, to a complex target distribution. This mapping is represented mathematically as:

$$\log p_X(x) = \log p_Z(f(x)) + \log \left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right| \tag{3.3}$$

Here, $\log p_X(x)$ represents the log-likelihood of data x in the data space. $\log p_Z(f(x))$ represents the log-likelihood of the transformed data $f(x)$ in the latent space. The last term, $\log \left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right|$, quantifies the volume change in the data space during the transformation. The determinant of the Jacobian matrix, $\frac{\partial f(x)}{\partial x^T}$, plays a critical role in this likelihood computation.

This mathematical framework offers several advantages, including the capacity for exact likelihood computation, enabling precise assessment of the model's ability to generate data akin to real-world observations. Moreover, it facilitates the direct optimization of the data log-likelihood, meaning it can fine-tune their parameters to enhance their performance in generating data that aligns with

real observations. This direct optimization is advantageous for model training and performance improvement.

Furthermore, an essential component in this process is the determinant of the Jacobian matrix, which quantifies the volume change in the data space during transformations, playing a pivotal role in likelihood estimation and ensuring the fidelity of generated data to the underlying distribution.

3.2.3.1 Advancements and Challenges

Flow-based models have undergone numerous advancements, including the introduction of various types of flows such as Planar Flows, Radial Flows [25], and Glow [26], which offer different approaches to modeling complex distributions. These models have found applications in a variety of domains, including image synthesis, where they can generate high-quality, realistic images.

Despite their potential, these models face certain challenges, including high computational costs and difficulties in modeling discrete data. The necessity to compute determinants of high-dimensional Jacobians often results in increased computational complexity. Moreover, these models sometimes struggle to capture intricate details in data, particularly when modeling complex, multi-modal distributions.

Future research in this domain is directed towards addressing these challenges, potentially leveraging advancements in computational hardware and the integration of quantum principles to enhance modeling capabilities and computational efficiency.

3.2.4 Diffusion Models

Diffusion models, grounded in stochastic processes, represent a unique approach to generative modeling. They model data generation as a diffusion process characterized by the progressive addition of noise over a series of time steps.

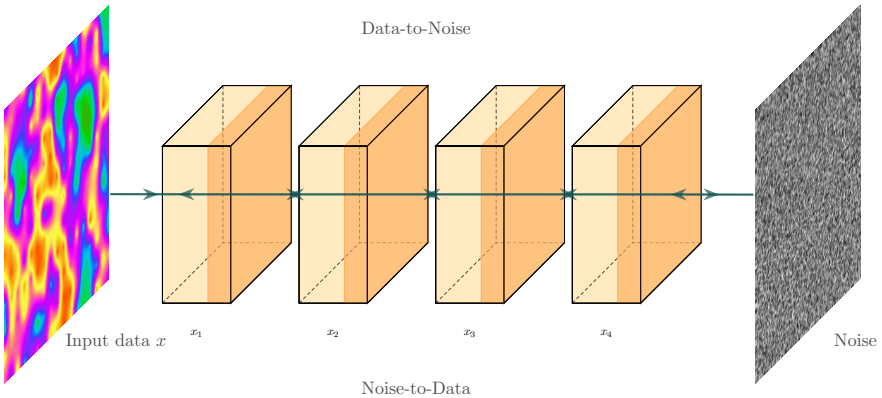


Figure 3.5: Diffusion Model Architecture: This image illustrates a diffusion model, a probabilistic generative model used in deep learning. Diffusion models are designed to capture complex data distributions for tasks like image generation and denoising.

Diffusion models are based on a stochastic differential equation that governs the diffusion process. The data undergoes a series of transformations, modeled as:

$$X_t = X_{t-1} + \sqrt{dt} \cdot \xi_t \tag{3.4}$$

Here, X_t denotes the data at time t, dt is a time step, and ξ_t is a noise term. This framework enables the generation of new data samples through a reverse process, where noise is progressively removed, guiding the data from a noisy state back to a clean, coherent state. This reverse process is governed by a denoising objective, which seeks to learn a score function that is consistent with the data distribution.

3.2.4.1 Applications and Recent Developments

Diffusion models have found applications in various fields, including image synthesis and natural language processing. The ability to model data generation as a diffusion process allows for the creation of complex, high-dimensional data samples.

Recent developments in the field have introduced techniques like denoising score matching and contrastive divergence, which offer novel perspectives in training these models. Furthermore, the field has witnessed the development of variational principles [27] and contrastive objectives, which enhance the capabilities of diffusion models, allowing for more nuanced approaches to data generation.

Future research in this area is poised to explore the integration of diffusion models with other generative modeling approaches, fostering the development of hybrid models with enhanced capabilities in data generation and analysis.

3.3 Conclusion

This chapter has provided an overview of the foundational concepts and contemporary trends within the field of generative AI, emphasizing the mathematical principles that underpin various generative models. The discussion has traced the evolution of GAN architectures and underscored the significant advancements and innovations that define the current landscape of generative AI. Additionally, it has provided descriptions of VAEs, diffusion models, and flow-based models, shedding light on their key principles and contributions. The generative AI ecosystem encompasses numerous other models, including transformers, which play a significant role but are not covered in detail within this section.

As we look to the future, the integration of quantum mechanics principles into generative AI models emerges as an enticing frontier for research and innovation. This convergence holds the potential to usher in a new era of computational capabilities, potentially catalyzing groundbreaking advancements in fields such as physics, engineering, and beyond.

In the next chapter, we will delve into Quantum GANs and explore how they harness the power of quantum computing to further expand the horizons of generative AI. This exploration will focus on the current GAN architectures developed for generative chemistry, providing an in-depth understanding of their applications in molecular design.

Chapter 4

GANs and Quantum Technologies for Generative Chemistry

4.1 Current AI Technologies for Chemistry

In the rapidly evolving domain of chemistry, the quest for efficient catalyst screening has witnessed a significant shift towards the adoption of AI technologies, particularly Graph Neural Networks (GNNs), to predict strain effects on adsorption energy. These networks stand as a robust solution to the inherently high-dimensional search space associated with identifying desirable strain patterns, a process known to be complex and challenging.

GNNs have emerged as a potent tool in predicting the catalyst's adsorption energy response under proposed surface strain patterns with remarkable precision. One of the foremost approaches in this endeavor is the application of Crystal Graph Convolutional Neural Networks (CGCNN) [28]. These networks have exhibited a remarkable accuracy rate, successfully predicting the adsorption energy response for approximately 85% of strains in unseen test data. This success rate signifies a considerable advantage, particularly in identifying the energy barriers associated with small-molecule adsorption energies, a critical parameter in the field of surface chemistry.

Aside from CGCNN, other notable architectures such as SchNet [29] and DimeNet++ [30] have also made substantial strides in this field. These architectures, deeply rooted in graph theory, have demonstrated substantial success in various tasks, including classification and regression. Consequently, they have become a cornerstone in the current trend of employing graph architectures for chemistry applications.

In addressing the challenges and facilitating the prediction of adsorption energies for catalyst discovery and optimization, many of these implementations have leveraged the datasets released by the OCP [9]. These datasets, comprising more than 1.2 million DFT-relaxed catalyst-adsorbate structures, serves as a vital resource in this domain. The extensive data available through the OCP aids in the nuanced understanding and prediction of adsorption energies, playing a pivotal role in the advancement of catalyst discovery and optimization processes. For a comprehensive understanding of the OCP, readers are referred to Chapter 2.2.3 of this thesis, which offers a detailed insight into the project and its significance in the field.

4.2 Current GAN Implementations for Generative Chemistry and its Enhancement with Quantum

GANs have proven to be a powerful tool in the field of generative chemistry, fostering innovations and advancements at an accelerated pace. The synergy between GANs and quantum mechanics is beginning to unfold, showcasing potential avenues to further enhance the capabilities of these networks. In this section, we delve deeper into the current state of the art, offering a comprehensive analysis of the most important implementations in the field of generative chemistry.

4.2.1 MolGAN: An Implicit Generative Model for Small Molecular Graphs

In the domain of generative chemistry, the MolGAN model emerges as a viable alternative to traditional techniques. Distinct from conventional methods that utilize string-based (SMILES) representations for molecules, MolGAN adopts a graph-based framework for depicting molecular structures, as extensively detailed in [31]. This innovation enables a more streamlined and efficient exploration of the intricate domain of chemical synthesis. It is worth noting that MolGAN does not incorporate quantum mechanics in its model.

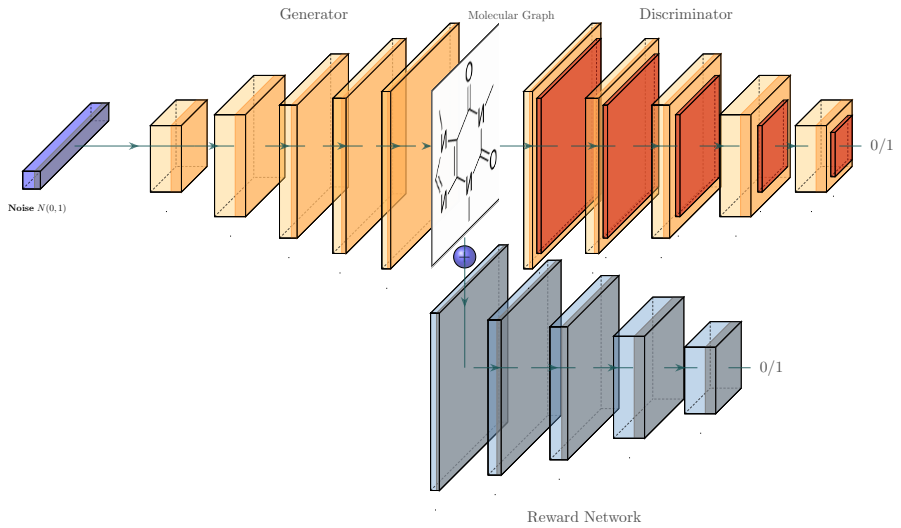


Figure 4.1: MolGAN Architecture: A vector is generated from a prior distribution and fed into the Generator, which then produces a graph-based representation of a molecule. The Discriminator evaluates the origin of the molecular graph, determining whether it is generated by the model or sourced from an existing dataset. Meanwhile, the Reward Network estimates the chemical properties of the generated molecule using external software, providing a reward metric for the system.

4.2.1.1 Data Representation in MolGAN

A critical aspect of MolGAN is the representation of molecular data. Unlike many other generative models that use string-based (SMILES) representations, MolGAN operates directly on graph-structured data. This representation is more native to the molecular structure, encapsulating the relationships between atoms (nodes) and bonds (edges) more naturally and effectively. This approach facilitates the direct generation of molecular graphs, bypassing the need to decode SMILES strings, thereby enhancing the efficiency and accuracy of the generative process.

1. **Generator**: The generator creates new molecular graphs from a random noise vector, utilizing a Graph Convolutional Network (GCN) to sequentially generate nodes and edges, forming a molecular graph.
2. **Discriminator**: This network assesses the authenticity of the molecular graphs produced by the generator, distinguishing genuine graphs from the dataset from those generated, utilizing a GCN to process the graph data and adjudicate their authenticity.
3. **Reward Network**: A vital component of the MolGAN architecture, the reward network evaluates the generated molecules based on predetermined criteria, such as specific chemical properties. The feedback from this network computes the reward for the reinforcement learning objective, guiding the generator to produce molecules aligning with the desired properties.

The training process combines adversarial training with reinforcement learning. The generator aims to deceive the discriminator while maximizing the reward obtained from the reward network. This dual objective function amalgamates the adversarial loss from the discriminator and the reward from the reward network, fostering a more focused and efficient generation process.

The adversarial loss is computed using the binary cross-entropy loss function, while the reward is gauged as the expected reward over the generated molecules, encouraging the generation of molecules with higher rewards.

As described in [31], MolGAN undergoes empirical evaluation using the QM9 chemical database, which is notable for its comprehensive collection of small organic molecules. The data garnered from this evaluation affirm MolGAN's ability to generate molecules with specified properties, suggesting its potential utility as a specialized tool in the field of generative chemistry.

4.2.2 QGAN-HG: Enhancing MolGAN with a Quantum Hybrid Generator

The paper titled 'Quantum Generative Models for Small Molecule Drug Discovery,' referenced in [32], introduces a novel approach in the area of drug discovery through the use of a new quantum generative model, Quantum Generative Adversarial Network with a Hybrid Generator (QGAN-HG). This model aims to navigate the chemical space more efficiently using fewer qubits compared to classical GANs. The traditional drug discovery pipeline often involves lengthy and costly procedures, taking an estimated 5 to 10 years and requiring significant financial investment.

The QGAN-HG model is described as a quantum-enhanced generative approach designed to explore a chemical space that could theoretically contain up to 10^{60} structures. According to the paper, the model has the potential to identify drug candidates more efficiently, which could have implications for the speed and cost of drug discovery.

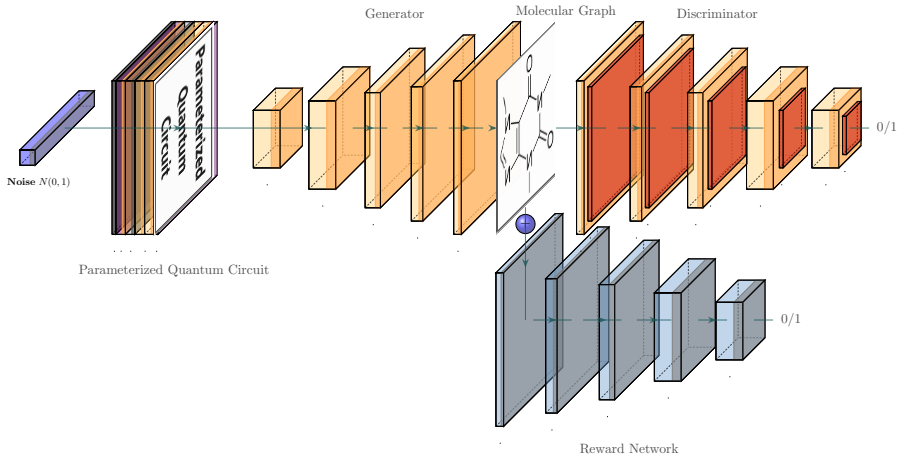


Figure 4.2: QGAN_HG Architecture: MolGAN-based GAN architecture with a parameterized quantum circuit in the generator. It features two stages: a quantum circuit that measures expectation values and a classical discriminator that classifies real vs. synthetic molecules from the QM9 dataset.

4.2.2.1 Architecture and Data Representation

The QGAN-HG model fundamentally builds upon the MolGAN framework, incorporating significant advancements. At its core is a parameterized quantum circuit integrated within the generator. This hybrid quantum generator is capable of supporting a varied number of qubits and quantum circuit layers, working alongside a classical discriminator, which remains identical to the one used in MolGAN.

Remarkably, even with a retention of only 14.93% of the parameters, the QGAN-HG can learn molecular distributions as efficiently as its classical counterparts. This efficiency underscores a quantum advantage in the generative process, promising a richer representation of molecules through a more efficient exploration of the exponentially large chemical space.

The data representation in this model is graph-structured, akin to MolGAN, providing a more direct and native representation of the molecular structures compared to the SMILES strings. This method facilitates a more accurate and direct generation of molecular graphs.

The model faces a significant challenge in terms of the quantum computing resources required. To generate QM9-like small molecules, a full quantum GAN would necessitate over 90 qubits, a capacity not supported by currently available commercial quantum computers. Despite this, the QGAN-HG model demonstrates remarkable efficiency, even with a substantial reduction in the number

of retained parameters.

4.2.2.2 Training Datasets and Cost Function

The QGAN-HG utilizes the QM9 database for training, a well-recognized repository of small organic molecules. However, it is worth noting that reconstructing synthetic molecules similar to those in the QM9 dataset requires over 90 qubits, given the 5 distinct bond and atom types contained within the dataset. This is a significant hurdle given the current limitations of commercially available quantum computers.

As for the cost function, it employs the Wasserstein lost function or Earth-Mover's (EM) distance, which is commonly used in GAN architectures, the precise details should be find from [32].

4.2.3 Enhancing MolGAN with a Hybrid Generator and Discriminator

The paper "Exploring the Advantages of Quantum Generative Adversarial Networks in Generative Chemistry", detailed in [33], represents another significant stride in the field of drug discovery. Building on the QGAN-HG model, it further extends the architecture to enhance the quantum advantage in small molecule drug discovery. Importantly, MolGAN serves as the base model for all implementations detailed in this implementation design, showcasing its pivotal role in the evolution of quantum generative models.

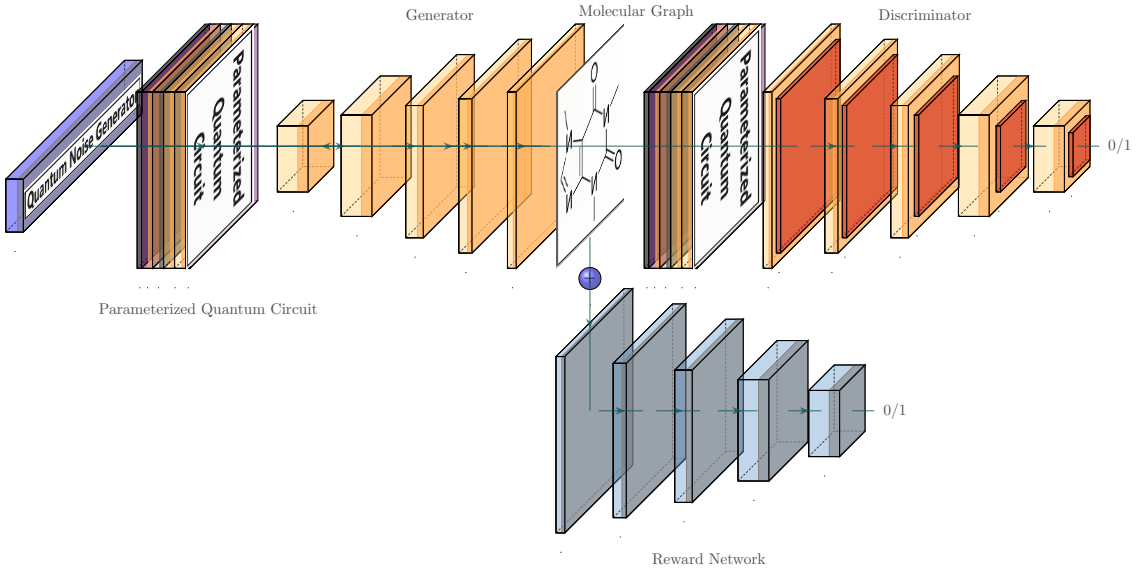


Figure 4.3: Hybrid quantum-classical GAN architecture for small molecule discovery, featuring a quantum noise generator and variational quantum circuits (VQCs) for the generator and discriminator.

4.2.3.1 Technical Aspects and Architectural Innovations

This implementation integrates a Variational Quantum Circuit (VQC) into both the generator and the discriminator, supplemented with a quantum noise generator, thereby amplifying the quantum advantage in the generative process. The extended architecture manifests in several forms, each building upon the foundational MolGAN model:

1. **MolGAN with VQC as the Noise Generator (QuMolGAN):** In this variant, a variational quantum circuit functions as a noise generator, enhancing the exploration efficiency within the chemical space and fostering a richer representation of molecules.

2. **MolGAN with Quantum Generator (MolGAN-QC):** Here, a quantum circuit is integrated within the generator, leveraging the quantum advantage to facilitate a more nuanced exploration of the chemical space.

3. **MolGAN with Quantum Discriminator (MolGAN-CQ):** This implementation incorporates a variational quantum circuit in the discriminator, augmenting its capability to discern the authenticity of generated molecular structures with heightened accuracy.

Notably, this extended architecture omits the reward network, a characteristic feature of the original MolGAN model, marking a shift in the approach towards generating molecules with desired properties.

4.2.4 Heterogeneous GAN Catalyst Design Implementation

The paper "Heterogeneous Catalyst Design by Generative Adversarial Network and First-Principles Based Microkinetics", detailed in [34], introduces a distinctive approach in the realm of catalyst design, utilizing GANs but not quantum methods. Notably, this research deviates from previous implementations as it is not based on the MolGAN model, offering a fresh perspective in the application of GANs in the field of generative chemistry.

4.2.4.1 Technical Aspects and Architectural Innovations

In this study, a Conditional Generative Adversarial Network (CGAN) is implemented to facilitate the generation of new alloy surfaces that are primed for ammonia catalysis. This innovative approach leverages reaction rate—also known as Turnover Frequency (TOF)—values and atomic compositions, which guide the generation process. Here, the CGAN is conditioned upon these parameters, creating potential catalyst designs which are adept at fostering more efficient ammonia synthesis reactions, a process commonly known as the Haber-Bosch process [4].

The architecture of the CGAN encapsulates two primary entities:

1. **Generator:** This component is tasked with crafting new catalyst designs, guided by the input conditions of TOF values and atomic compositions.

2. **Discriminator:** This segment of the network evaluates the catalyst designs generated, distinguishing between authentic and generated data points, thereby steering the generator towards the creation of more realistic and viable catalyst designs.

4.2.4.2 Results Presented in the Paper

The paper discusses the successful generation of new alloy surfaces using a Conditional Generative Adversarial Network (CGAN), which employs first-principles based microkinetic models. Unlike previous models that integrated quantum methods, this research solely utilizes CGAN for creating surfaces optimized for ammonia catalysis. The approach shows promise in advancing the field of catalyst design.

The research underscores the efficacy of utilizing CGAN in creating catalyst designs that are not only viable but also potentially more efficient in facilitating the Haber-Bosch process for ammonia synthesis. This stands as a testament to the potential of implementing GANs in a manner divergent from the established MolGAN-based methodologies, paving the way for more innovative approaches in the field of generative chemistry.

4.3 Challenges and Future Perspectives

The state-of-the-art in Quantum GANs highlights a range of challenges and avenues for future exploration. A key issue centers around the static nature of parameterized quantum circuits. For instance, in current models like [32] and [33], theta parameters are set initially but remain static through subsequent iterations. This limitation effectively demotes the quantum circuit to a simple noise generator post-initialization, thereby missing the opportunity for a synergistic relationship between classical and quantum computations. The static nature of these parameters hampers the optimization process and restricts the search for new and energetically favorable molecular configurations.

Scalability remains another significant challenge. While the theoretical capabilities of quantum mechanics could revolutionize generative chemistry, existing quantum hardware fall short of scaling up to handle complex molecular structures in real-world scenarios. This is a critical issue, especially considering that the computational demands for simulating larger molecules grow exponentially, thus setting back the broader industrial application of Quantum GANs.

In terms of integrating quantum mechanics with AI, the process is far from straightforward. Quantum computations, characterized by intrinsic phenomena such as superposition and entanglement, necessitate a distinct computational framework that diverges from classical paradigms. This incongruity presents significant hurdles for seamless integration between these disparate technologies and complicates the development of algorithms capable of harnessing the unique strengths of each domain.

In the following section, we will introduce a new Quantum GAN model designed to address these challenges currently faced by state-of-the-art GANs. This model aims to offer solutions that could make quantum-enhanced generative chemistry more scalable, compatible, and dynamically optimized.

Chapter 5

Implementing a Quantum GAN for Cu-based Binary Alloy Catalyst Generation

In recent years, the catalysis community has witnessed a substantial surge in research initiatives centered around copper (Cu)-based binary alloy catalysts. This heightened interest is primarily driven by the groundbreaking discoveries that pinpoint the exceptional efficacy of these catalysts in carbon dioxide reduction processes. The fusion of machine learning methodologies with empirical experiments has further propelled these discoveries, paving the way for novel and efficient solutions in the field of catalysis.

Building on this momentum, the research conducted in this thesis seeks to utilize the transformative capabilities of Quantum GANs to facilitate the creation of Cu-based binary alloy catalysts. The objective is to create a symbiotic relationship between classical and quantum computing paradigms to craft a sophisticated model capable of generating new binary Copper alloy catalysts with optimal properties.

In this venture, we utilize a rich dataset extracted from the OCP [9], which encapsulates a diverse range of binary copper alloy catalysts denoted as Cu_xM_{1-x}, where M represents an alloying element, in conjunction with 27 different adsorbates, laying a robust foundation for our strain investigation. The Quantum GAN model, implemented through PyTorch [35] and PennyLane [36], operates in a hybrid fashion, amalgamating classical architecture with quantum components, thereby unveiling new horizons in catalyst generation.

The upcoming sections outline the intricate structure of our implementation [37], detailing both the classical architecture without quantum components and the quantum architecture that includes a quantum noise generator and a parameterized quantum circuit, and the patch architecture which utilizes quantum GAN methods to construct the final result through the aggregation of several sub-generators.

Furthermore, we delve into the quantum noise generator and the parameterized quantum circuit, elucidating their roles in the generation process. Additionally, we discuss the optimization strategies employed, focusing on the cost function and the evaluation metrics that guide the fine-tuning of the model.

5.1 Structure of the Implementation

The Quantum GAN implementation in this work [37], which combines the power of PyTorch and PennyLane, strides into this domain with the goal of streamlining the creation of novel Cu-based binary alloy catalysts. By harnessing the capabilities of quantum computing, it elevates the performance of generative

adversarial networks, opening doors to more precise and efficient catalyst generation.

In this context, PyTorch serves as a versatile deep learning framework that provides the essential tools for neural network design and training. On the other hand, PennyLane stands as a quantum machine learning library, enabling seamless integration of quantum computing elements into our implementation.

Our implementation offers three distinct architectural options, each configurable to cater to specific needs.

- **Classic Architecture:** Our classic architecture forms the foundational layer of the Quantum GAN model. It is based on the MolGAN architecture [31] but does not incorporate a reward network. This architecture operates entirely within the classical computing domain and serves as the foundation upon which quantum components are integrated. It ensures seamless integration of classical and quantum computing principles.

- **Quantum Architecture:** The quantum architecture incorporates a quantum noise generator and a parameterized quantum circuit specifically within the generator component of the model. These elements are compatible with both IBM hardware devices and simulators, aiming to improve the model's ability to generate complex Cu-based binary alloy catalysts.

- **Quantum Patch Architecture:** The patch architecture employs a technique where the final output is constructed by concatenating multiple smaller patches, called sub-generators, which are all identical and of the same size.

  This original patch technique was first introduced in reference [38]. They presented a quantum patch GAN mechanism to make efficient use of limited qubits for generating hand-written digit images. More recently, this technique has found its way into the realm of generative chemistry, where several Quantum GAN models have adopted it [33, 32].

  One of the standout advantages of the patched technique is its inherent efficiency. It requires fewer quantum resources and permits the execution of multiple sub-circuits sequentially. Furthermore, this technique facilitates more efficient simulation of each circuit, dramatically expediting the learning process. These advantages are of paramount importance, especially when dealing with hardware constraints, such as a limited number of available qubits.

  Recognizing its manifold advantages, we have chosen to integrate this technique into our own implementation. In our implementation, the set of sub-generators replicates the parameterized quantum circuits described in Figure 5.5. These sub-generators may either scale down, scale up, or maintain the number of qubits compared to the circuits depicted in Figure 5.5. The model accommodates configurations with 1, 2, or 4 sub-generators (also called patches), all with the goal of maximizing the precision and accuracy of catalyst structure generation.

Figure 5.1 provides an illustrative example of the patch method. The figure showcases the utilization of two parameterized Quantum Circuits with 2 qubits each, which serve as sub-generators. Each of these sub-generators is responsible for generating a partial segment of the catalysis reaction. After completion, the generated segments are seamlessly concatenated together, providing a comprehensive representation of the reaction.

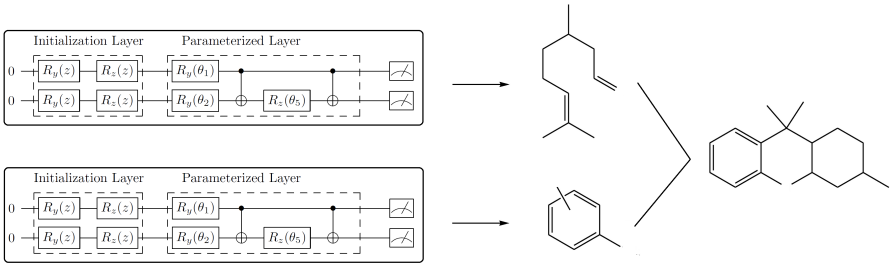


Figure 5.1: Patch Method: It employs multiple Parameterized Quantum Circuits as sub-generators. Each sub-generator produces a partial segment of the final catalyst reaction.

Our Quantum GAN model distinguishes itself through extensive configurability, enabling users to fine-tune various parameters to meet specific requirements. These options encompass training configurations, activation function parameters, dropout rates, and quantum circuit settings. Notably, quantum circuit configurations include the choice of 1, 2, or 4 patches, providing users with flexibility to control the level of complexity and detail in the generated catalysts. For a comprehensive list and detailed descriptions, please refer to Appendix A in this thesis.

To facilitate this configurability, our implementation allows users to specify and customize these parameters. This level of flexibility ensures that our Quantum GAN implementation serves as a versatile and adaptable tool for Cu-based binary alloy catalyst generation, empowering researchers to achieve their desired outcomes.

The subsequent portions of this section delve into the architectural intricacies and the constituent components that form the bedrock of our Quantum GAN model, shedding light on how these components collaborate to achieve our catalyst generation objectives.

5.2 Catalyst Reaction Representation

One of the most significant challenges encountered in this project was determining the most suitable representation for the catalyst reaction within the model. The choice of representation is instrumental as it serves a dual purpose: it provides a structured way to encode the complexities of the catalytic process, and it significantly influences the model's proficiency in generating new catalytic reactions. This is a particular point of concern because, unlike other

chemical processes that have several well-established methods of representation, there is a noticeable absence of a standardized, universally-accepted method for portraying catalytic reactions.

In an effort to fill this gap, we embarked on an exhaustive survey of existing techniques and methodologies used in the state-of-the-art representation of molecules [31, 32]. Among these, two approaches stood out for their widespread acceptance and utility in the domain of Generative Adversarial Networks (GANs): SMILES (Simplified Molecular Input Line Entry System) and graph-based representations.

- **SMILES**: The Simplified Molecular Input Line Entry System, or SMILES, is a text-based, string representation method that efficiently encodes the structural and bonding information of a wide array of chemical species. One of its most compelling features is its human-readable format, coupled with its compactness, which has led to its extensive adoption in various chemical domains.

- **Graph-based Representation**: The graph-based representation offers a visually and structurally organized method for portraying molecular configurations. In this approach, atoms within the molecule are represented as nodes, each tagged with an integer that corresponds to their atomic number. The chemical bonds connecting these atoms are illustrated as edges in the graph and are further detailed in a bond matrix, which categorizes the bonds into types such as single or double. This tailored representation is particularly advantageous for machine learning algorithms like Graph Neural Networks (GNNs), which are designed to natively interpret and analyze graph-based data structures.

Figure 5.2 provides an illustration of these representation techniques using a formaldehyde molecule (CH_2O) as an example, depicted both in graph-based and SMILES notation forms. In the graph-based representation, atom types for Carbon (C) and Oxygen (O) are encoded with their respective atomic numbers 6 and 8, as found in the periodic table. Bond types, such as single and double bonds, are denoted by the integers 1 and 2, respectively, within the bond matrix.

5.2.1 Our Approach to Catalytic Reaction Representation

Since our primary focus was to represent catalytic reactions, we embarked on designing a novel approach representation that could effectively encapsulate a catalytic reaction in a simplified manner. Our approach draws inspiration from the graph-based representations commonly employed for molecules. In our custom-designed representation, we identify three essential components that, in our view, constitute the fundamental elements of a catalytic reaction: the composition of the alloy catalyst, the adsorbent molecule involved, and the quantifiable energy associated with the reaction.

We use separate arrays to encapsulate the information of each component involved in the catalytic reaction. For the alloy catalyst and the adsorbent molecule, each element in the respective array corresponds to an individual atom, which is identified by its atomic number as specified in the periodic table. As a deliberate design choice, we have chosen to omit the inclusion of bond

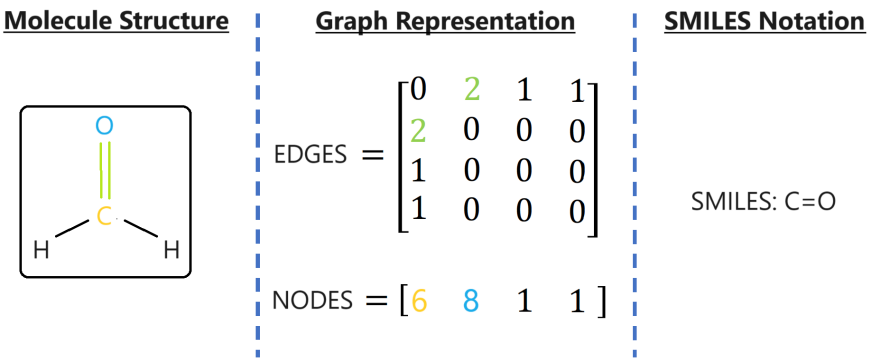


Figure 5.2: Representation of the Formaldehyde Molecule: This figure illustrates the formaldehyde molecule through three different formats: a 2D structural model, a graph-based representation, and SMILES notation. In the graph representation, atom types for Carbon (C) and Oxygen (O) are encoded with their atomic numbers 6 and 8, respectively, as per the periodic table. Bond types are denoted by integers in the bond matrix: 1 for single bonds and 2 for double bonds. The SMILES notation provides a text-based shorthand for the molecular structure.

types between atoms, considering this information to be less pivotal during the initial phases of exploring new catalytic reactions.

On the other hand, the energy aspect of the catalytic reaction is represented by a single-element array. This element is coded with a value between 0 and 3, each representing a distinct energy range for the catalytic reaction, quantified in electronvolts (eV).

In summary, our novel representation framework consists of three distinct arrays:

- **The Binary Copper Alloy Surface Array**: This array conveys information about the composition of the alloy. Given that the alloy under consideration is binary, this array comprises elements for only two types of atoms: Copper, represented by its atomic number 29, and another element from the periodic table.
- **The Adsorbent Array**: This array provides a representation of the adsorbent actively participating in the catalytic process. It includes elements that correspond to the adsorbent molecule, with each element representing the atomic number of a specific atom in the molecule.
- **The Energy Array**: This single-element array serves as a categorical representation of the reaction's energy dynamics, with four categories:
  - **Category 0**: Energy level less than -2 eV.
  - **Category 1**: Energy level between -2 eV and 0 eV.
  - **Category 2**: Energy level ranging from 0 eV to 2 eV.
  - **Category 3**: Energy level greater than 2 eV.

Each of these categorical representations corresponds to a specific range of energy levels, measured in electronvolts (eV). This approach provides detailed insights into the energy dynamics involved in the reactions, thereby facilitating a detailed analysis of the catalytic processes within the model. These categorical representations are crucial in characterizing various aspects of the catalyst reaction and contribute to a deeper understanding of the overall catalytic process.

5.3 Components of the Architecture

In this section, we will provide an overview of the various components that constitute our architecture. These components include the discriminator, generator, cost function used in both generator and discriminator, and the database used for training the network, are illustrated in the overview Figure 5.3.

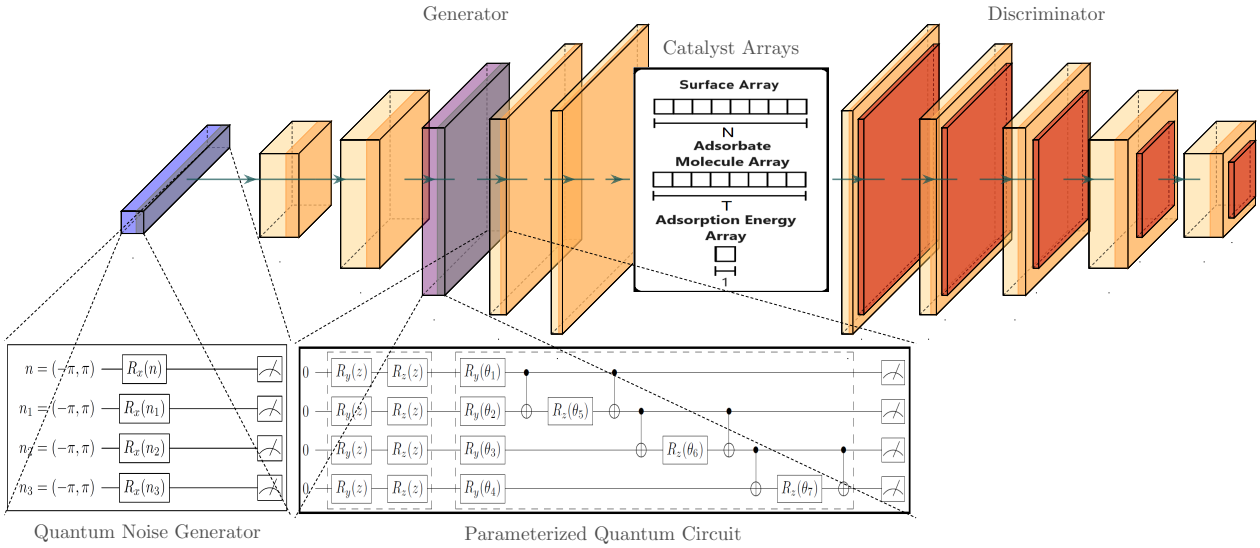


Figure 5.3: GAN Framework for Copper-Based Binary Alloy Catalysis: Incorporating a Parameterized Quantum Circuit in the Generator Network and Utilizing a Quantum Noise Generator.

5.3.1 Generator

The generator is a crucial component of our implementation, designed to create three distinct outputs that are vital in the simulation of Cu-based binary alloy catalysis reactions. These outputs are the **Binary Copper Alloy Surface array**, the **Adsorbent array**, and the **Energy array**, described previously.

Each of these outputs is generated through a separate, yet structurally similar network embedded within the generator component. These networks are characterized by the following layer structure:

1. **Linear Layer**: Responsible for linear transformations of the input data, this layer acts as the entry point of the network, channeling the input data (either random data for the classical mode or quantum data for the quantum mode) through a series of transformations to facilitate the recognition of complex patterns at subsequent layers.

2. **Batch Normalization Layer**: Following the linear layer, this layer normalizes the batch of inputs, ensuring a consistent and stable distribution of data, a fundamental prerequisite for the smooth and efficient training process.

3. **LeakyReLU Activation Function**: Post-normalization, the inputs pass through the LeakyReLU activation function, a function that allows a small, positive gradient when the input is negative, promoting a more efficient training process by accommodating a broader spectrum of input values.

4. **Quantum Layer**: Situated centrally within the network, this layer signifies the heart of our quantum integration. Unlike conventional architectures where the quantum layer often resides at the beginning, integrating it in the middle allows for a seamless blend of classical and quantum computations, enhancing the network's ability to generate more nuanced and complex patterns. This layer is parameterized by the LeakyReLU's output and the number of patches, and is configured based on the quantum setup defined for the implementation. In the following sections of this chapter, we will delve deeper into the details of this layer.

5. **Another Linear Layer and Sigmoid Activation Function**: As the data exits the quantum layer, it passes through another linear layer, which is tailored to generate values specific to each output array (surface, adsorbent, or energy). Following this, a sigmoid activation function is applied to constrain the output values within a range of 0 to 1, finalizing the data for output.

This structural configuration of the network within the generator, especially the central integration of the quantum layer, stands as a novel approach in the domain of hybrid quantum-classical computations, promising more complex and nuanced data generation capabilities.

5.3.2 Discriminator

The discriminator is another vital component of our implementation, functioning as a binary classifier that discerns the real data points from the fake ones generated by the generator. Its role is pivotal in the training process, as it guides the generator to produce more realistic data over time. The structure of the discriminator's neural network is as follows:

1. **Linear Layer**: Serving as the initial layer, it receives input data, which includes the three arrays generated by the generator: surface, absorbent, and energy. This layer performs linear transformations on the input data, preparing it for subsequent processing.

2. **LeakyReLU Activation Function**: After the linear layer, the data is passed through a LeakyReLU activation function. This function, characterized by a negative slope, allows a small gradient even when the input is negative, promoting more nuanced learning during the training phase.

3. **Dropout Layer**: To prevent overfitting and promote a more generalized model, a dropout layer is incorporated. It randomly sets a fraction of the input units to 0 during training, which helps prevent overfitting.

4. **Another Set of Linear, LeakyReLU Activation Function, and Dropout Layer**: Following the dropout layer, the network consists of another set of Linear, LeakyReLU activation function, and dropout layers, enhancing the network's ability to learn from the data by adding complexity to the model.

5. **Output Layer with Sigmoid Activation Function**: The final layer is an output layer with a single neuron, coupled with a sigmoid activation function. This setup ensures that the output of the discriminator is a probability score between 0 and 1, indicating the likelihood of the input data being real.

This structure signifies a classical architecture, skilled at analyzing the data generated by the generator and guiding the training process through its binary classification. Its synergistic interaction with the generator lies at the heart of the model, fostering an environment where the generator continually evolves to produce more realistic data, guided by the feedback from the discriminator.

5.3.3 Quantum Components

The integration of quantum components into the architecture introduces a new dimension of complexity and potential to the GAN model. These components, leveraged through the PennyLane library, facilitate the exploration of quantum phenomena to enhance the machine learning process. Here, we detail the two significant quantum components incorporated in the implementation: the Quantum Noise Generator and the Parameterized Quantum Circuit.

5.3.3.1 Quantum Noise Generator

The Quantum Noise Generator serves as a potent tool in the generation process, specifically in the quantum mode. This generator employs a quantum node, wherein a single qubit undergoes a rotation through an R_x gate, followed by a measurement in the Pauli-Z basis. The output of this process is a noise profile, characterized by a uniform distribution ranging from $-\pi$ to π. This noise profile acts as the input to the generator component, fostering a rich diversity in the generated data, which is vital for the training process. An example of a 4-qubit circuit illustrating these concepts is shown in Figure 5.4.

The method within the Quantum Noise Generator can be outlined as follows:

1. R_x **Rotation**: The R_x gate performs a rotation around the X-axis of the Bloch sphere. The transformation it effects on a qubit state $|\psi\rangle$ is given by the unitary operator:

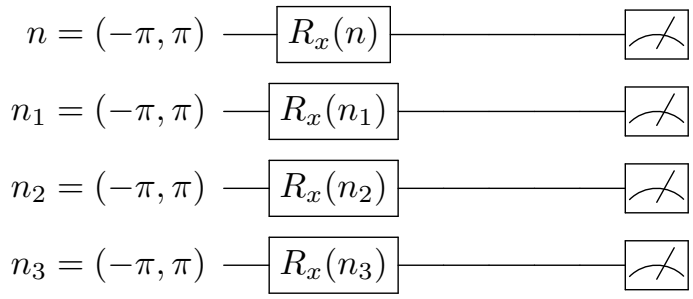


Figure 5.4: Quantum Noise Generator: A 4-qubit quantum circuit with a uniform distribution input ranging from π and $-\pi$.

$$R_x(\theta) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

Here, θ is the rotation angle, which is chosen from a uniform distribution input ranging from $-\pi$ to π. This introduces a quantum noise element into the system.

2. **Pauli-Z Measurement**: After the rotation, a measurement is taken in the Pauli-Z basis. The Pauli-Z operator is represented as:

$$\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

The expectation value of this measurement, given a state $|\psi\rangle$, can be calculated as $\langle\psi|\sigma_z|\psi\rangle$, providing a scalar output that constitutes the quantum noise profile.

The Quantum Noise Generator, by introducing an element of randomness and leveraging quantum characteristics within the input data, aligns with the fundamental probabilistic nature of quantum mechanics. This synergy sets the stage for the generator to produce more nuanced and diverse outputs.

5.3.3.2 Parameterized Quantum Circuit

The Parametrized Quantum Circuit (PQC), depicted in Figure 5.5, is integrated within the middle of the neural networks of the generator, a strategic placement that is slightly unconventional in hybrid quantum-classical models. This placement allows for a richer interplay between classical and quantum computations, potentially harnessing the best of both realms.

The structure of the PQC is as follows:

1. **Initialization Layer**: The initialization layer is encoded into the quantum state through sequences of R_y and R_z gates providing rotations along the

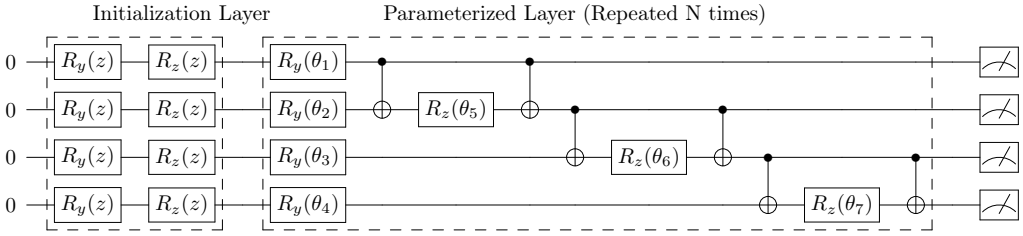


Figure 5.5: Parameterized Quantum Circuit: The circuit comprises an initialization layer with Ry and Rz gates, followed by a 4-qubit ansatz circuit with parameterized gates (Ry and Rz). The parameters (θ_k) of the gates can be learned through back-propagation. The measurement calculates the expected value of each qubit.

y-axis and z-axis respectively. The choice of using R_y and R_z gates in the initialization layer allows for flexibility in preparing quantum states. By adjusting the rotation angles (parameters) of these gates, a high degree of expressiveness can be achieved. The unitary operators for these rotations are given by:

$$R_y(\phi) = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right) & -\sin\left(\frac{\phi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right) & \cos\left(\frac{\phi}{2}\right) \end{bmatrix}, \; R_z(\phi) = \begin{bmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{bmatrix}$$

Here, ϕ represents the data inputs, which are utilized as parameters in these rotations, facilitating the encoding of classical data into quantum states.

2. **Parameterized Layer**: The parameterized layer evolves through a series of quantum gates parameterized by weights.

Our architectural choice for the parameterized quantum layer, incorporating R_y, R_x, and CNOT gates, is designed to serve specific functions within our quantum algorithms. The initial R_y gates play a pivotal role in generating superpositions of quantum states, each with varying amplitudes. In contrast, the Controlled-NOT (CNOT) gates serve as entangling elements, establishing entanglement connections across all qubits within the circuit. Additionally, the R_z gates are responsible for precisely adjusting the phase of individual qubits, a crucial factor that can significantly impact interference patterns.

To aid in reminding, the following matrix illustrates the representation of the CNOT gate:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The parameterized layers, which can be repeated N times, allow for the learning of the parameters θ through back-propagation

3. **Expectation Values**: The final stage involves measuring the expectation values of the Pauli-Z observables, given by $\langle\psi|\sigma_z|\psi\rangle$ for each qubit, which

constitute the output of the PQC. These outputs, then integrated into the classical neural network, carry quantum correlations and properties, enhancing the learning capabilities of the model.

Because the architecture also incorporates the patching method, it can utilize multiple PQC (1, 2, or 4 in our implementation). Each sub-generator is tasked with constructing a partial section of the input. The final output is obtained by concatenating the outputs from all the patches. This adaptability to operate with 1, 2, or 4 circuits adds flexibility to the model, enabling the extension of the circuit when dealing with a limited number of qubits, effectively increasing its versatility.

This integration of quantum components within the classical architecture of the GAN not only enhances its computational capabilities but also opens up new avenues for exploring quantum phenomena in the domain of machine learning and catalysis generation.

5.3.4 Loss Function

In the training process of Quantum GANs, the loss function plays a pivotal role in optimizing the performance of both the generator and the discriminator.

The optimization strategy in the training process of GANs resembles a two-player minimax game. Here, the generator aspires to minimize its loss function L_G, aiming to fool the discriminator, while the discriminator strives to minimize L_D, focusing on correctly classifying real and fake data. This optimization problem can be mathematically portrayed as:

$$\min_G \max_D V(D,G) = E_{x\sim p_{\text{data}}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

In this expression, E denotes the expectation, $p_{\text{data}}(x)$ symbolizes the real data distribution, and $p_z(z)$ depicts the noise distribution, encapsulating the core optimization strategy in the Quantum GANs training process. Here, we delve into the intricacies of the loss functions utilized for both components, and the optimization strategy employed.

5.3.4.1 Generator Loss Function

The generator loss function is an essential component in training the GAN, tasked with measuring how well the generator can deceive the discriminator. The process is detailed in the Python implementation of the code [37]. The loss function operates through the following steps:

1. **Fake Data Generation**: The generator G creates fake data instances, which are represented as $G(z)$, where z is a noise vector. This noise can either be classical random data or quantum data, depending on the mode of operation. This step is depicted in the Python code as:

```
# Step 1: Generate fake data using the generator
fake_catalyst_data = G(z)
```

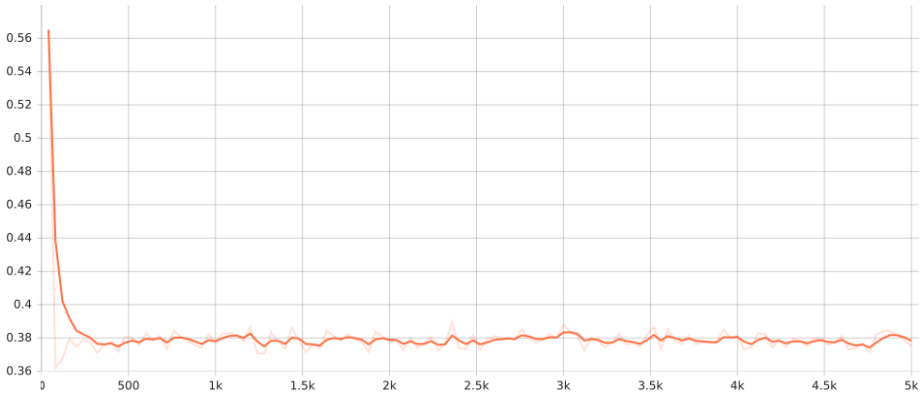


Figure 5.6: Example of Generator Loss during GAN Training. Training over 5000 Epochs. The graph demonstrates that the generator loss rapidly stabilizes to a value of 0.38 within the initial 200 epochs. This quick stabilization suggests that the generator is effectively learning to generate data that closely mimics the real data distribution, as indicated by the low and stable loss value.

2. **Post-processing**: Following the generation of data, it undergoes post-processing methods such as Gumbel softmax to refine the data instances for discriminator evaluation. This is illustrated in the code as:

```
# Step 2: Post-process the generated data
processed_fake_data = postprocess(fake_catalyst_data,
↪  method='Gumbel softmax')
```

3. **Discriminator Evaluation**: The discriminator D evaluates the fake data instances, providing scores $D(G(z))$ which quantify the perceived realism of the fake data. This stage is implemented in the Python code as:

```
# Step 3: Evaluate the fake data using the discriminator
discriminator_scores = discriminator(processed_fake_data)
```

4. **Loss Calculation**: The loss is calculated as the Binary Cross Entropy loss between the discriminator's scores for the fake data and an array of ones. The value of one typically represents the discriminator's high confidence that the input is real data. Mathematically, it is given by:

$$L_G = -\frac{1}{N}\sum_{i=1}^{N}\log(D(G(z_i)))$$

Here, N is the batch size, and z_i represents the noise vector for the i-th data instance. This step is illustrated in the Python code as:

```
# Step 4: Calculate the loss
target_labels = torch.ones_like(discriminator_scores)
generator_loss = loss_function(discriminator_scores,
↪  target_labels)
```

The generator endeavors to minimize this loss, enhancing its ability to deceive the discriminator.

Figure 5.6 is an example showing how the generator loss values stabilize rapidly within the first few hundred iterations over a training course of 5000 epochs. This rapid stabilization indicates the effectiveness of the generator's learning process in producing data that closely mimics the real data distribution.

5.3.4.2 Discriminator Loss Function

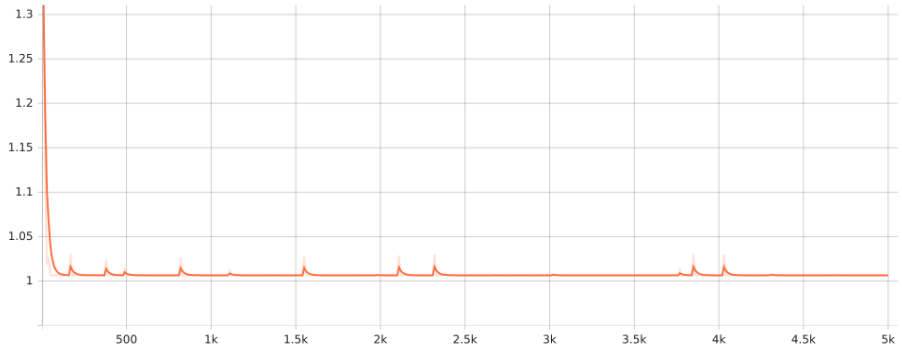


Figure 5.7: Example of Discriminator Loss during GAN Training. The graph illustrates the convergence of the discriminator loss to a value of 1 within the first few hundred epochs, maintaining stability throughout the 5000-epoch run. This rapid stabilization may indicate effective discrimination between real and generated samples

The discriminator loss function aims to optimize the discriminator's ability to correctly classify real and fake data. The process is delineated through its Python code implementation [37], involving the following steps:

1. **Real Data Evaluation**: The discriminator evaluates real data instances, generating scores $D(x)$, where x denotes real data instances. This is highlighted in the Python code snippet as:

```
# Step 1: Get the real data
real_catalyst_data = get_real_data()
```

2. **Fake Data Generation and Evaluation**: Similarly to the generator loss function, fake data instances are generated and evaluated by the discriminator, yielding scores $D(G(z))$. This is represented in the code snippet as:

```
# Step 2: Generate and evaluate fake data
fake_catalyst_data = G(z)
```

3. **Loss Calculation**: The loss for the discriminator is determined as the sum of the Binary Cross Entropy losses for both the real and fake data. Mathematically, it is represented as:

$$L_D = -\frac{1}{N}\sum_{i=1}^{N}(\log(D(x_i)) + \log(1 - D(G(z_i))))$$

Here, N is the batch size, x_i denotes the i-th real data instance, and z_i signifies the noise vector for the i-th fake data instance. This process is implemented in the Python code as:

```
# Step 3: Calculate the loss
labels_for_real_data = torch.ones(batch_size, 1)
labels_for_fake_data = torch.zeros(batch_size, 1)
loss_for_real_data =
↪  loss_function(discriminator(real_catalyst_data),
↪  labels_for_real_data)
loss_for_fake_data =
↪  loss_function(discriminator(fake_catalyst_data),
↪  labels_for_fake_data)

# Step 4: Combine and backpropagate the losses
discriminator_total_loss = loss_for_real_data +
↪  loss_for_fake_data
```

These losses are pivotal in adjusting the weights of the generative and discriminative models.

Figure 5.7 serves as an example illustrating how the discriminator loss behaves after training our implementation for 5000 epochs. The loss rapidly converges to a value of 1 within the first few hundred epochs, indicating effective discrimination between real and generated samples.

5.3.4.3 Selection of an Appropriate Loss Function

The exploration to identify the most appropriate cost function for our project involved an in-depth assessment of various functions, with a focus on both their performance and computational requirements. One of the initially considered options was the Wasserstein GAN with Gradient Penalty (WGAN-GP) [16], a variant of the standard GAN framework.

The Wasserstein GAN, often referred to as WGAN, is a type of Generative Adversarial Network that introduces the Wasserstein distance (also known as the Earth Mover's Distance) as a new metric to measure the difference between the distribution of real data and generated data. It offers certain advantages over the traditional GAN, such as improved stability in training and a more meaningful loss metric. WGAN-GP further enhances the stability of WGAN by adding a gradient penalty term to the loss function, which helps prevent mode collapse and encourages the generator to produce diverse and high-quality samples.

However, despite its potential benefits, our experimentation with WGAN-GP revealed that it significantly prolonged the training process. This led us to explore alternative cost functions that could provide a more time-efficient training process without compromising the reliability of performance.

5.3.5 Training Database

In the endeavor to implement the Quantum GAN for generating Cu-based binary alloy catalysts, the primary requirement was to find an extensive and reliable dataset that encapsulates the necessary characteristics of these catalysts. Our implementation leverages the rich data repository provided by the **Open Catalyst Project (OCP)** database [1], a massive storage encompassing over 100GB of data, which has been instrumental in fueling various research projects in the catalysis community. This section sheds light on the data extraction and preprocessing methodologies, emphasizing the significance of the chosen dataset in the context of our project [37].

5.3.5.1 Open Catalyst Project Database

The Open Catalyst Project database is a comprehensive repository that harbors a wealth of information pertaining to various catalysts and their properties. The database is particularly noted for its substantial storage size, exceeding 100GB, which facilitates the inclusion of a diverse range of data. This vast repository has been crucial in sourcing the training data for our implementation, chiefly because it houses the energy structures required to access the Density Functional Theory (DFT) data.

For our project, we have particularly focused on the Initial Structure to Relaxed Energy (IS2RE) dataset available within the OCP database [1]. This dataset is pivotal as it encompasses the input initial structures alongside the output relaxed structures and energies, stored meticulously as an LMDB file. The primary constituents of this dataset include:

- **y_init**: Denotes the initial energy structure of the system, providing a baseline representation of the catalysts before any reactions or processes.
- **y_relaxed**: Represents the energy of the system once it has reached a relaxed state, offering insights into the potential energy changes during reactions.
- **pos_relaxed**: Furnishes the Cartesian coordinates (x, y, z) that depict the relaxed positions of all atoms within the system, facilitating a detailed analysis of the structural transformations.

5.3.5.2 Data Extraction and Preprocessing

Our data extraction process was precisely tailored to focus on binary Cu alloy catalysts with adsorbates. This curated dataset was extracted from the OCP collection and contains information regarding the adsorption energies of critical small-molecule adsorbates on various Cu-binary alloy surfaces. The increased interest within the catalysis community in Cu alloys, particularly after the discovery of Cu-based catalysts performing exceptionally well in carbon dioxide reduction, motivated this specific focus.

To streamline the preprocessing of data obtained from the OCP collection, we developed a Python script, which is available in our implementation [37]. This script simplifies the extraction and organization of relevant data, ensuring an efficient training process. As a result, we successfully collected data related to 5400 binary Cu alloy catalysts, including their respective energy values. This

extracted dataset formed the cornerstone of our model training, enhancing the robustness and knowledge-driven aspects of our learning process.

Through this meticulous data extraction and preprocessing phase, we have established a strong foundation for the effective implementation and optimization of our Quantum GAN.

5.4 Generative Model Evaluation Methodology

In the evolving field of GANs, Frechet Inception Distance (FID) has traditionally been the benchmark metric for gauging the quality of generated outputs. Our investigations, however, reveal that FID excels in the domain of image data but is less effective when applied to molecular structures. To address this gap, we have turned to the Euclidean distance metric, also known as the L2 norm, as an alternative measure for evaluating the quality of the molecules we generate. This metric undergoes careful normalization to temper the impact of outliers, making it particularly apt for assessing complex chemical data structures.

It is crucial to distinguish between the roles of the loss function and performance evaluation metrics in our model. The loss function serves as a key component, zeroing in on the discrepancies between synthetic and real data. It acts as a rigorous evaluator, guiding the model towards more faithful data generation. In contrast, the Euclidean distance specializes in quantifying the similarities between the generated and original datasets, with particular emphasis on attributes like absorbent, surface, and energy arrays.

In summary, the loss function governs the general trajectory of data generation, while the Euclidean distance provides a nuanced lens for evaluation, especially pertinent for molecular structures. This multi-metric approach lays down a comprehensive framework for assessing our model's performance. It signifies a departure from the widespread reliance on FID in existing GANs, a metric that is better tailored for image-based applications.

5.5 GAN Hyperparameter Tuning

In GANs, hyperparameters are configurable settings that are not derived from training data. Hyperparameters, which must be set prior to training, include parameters such as the learning rate, dropout rate, and various architectural elements. These hyperparameters play a critical role in influencing both the training process and the final model performance. Incorrectly configured hyperparameters may result in suboptimal performance, overfitting, or extended training durations. Therefore, hyperparameter tuning is a necessary step in optimizing GANs for high performance.

For this optimization task, we utilized Optuna, a framework specifically designed for efficient hyperparameter optimization. Optuna [39] automates the selection of optimal hyperparameter combinations, aiming to improve model accuracy and computational efficiency.

These hyperparameters were systematically adjusted and evaluated to optimize the GAN model's performance, which was then assessed using the Euclidean distance metric methodology.

The optimization is an iterative procedure, testing different hyperparameter combinations for effectiveness. This iterative approach serves to fine-tune the model's capabilities and helps prevent overfitting, contributing to a more stable and effective model performance.

Additionally, the GAN's hyperparameters in the model were initially fine-tuned within the classic architecture and subsequently applied without modification to the other architectures. Tailoring the GAN's hyperparameters to each specific architecture could potentially enhance the performance of each one. By individually adjusting these hyperparameters for each architecture, optimized outcomes can be achieved, elevating the overall performance across the diverse architectures.

Chapter 6

Performance Evaluation of the Quantum GAN Implementation

In the previous chapter, we outlined the details of our Quantum GAN implementation, which can be found in [37]. This chapter will focus solely on the evaluation of that implementation. We have evaluated the three architectures developed within the implementation: classic, quantum, and quantum patch. These architectures are explained in greater detail in Section 5.3. The rest of the chapter will delve into these architectures and offer insights into our evaluation criteria. We begin this chapter by detailing our experimental setup.

6.1 Experimental Setup

Our evaluation experiments were executed on Google Colab [40], a cloud-based service developed by Google that provides an interactive environment for machine learning and data science projects. This platform offers the advantage of running Python code within a Jupyter notebook interface, facilitating both code execution and result visualization.

The virtual machine allocated for our experiments was a high-performance computing resource, furnished with an NVIDIA A100 GPU that boasts 40GB of VRAM (Video Random-Access Memory). In addition to the robust GPU, the system was equipped with an Intel Xeon CPU and 83.5 GB of system RAM, providing ample computational capability for our tasks. Storage was not a constraint either, as the virtual machine came with a 166 GB hard drive to store intermediate and final datasets, models, and other relevant files.

Due to the extensive waiting times associated with the IBM Quantum platform [41], we decided not to use real IBM quantum machines during the training stage of our experiments. Instead, we opted for a more efficient approach by utilizing the backend simulator provided by IBM, specifically the *ibmq_qasm_simulator*. This choice allowed us to expedite the experimentation process and efficiently test our quantum algorithms, mitigating the delays typically encountered when accessing physical quantum hardware.

For the optimization of both the generator and discriminator networks in our GAN, we employed the Adam optimizer. Adam, short for "Adaptive Moment Estimation", is an optimization algorithm that computes adaptive learning rates for each parameter. It combines the advantages of two other extensions of stochastic gradient descent: AdaGrad and RMSProp. This makes it particularly effective for problems with large data or many parameters, such as our GAN model.

The learning rate for the generator network was set at 0.00014, while for

the discriminator network, it was 5.62. The learning rate was programmed to undergo a uniform decay after 19 epochs, helping to stabilize training as it progressed. In our training regime, the discriminator was updated five times for every single update of the generator, maintaining a 5:1 update ratio. This strategy ensures that the discriminator is well-trained to distinguish real from generated samples, thereby aiding the generator in producing more realistic data.

With respect to the database, our training set comprised 5400 catalyst reactions extracted from a larger dataset known as the OCP dataset [9]. The data was selectively filtered to include only Cu-based binary catalyst alloys, aligning with the specific training requirements of our GAN model.

6.2 Evaluation Results

Our evaluation is centered on the training phase, which is the most computationally demanding aspect of the entire process. To evaluate the performance of our implementation, we use the Euclidean distance metric. This metric is calculated every 10 epochs to assess the similarity between the actual and the generated catalyst distributions. The primary goal of our evaluation is to examine the ability of our implementation to generate high-quality new catalyst reactions.

6.2.1 Classic Architecture

The Classic architecture serves as the foundational layer of our Quantum GAN model and operates exclusively within the realm of classical computing. To evaluate its performance, we varied the batch size (testing sizes of 8, 16, and 32) while maintaining a constant total of 5000 training epochs. We assessed the quality of the newly generated catalysts by calculating the Euclidean distance every 10 epochs. These results are displayed in Figure 6.1.

Our findings indicate that a batch size of 8 samples yields the lowest Euclidean distance. This is consistent with our expectation that smaller batch sizes would generate catalysts more similar to the original data. Conversely, a batch size of 32 resulted in the most divergent generated catalyst reactions.

It is also worth noting the stability of the generated samples throughout the whole training process. The Euclidean distances did not vary significantly between 1000 and 5000 epochs, suggesting that the GAN model quickly stabilizes after a few epochs.

6.2.2 Quantum Architecture

Our Quantum architecture integrates both a quantum noise generator and a parameterized quantum circuit within the Generator component. Due to the computational intensity of simulating quantum circuits, we limited the training epochs to 1000 when running our experiments on Google Colab. Our evaluations included parameterized quantum circuits with configurations of 8 and 16 qubits, as well as a dual-layer 8-qubit circuit (denoted as L=2 in Figure 6.2). We continued to use the Euclidean distance metric, calculated at intervals of every

10 epochs, to assess the quality of the generated catalysts. For all evaluations, the batch size was kept constant at 16.

The results, illustrated in Figure 6.2, reveal several noteworthy observations. First, the Euclidean distance values were, on average, lower than those generated by the Classic architecture. This suggests a closer resemblance between the generated and real catalysts. However, this architecture also produced more outlier values, resulting in peaks in the Euclidean distance data. We hypothesize that the inherent randomness of quantum computing may contribute to these outlier points.

Additionally, the choice between 8 and 16 qubits did not yield significant differences in terms of the Euclidean distance. However, the computational time for the 16-qubit configuration was noticeably longer, taking considerably more time to complete the 1000-epoch training compared to the 8-qubit configuration.

6.2.3 Quantum Patch Architecture

The Patch architecture adopts a segmented approach, utilizing multiple smaller quantum generators referred to as sub-generators. Designed to accommodate 1, 2, or 4 patches, our implementation was rigorously evaluated across all these configurations. Each configuration consisted of 8 qubits and maintained a consistent batch size of 16 for all experiments. The impact of these configurations on the generative capabilities for new catalyst reactions was thoroughly assessed.

The evaluation results, presented in Figure 6.3, show a notable similarity to those obtained for the quantum architecture. This is encouraging news, as it suggests that the patch method is a viable option for quantum devices with limited computational resources. In this setup, each device could house a quantum generator responsible for processing a small subset of the code. The outputs from these various sub-generators can then be concatenated to form a unified result.

Furthermore, the simulation time for patched quantum circuits is significantly reduced due to the smaller qubit count and early convergence. This reduction in simulation time is particularly advantageous because it addresses GAN training issues such as instability and vanishing gradients. Shortening the neural network depth by using the patch method not only improves the efficiency of the quantum generator but also contributes to the overall stability of the training process. This combination of reduced computational requirements and enhanced stability further underscores the potential of the patch method as an effective strategy for quantum code generation on resource-constrained quantum devices.

6.2.4 Key Takeaways

In this chapter, we concentrated on a thorough evaluation of the three pivotal architectures within our Quantum GAN implementation: classic, quantum, and patch. Each was evaluated according to specific configurations, and key metrics, specifically the Euclidean distance, were calculated at regular intervals to ascertain their capabilities in generating new catalyst reactions.

The classic architecture, operating purely within the classical computing domain, demonstrated more stability throughout the training epochs compared to the quantum and patch architectures. The introduction of quantum components

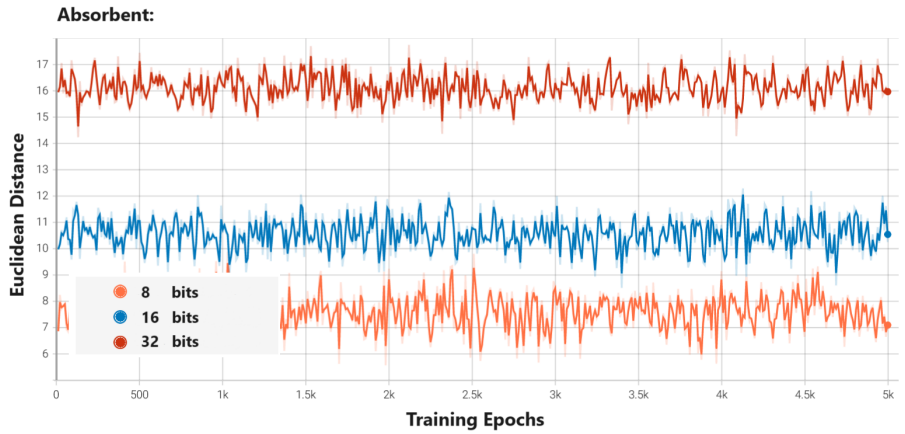
into the quantum architecture led to generally lower Euclidean distance metrics, suggesting a closer similarity to actual catalysts. However, this came at the expense of introducing more outlier values, likely due to the inherent randomness of quantum computing. Interestingly, we found that increasing the number of qubits from 8 to 16 did not yield significant performance improvements but did require additional computational resources.

The patch architecture was primarily designed with a focus on scalability and demonstrated performance metrics comparable with those of the quantum architecture. This indicates its suitability for deployment on quantum devices with limited resources, such as a restricted number of qubits. This architecture could also enable parallel processing across multiple devices, providing a flexible and resource-efficient approach to the computational limitations of current quantum hardware.

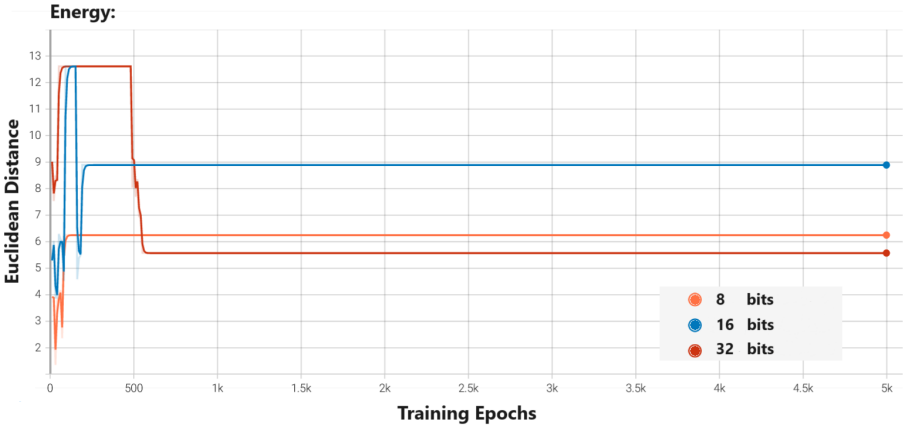
It is worth noting that the GAN model converged quite rapidly during the training phase. Future work might explore strategies to slow down this convergence to assess its impact on the diversity and quality of the generated samples. Tuning GANs effectively remains a complex challenge, often requiring a balanced approach to prevent the discriminator from becoming too effective, thereby rendering it unable to offer useful gradients for the generator's training.

Furthermore, it is also important to note that the hyperparameters of the GAN were tuned in the classic architecture and used in the rest of the architectures as is. Tuning the hyperparameters of the GAN for each architecture might improve the performance of each architecture. Adjusting these hyperparameters individually for each architecture could lead to optimized results and enhance the overall performance of the different architectures.

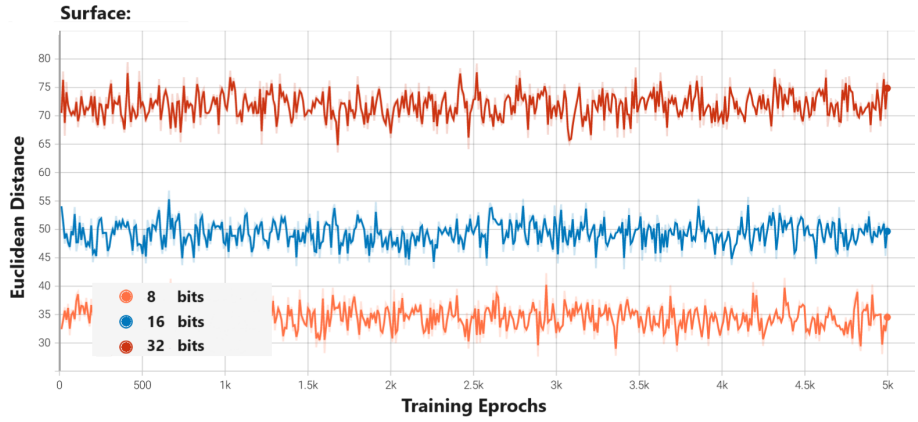
In conclusion, our findings suggest that each architecture has its unique advantages and challenges. While the classic architecture offers stability and reliability, the quantum and patch architectures introduce new dimensions of flexibility and efficiency, albeit at the cost of computational resources.



(a)



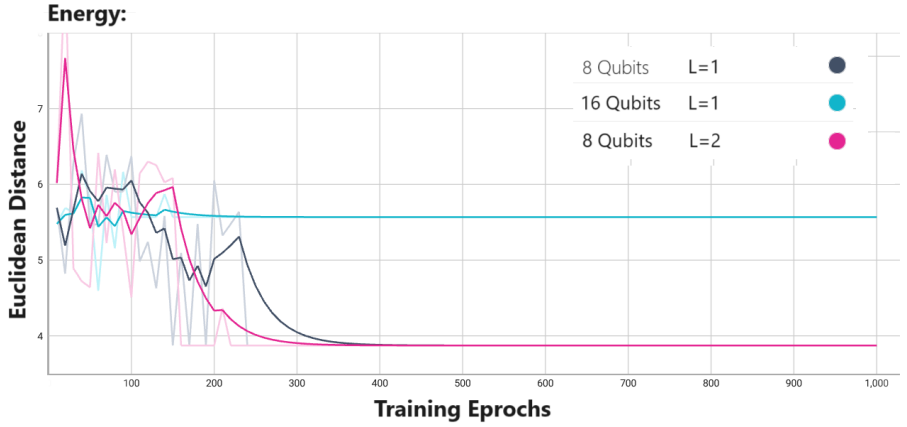
(b)



(c)

Figure 6.1: Assessment Metrics for the Classic Architecture Based on Euclidean Distance, Conducted Over 5000 Epochs with Batch Sizes of 8, 16, and 32: (a) Euclidean Distance for the Composition of Absorbent Molecules; (b) Euclidean Distance for Energy Levels in Catalyst Reactions; (c) Euclidean Distance for the Surface Properties of the Catalyst.

(a)

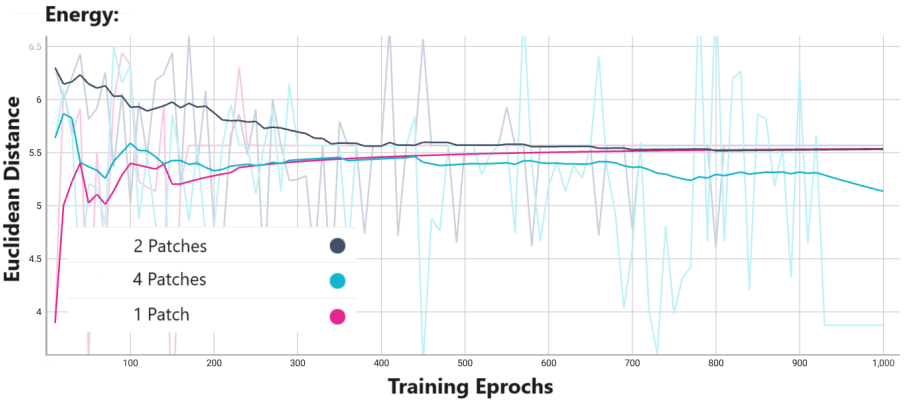


(b)

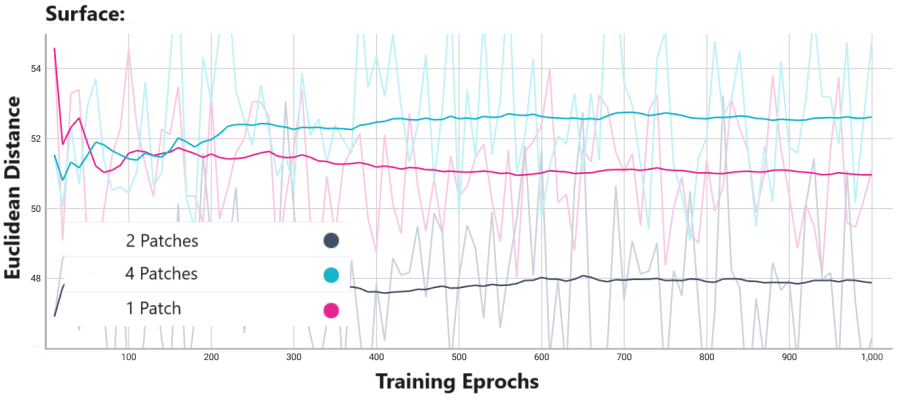
(c)

Figure 6.2: Performance Metrics for the Quantum Architecture Assessed Through Euclidean Distance Over 1000 Epochs, Utilizing a Batch Size of 16 and Configurations of 8 Qubits with 1 (L=1) and 2 Layers (L=2), as well as 16 Qubits with a Single Repeated Layer: (a) Euclidean Distance for Absorbent Molecule Structures; (b) Euclidean Distance Measuring Energy Profiles in Catalyst Reactions; (c) Euclidean Distance for Surface Attributes of Generated Catalysts.

(a)



(b)



(c)

Figure 6.3: Comprehensive Evaluation of the Quantum Patches Architecture Using Euclidean Distance Metrics, Conducted Over 5000 Epochs with 8 Qubits and a Batch Size of 16 Across Patches of 1, 2, and 4: (a) Euclidean Distance for Absorbent Molecule Structures; (b) Euclidean Distance for Energy Profiles in Catalyst Reactions; (c) Euclidean Distance for Surface Attributes of the Catalyst.

Chapter 7

Final Conclusions

In the realm of chemical catalysis, the development of new methods is instrumental not only for advancing our understanding and optimization of various chemical catalysis processes but also for overcoming the fundamental challenges in computational chemistry, such as the complex process of modeling the energy and forces in atomic systems.

This study seeks to explore the capabilities of Generative Adversarial Networks (GANs) as a tool for innovating in the field of chemical catalysis. A special focus has been placed on the formulation and optimization of Copper (Cu) binary catalysts. The research community has recently rekindled its interest in Cu alloy catalysts, largely due to seminal works that demonstrated their efficacy in reducing carbon dioxide levels. These pioneering discoveries were the product of a harmonious blend of machine learning algorithms and hands-on experimental research. Against this backdrop, our research aspires to push the envelope by leveraging quantum computing and artificial intelligence technologies. Specifically, we developed a GAN model enriched with quantum layers to pave the way for the next generation of catalysts.

7.1 Research Challenges and Limitations

While our research has significantly advanced our understanding of chemical catalysis by integrating quantum technology and GANs, it is crucial to candidly address the challenges and limitations we encountered. These factors not only contextualize our findings but also serve as indicators for future research endeavors. In the subsequent paragraphs, we delve into the primary obstacles we faced, specifically focusing on dataset limitations, approximation methods, and the constraints inherent to our theoretical framework.

One of our primary obstacles was related to the dataset limitation. Our model was trained on a relatively modest dataset comprising approximately 5,400 catalyst reactions. This dataset size imposes constraints on the model's ability to generalize effectively, thereby limiting its potential to discover new and diverse catalysis reactions.

Another key challenge lies in the approximation of the global minimum energy for adsorption. Our methodology incorporated multiple initial configurations of the adsorbate on the catalyst surface, followed by local relaxation processes. Although this approach serves as a useful approximation for identifying a global minimum, it remains sensitive to several variables. These include the initial sampling strategy and the size and composition of both the adsorbate and the catalyst material, making the approach less than definitively accurate.

Our final limitation is primarily centered on the theoretical framework we employed. Our computational models are rooted in Density Functional Theory (DFT), which, while offering a reasonable balance between computational cost and accuracy, has its own set of limitations. Specifically, our model does not

consider real-world variables like temperature, solvents, and surface imperfections. Such omissions could lead to discrepancies when the model's predictions are compared with empirical results.

7.2 Future Research Directions

As we look toward the future, several avenues for research emerge that could deepen our understanding of Quantum GAN models and perhaps provide solutions to the challenges and limitations we have encountered. The integration of quantum layers into generative AI models is still an evolving field, and questions remain about the best ways to incorporate these quantum layers within the architecture. For instance, the position of the quantum layer in the generator network of the GAN could significantly impact the model's performance, and early testing suggests that its location relative to other layers like *Batch Normalization* or *Linear layers* could yield different outcomes. Additionally, the advent of more powerful quantum computing platforms could offer the prospect of fine-tuning the model's hyperparameters more effectively, potentially leading to a more reliable and accurate Quantum GAN model. Beyond the focus on Cu-based catalysts, there is also an opportunity to extend the model to generate other types of molecular structures and catalytic processes, further broadening its range of applicability.

7.3 Personal Reflection

As we stand on the cusp of unprecedented technological advancement, quantum technology is poised for exponential growth. As Richard Feynman aptly stated, quantum mechanics adheres to the fundamental rules of nature, thus allowing us to describe the world with unparalleled accuracy. We are at a pivotal juncture in human history, where innovative technologies may be our last hope to mitigate the impending environmental collapse due to rising carbon dioxide levels. The need for immediate action is imperative, or else we face irreversible consequences.

To echo the words of Albert Einstein, *"The most incomprehensible thing about the universe is that it is comprehensible"* [42]. This research journey has been nothing short of magnificent, and it is my hope that the interplay of quantum and AI technologies will help us better comprehend and, consequently, preserve the universe we inhabit.

Bibliography

[1] OPC, "Open Catalyst Project - Github," https://github.com/Open-Catalyst-Project/ocp/, accessed: September 30, 2023.

[2] R. Tran, J. Lan, M. Shuaibi, B. M. Wood, S. Goyal, A. Das, J. Heras-Domingo, A. Kolluru, A. Rizvi, N. Shoghi, A. Sriram, F. Therrien, J. Abed, O. Voznyy, E. H. Sargent, Z. Ulissi, and C. L. Zitnick, "The Open Catalyst 2022 (OC22) Dataset and Challenges for Oxide Electrocatalysts," *ACS Catalysis*, vol. 13, no. 5, pp. 3066–3084, feb 2023. [Online]. Available: https://doi.org/10.1021%2Facscatal.2c05426

[3] R. Medina, *Fermentation technology*. ED-Tech Press, 2019. [Online]. Available: https://doi.org/10.1016/C2013-0-00186-7

[4] M. D. Appl, "The Haber-Bosch Process and the Development of Chemical Engineering," 1982. [Online]. Available: https://api.semanticscholar.org/CorpusID:113894703

[5] J. D. Whitfield, N. Schuch, and F. Verstraete, "The Computational Complexity of Density Functional Theory," in *Many-Electron Approaches in Physics, Chemistry and Mathematics*. Springer International Publishing, 2014, pp. 245–260. [Online]. Available: https://doi.org/10.1007%2F978-3-319-06379-9_14

[6] M. R. Badger and G. D. Price, "The Role of Carbonic Anhydrase in Photosynthesis," *Annual Review of Plant Physiology and Plant Molecular Biology*, vol. 45, no. 1, pp. 369–392, 1994.

[7] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Quantum Chemistry Structures and Properties of 134 Kilo Molecules," *Scientific Data*, vol. 1, 2014.

[8] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, "Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17," *Journal of Chemical Information and Modeling*, vol. 52, no. 11, pp. 2864–2875, 2012, pMID: 23088335.

[9] Open Catalyst Project. Accessed 30 September 2023. [Online]. Available: https://opencatalystproject.org/

[10] L. Chanussot, A. Das, S. Goyal, T. Lavril, M. Shuaibi, M. Riviere, K. Tran, J. Heras-Domingo, C. Ho, W. Hu, A. Palizhati, A. Sriram, B. Wood, J. Yoon, D. Parikh, C. L. Zitnick, and Z. Ulissi, "Open Catalyst 2020 (OC20) Dataset and Community Challenges," *ACS Catalysis*, vol. 11, no. 10, pp. 6059–6072, may 2021. [Online]. Available: https://doi.org/10.1021%2Facscatal.0c04525

[11] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas," *Phys. Rev.*, vol. 136, pp. B864–B871, Nov 1964. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.136.B864

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," 2014.

[13] J. F. Nash, "Equilibrium Points in N-Person Games," *Proc Natl Acad Sci U S A*, vol. 36, no. 1, pp. 48–49, 1950.

[14] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," 2014.

[15] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," 2020.

[16] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017.

[17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," 2017.

[18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral Normalization for Generative Adversarial Networks," 2018.

[19] U. Michelucci, "An Introduction to Autoencoders," 2022.

[20] D. P. Kingma and M. Welling, "An Introduction to Variational Autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019. [Online]. Available: https://doi.org/10.1561%2F2200000056

[21] D. Kingma and M. Welling, "Auto-Encoding Variational Bayes," 2022.

[22] Z. Lu, C. Wu, X. Chen, Y. Wang, L. Bai, Y. Qiao, and X. Liu, "Hierarchical Diffusion Autoencoders and Disentangled Image Manipulation," 2023.

[23] J. D. Havtorn, J. Frellsen, S. Hauberg, and L. Maaløe, "Hierarchical VAEs Know What They Don't Know," 2022.

[24] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Sy2fzU9gl

[25] D. J. Rezende and S. Mohamed, "Variational Inference with Normalizing Flows," 2016.

[26] D. P. Kingma and P. Dhariwal, "Glow: Generative Flow with Invertible 1x1 Convolutions," 2018.

[27] D. P. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational Diffusion Models," 2023.

[28] T. Xie and J. C. Grossman, "Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties," *Phys. Rev. Lett.*, vol. 120, p. 145301, Apr 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.120.145301

[29] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions," 2017.

[30] J. Gasteiger, J. Groß, and S. Günnemann, "Directional Message Passing for Molecular Graphs," in *International Conference on Learning Representations (ICLR)*, 2020.

[31] N. D. Cao and T. Kipf, "MolGAN: An implicit generative model for small molecular graphs," 2022.

[32] J. Li, R. Topaloglu, and S. Ghosh, "Quantum Generative Models for Small Molecule Drug Discovery," 2021.

[33] P.-Y. Kao, Y.-C. Yang, W.-Y. Chiang, J.-Y. Hsiao, Y. Cao, A. Aliper, F. Ren, A. Aspuru-Guzik, A. Zhavoronkov, M.-H. Hsieh, and Y.-C. Lin, "Exploring the Advantages of Quantum Generative Adversarial Networks in Generative Chemistry," *Journal of Chemical Information and Modeling*, vol. 63, no. 11, pp. 3307–3318, may 2023. [Online]. Available: https://doi.org/10.1021%2Facs.jcim.3c00562

[34] A. Ishikawa, "Heterogeneous catalyst design by generative adversarial network and first-principles based microkinetics," *Scientific Reports*, vol. 12, no. 1, p. 11657, Jul 2022.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," 2019.

[36] V. Bergholm, J. Izaac, M. Schuld, I. Sinayskiy, F. Petruccione, and N. Killoran, "PennyLane: Automatic differentiation of hybrid quantum-classical computations," *arXiv:1811.04968*, 2018.

[37] O. Novo, "Quantum GAN with Hybrid Generator for Cu-based Binary Alloy Catalyst Generation," https://github.com/onovo1/Quantum-GAN-Catalysis, 2023, gitHub Repository. [Online]. Available: https://github.com/onovo1/Quantum-GAN-Catalysis

[38] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu, R. Yang, T. Liu, M.-H. Hsieh, H. Deng, H. Rong, C.-Z. Peng, C.-Y. Lu, Y.-A. Chen, D. Tao, X. Zhu, and J.-W. Pan, "Experimental Quantum Generative Adversarial Networks for Image Generation," *Physical Review Applied*, vol. 16, no. 2, aug 2021. [Online]. Available: https://doi.org/10.1103%2Fphysrevapplied.16.024051

[39] T. Akiba, S. Sano, R. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," 2019, software available at https://optuna.org/. [Online]. Available: https://optuna.org/

[40] G. Inc., "Google Colab," https://colab.research.google.com/, 2023, accessed: 30th September 2023.

[41] I. Inc., "IBM Quantum Platform," https://quantum-computing.ibm.com/, 2023, accessed: 30th September 2023.

[42] A. Einstein, "Physics and Reality," *Journal of the Franklin Institute*, vol. 221, no. 3, pp. 349–382, 1936.

Appendices

Appendix A: Detailed Program Parameters

In this section, we provide detailed information about the program parameters used in the Quantum GAN implementation for catalysis generation. The parameters listed below are essential for understanding the configuration of the model and its training process.

Listing 1: Command Line Options for Quantum GAN

```
python main.py -h --quantum BOOL --patches {1,2,4}
               --qubits INT --IBM BOOL --IBM_backend STR
               --g_ReLU_negative_slope FLOAT FLOAT FLOAT
               --d_ReLU_negative_slope FLOAT
               --g_output_features_dim INT INT INT
               --d_output_features_dim INT --dropout FLOAT
               --post_method {softmax,soft_gumbel,hard_gumbel}
               --batch_size INT --num_iters INT
               --n_critic INT --num_iters_decay INT
               --lr_update_step INT --g_lr FLOAT --d_lr FLOAT
               --beta1 FLOAT --beta2 FLOAT
               --weight_decay FLOAT
               --model_iters INT --model_save_step INT
               --model_save_dir STR --store_logs BOOL
               --log_dir STR --log_step INT
               --data_dir STR
               --result_dir STR --mode {train,generation}
```

1. `--quantum`: This parameter determines whether to use the quantum GAN with a hybrid generator. The default value is `True`, which signifies the use of the hybrid quantum-classical generator.

2. `--patches`: The `--patches` parameter specifies the number of quantum circuit patches used in the model. It controls the complexity of the quantum circuits employed during training. The default value is `1`.

3. `--qubits`: This parameter defines the number of qubits and the dimension of domain labels.

4. `--IBM`: When set to `True`, the `--IBM` parameter enables the use of IBM Quantum hardware devices for computation. It is set to `False` by default. Please note that before using this option, IBM Quantum credentials should be set up from `https://quantum-computing.ibm.com/`.

5. `--IBM_backend`: If `--IBM` is enabled, the `--IBM_backend` parameter specifies the backend simulator to use in the IBM Quantum device. The default choice is `ibmq_qasm_simulator`.

6. `--g_output_features_dim`: This parameter determines the number of output features in a fully connected (linear) layer in the generator (`G`). The format is [surface, absorvant, energy].

7. `--d_output_features_dim`: This parameter specifies the number of output features in a fully connected (linear) layer in the discriminator (`D`).
8. `--batch_size`: This parameter sets the mini-batch size for training.
9. `--num_iters`: This parameter defines the total number of iterations (epochs) for training both the discriminator and generator.
10. `--n_critic`: This parameter specifies the number of discriminator updates per each generator update. It helps control the training dynamics.

Please note that this is not an exhaustive list of parameters, and additional parameters are available for fine-tuning the model and training process. For more comprehensive information and additional parameters, please refer to the source code documentation or the program's help message using `python main.py --h`.

Appendix B: Explanation of the Training GAN Implementation

The implementation outlines the training process for a Generative Adversarial Network (GAN) specialized in generating Cu binary catalysts structures. The function comprises five primary segments:

1. **Preprocessing Input Data**: The data for atomic absorbance, surface, and energy are converted to PyTorch tensors and sent to the computing device (GPU or CPU). These tensors are then one-hot encoded.
2. **Training the Discriminator**: The discriminator is trained using both real and generated data. Real data come from the dataset, and fake data are generated by the generator based on random noise, which can be either classical or quantum-based.
3. **Training the Generator**: After a certain number of iterations, the generator is trained to produce data that the discriminator will classify as real.
4. **Miscellaneous**: This involves logging training information and saving the model at specified intervals.
5. **Computing the Euclidean Distance Score**: This is a post-training step where the Euclidean Distance between real and generated data is computed for evaluating the quality of the generated molecular structures.

The pseudocode of the training implementation goes as follows:

Algorithm 1 Training and Evaluating a GAN for Cu Binary Catalysts

```
Initialize loss function to Binary Cross-Entropy with Logits
for each training epoch do
    Load and preprocess real data (absorbance, surface, energy)
    Generate random noise (either classical or quantum)
    Train the Discriminator:
      Generate fake data using the Generator
      Calculate loss for real and fake data
      Update Discriminator parameters
    if epoch mod N == 0 then
       Train the Generator
          Generate fake data using the Generator
          Calculate loss by fooling the Discriminator
          Update Generator parameters
    end if
    Log metrics and save model state
end for
After training, evaluate the quality of the generated Cu binary catalysts
using the Euclidean Distance
```