entropy

IMPACT FACTOR 2.1

Indexed in: PubMed

CITESCORE 4.9

Article

Mapping Data to Concepts: Enhancing Quantum Neural Network Transparency with Concept-Driven Quantum Neural Networks

Jinkai Tian and Wenjing Yang

MDPI

https://doi.org/10.3390/e26110902

entropy

MDPI

Article

Mapping Data to Concepts: Enhancing Quantum Neural Network Transparency with Concept-Driven Quantum Neural Networks

Jinkai Tian [1,*] and Wenjing Yang [2,*]

[1] Intelligent Game and Decision Lab, Beijing 100071, China
[2] Department of Intelligent Data Science, College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China
* Correspondence: tianjinkai13@nudt.edu.cn (J.T.); wenjing.yang@nudt.edu.cn (W.Y.)

Abstract: We introduce the concept-driven quantum neural network (CD-QNN), an innovative architecture designed to enhance the interpretability of quantum neural networks (QNNs). CD-QNN merges the representational capabilities of QNNs with the transparency of self-explanatory models by mapping input data into a human-understandable concept space and making decisions based on these concepts. The algorithmic design of CD-QNN is comprehensively analyzed, detailing the roles of the concept generator, feature extractor, and feature integrator in improving and balancing model expressivity and interpretability. Experimental results demonstrate that CD-QNN maintains high predictive accuracy while offering clear and meaningful explanations of its decision-making process. This paradigm shift in QNN design underscores the growing importance of interpretability in quantum artificial intelligence, positioning CD-QNN and its derivative technologies as pivotal in advancing reliable and interpretable quantum intelligent systems for future research and applications.

Keywords: quantum artifical intelligence; quantum neural networks; explainable artificial intelligence; autoencoder; concept-driven

Citation: Tian, J.; Yang, W. Mapping Data to Concepts: Enhancing Quantum Neural Network Transparency with Concept-Driven Quantum Neural Networks. *Entropy* **2024**, *26*, 902. https://doi.org/10.3390/e26110902

Academic Editor: Giuliano Benenti

Received: 23 September 2024
Revised: 22 October 2024
Accepted: 23 October 2024
Published: 24 October 2024

1. Introduction

The field of quantum computing has garnered significant attention due to its potential to solve complex problems more efficiently than classical computing [1–5]. In particular, quantum computing offers promising advantages in processing large-scale remote sensing data, which is critical in applications like earth observation and remote sensing [6–9]. Quantum neural networks (QNNs), leveraging the principles of quantum mechanics, have shown substantial promise in enhancing computational capabilities and addressing problems that are intractable for classical neural networks [10–13]. Foundational works, such as the Quantum Approximate Optimization Algorithm [14] and the Variational Quantum Eigensolver [15], have paved the way for integrating quantum computing with neural network architectures. Significant advancements in this domain include the introduction of quantum perceptrons [16], quantum support vector machines [5], quantum convolutional neural networks [17], quantum generative adversarial networks [18], and quantum autoencoders [19].

While these investigations elucidate the potential of QNNs, they concurrently highlight a critical challenge: interpretability. In numerous vital domains, including healthcare, finance, and remote sensing, understanding the reasoning behind a model's decisions is as crucial as the decisions themselves. This necessity underscores the importance of model transparency and reliability [20–27]. Quantum eXplainable Artificial Intelligence (QXAI) seeks to enhance the interpretability of quantum models, thereby making them more transparent and reliable for practical applications.

Generally, the complexity of a model is directly related to its accuracy but inversely related to its interpretability. Models with simpler structures are more interpretable but tend to exhibit lower accuracy. Conversely, models with complex structures demonstrate high accuracy; however, due to the large number of parameters, complex mechanisms, and low transparency, their interpretability is relatively poor.

In practical learning tasks, a decision must be made between selecting a simple, easily interpretable model and training it, or training an optimal, complex model and subsequently developing interpretability techniques to explain it. Based on these two approaches, the interpretability of machine learning models can generally be divided into two categories: ante hoc methods [28–30] and post hoc methods [31–33]. *Ante hoc methods* refer to models that are inherently interpretable, either by training models with simple structures or by incorporating interpretability into specific model architectures, thus making the model itself interpretable. Examples of self-explainable models include linear models, generalized linear models [34], decision trees [35,36], and random forests [37]. In contrast, *post hoc methods* treat trained models as black boxes, employing developed interpretability techniques to explain them.

Despite significant advancements in the field of eXplainable Artificial Intelligence (XAI) [23,25,38–41], the development of Quantum eXplainable Artificial Intelligence (QXAI) has been comparatively gradual. Recent research on QXAI has primarily focused on post hoc methodologies [42–46], which are often criticized for potentially providing unreliable or misleading explanations. In contrast, inherently interpretable models offer explanations that are intrinsically aligned with the model's computational processes. Interpretability is domain-specific and frequently constrained by structural knowledge, such as monotonicity [47], causality, additivity [48], or domain-specific physical constraints. For structured data, sparsity is often a valuable measure of interpretability, given that humans can cognitively process only three to five entities simultaneously [49]. Sparse models facilitate an understanding of how variables interact collectively rather than in isolation.

Given these considerations, this study aims to develop an ante hoc method that addresses key challenges in QXAI. Specifically, this research will focus on:

- Formulating an effective strategy for disentangling and representing concepts within a QNN model.
- Ensuring that the concepts are sparse and independent of each other.
- Designing a QNN that seeks to balance high predictive accuracy with interpretability.

To tackle these challenges, this study proposes the development of the concept-driven quantum neural network (CD-QNN) model. The CD-QNN model aims to maintain high predictive accuracy while providing clear and meaningful explanations for its decisions. By integrating the strengths of QNNs with the interpretability of self-explanatory models, CD-QNN maps input data into a human-understandable concept space and bases its decisions on these concepts. This approach not only bridges the gap between the computational power of QNNs and the need for model interpretability but also enhances the trustworthiness and reliability of quantum artificial intelligence systems.

To ensure that the explanations provided by the CD-QNN model are direct and easily understandable, measures have been implemented to align these explanations closely with the model's behavior, accurately reflecting the true importance of features in the decision-making process. It is crucial to maintain the consistency and reliability of the explanations, ensuring that the model generates similar explanations for similar input samples. This consistency allows users to quickly grasp the basis of the model's decisions, thereby enhancing the model's transparency and credibility.

The contents of this paper are organized as follows. Section 2 reviews the pertinent literature and identifies gaps in the current research. Section 3 examines the design and implementation of the concept generator, demonstrating how abstract, human-interpretable concepts are derived from raw input data using a QVAE. Section 4 outlines a detailed algorithmic framework for CD-QNN, covering the *concept generator*, *feature extractor*, and *feature integrator*. Section 5 focuses on optimizing the training strategy to enhance and

balance classification performance and interpretability. Section 6 presents the experimental results validating the effectiveness of CD-QNN, emphasizing its predictive accuracy and the clarity of its model explanations. Finally, Section 7 summarizes this study's contributions and proposes directions for future research.

2. Related Works

The field of Quantum Neural Networks (QNNs) has seen significant advancements in recent years, driven by the potential of leveraging quantum computing to solve complex problems more efficiently than classical methods [12]. Numerous studies have explored the computational advantages and applications of QNNs across various domains [50–52].

One of the foundational developments in this area is the quantum variational algorithm, which employs quantum circuits with parameterized gates that can be optimized similarly to neural networks in classical machine learning [53]. Farhi et al. [14] introduced the Quantum Approximate Optimization Algorithm (QAOA), demonstrating the potential of quantum circuits to solve combinatorial optimization problems. Similarly, the Variational Quantum Eigensolver (VQE) has been widely used in quantum chemistry to find the ground state energies of molecules [15]. These algorithms laid the groundwork for integrating quantum computing with neural network architectures [54].

Building on these concepts, several researchers have proposed quantum versions of classical neural network models. For instance, Schuld et al. [55] developed a quantum perceptron model, highlighting the feasibility of implementing neural network operations on quantum computers. Subsequent works extended these ideas, introducing quantum convolutional neural networks [17,56]. These studies have demonstrated that QNNs can outperform their classical counterparts in specific tasks, especially as quantum hardware continues to advance [57].

Despite these promising developments, a significant challenge in the field of QNNs is the interpretability of the models. Most existing QNNs function as black boxes, making it difficult for users to understand the decision-making processes. This lack of transparency hinders the adoption of QNNs in critical applications where interpretability is essential, such as healthcare and finance [27].

Traditional methods in classical machine learning to enhance the interpretability include techniques like saliency maps [32,58,59], LIME (Local Interpretable Model-agnostic Explanations) [31], and SHAP (SHapley Additive exPlanations) [60]. These methods aim to provide insights into how individual features contribute to the model's predictions. Directly applying these techniques to QNNs is not straightforward due to the intrinsic differences between classical and quantum data representations [13].

In the context of quantum computing, a few pioneering studies have attempted to introduce interpretability. Burge et al. [42] introduced a quantum algorithm to estimate Shapley values, facilitating fair payoff distribution in cooperative game scenarios through innovative beta function approximations. Heese et al. [43] extended classical Shapley values to quantum circuits, developing PolynomialSHAP for feature importance in Quantum Machine Learning (QML) models, demonstrating robustness across simulated and real quantum environments. Steinmüller et al. [46] explored ways to accelerate the computation of Shapley values using the internal mechanics of QNNs. Pira et al. [45] proposed Quantum LIME, an adaptation of classical techniques to the quantum domain. Mercaldo et al. [44] investigated QML applications in mobile malware detection, highlighting the critical role of explainability in security contexts. Collectively, these studies enhance the transparency and trustworthiness of QML, fostering its adoption in practical applications.

3. Concept Generator

In traditional interpretable model frameworks, each input variable is typically treated as a fundamental unit in the explanation process. While this approach is technically rigorous, it does not fully align with how humans process information. For example, when interpreting an image, humans do not rely on individual pixels to explain the content;

instead, they use more abstract, high-level concepts such as object contours or brushstroke thickness. Similarly, in the quantum realm, a single qubit state can be compared to a pixel in computer vision, while the relationships and entanglements between multiple qubits represent higher-order, human-understandable concepts.

To incorporate these concepts into the model instead of relying on raw inputs, we mathematically define a mapping function $c(\cdot) : \mathcal{X} \to \mathcal{C} \subset \mathbb{R}^k$, where $\mathcal{X}$ represents the original input space, and $\mathcal{C}$ represents the space composed of human-interpretable concepts. To ensure sparsity, these concepts should be simple and easy for humans to understand, with the number of concepts k controlled within a small range.

The construction of the mapping function $c(\cdot)$, or the abstraction of key concepts from raw inputs, must not only provide sufficient information for subsequent model judgments but also ensure that these concepts are intuitive and verifiable from a human perspective. One method to achieve this is through predefined feature extractors, typically built based on domain experts' knowledge [61–63]. This approach often requires customized feature extractors for different domains, which can be costly and lack general applicability. An alternative approach is to learn latent space representations to extract concepts and impose specific constraints on these representations to ensure that the extracted concepts have practical significance [64,65].

When constructing a conceptualized model, it is essential to focus on the following core attributes:

1. The critical information in the input data should be maximally captured by the abstracted concepts. This can be achieved by defining $c(\cdot)$ as the encoder part of a quantum variational autoencoder (QVAE) [66]. In this architecture, the input $\boldsymbol{x}$ is transformed into a lower-dimensional representation by the encoder $c(\cdot)$ and then reconstructed by the decoder $d(\cdot)$, i.e., $\hat{\boldsymbol{x}} = d(c(\boldsymbol{x}))$. This low-dimensional latent space representation encapsulates the most critical information contained in the input $\boldsymbol{x}$.
2. To ensure both sparsity and disentanglement, the input data should be representable by a finite and non-overlapping set of concepts. This can be accomplished through the sparsity control of the QVAE. A sparse autoencoder activates only a small part of the latent space dimensions for a given input. Increasing sparsity encourages the model to represent the input with fewer concepts while leveraging the advantages of β-VAE can make the latent space representations more inclined toward disentanglement and independence.
3. The extracted concepts need to be intuitively interpretable by humans. This can be achieved through *prototype interpretation* of concepts. By varying $c(\boldsymbol{x})_i$ for the same data and inputting the altered latent space representation into the decoder to obtain reconstructed data, the impact of concept representation changes on the reconstructed data can be observed. This method of interpreting concepts is referred to as prototype interpretation.

3.1. Designing QVAE as a Concept Generator

Existing research on QVAE [66] primarily employs energy models [67], such as quantum Boltzmann machines [68,69], as decoders. This reliance on specific quantum models limits QVAE's flexibility in training and scalability, particularly in leveraging contemporary deep neural network technologies. Therefore, this study aims to redesign the QVAE architecture to enhance its efficiency and scalability, making it more suitable as a concept generator in CD-QNNs.

The model architecture of QVAE used to train the concept generator is shown in Figure 1. Inspired by the flexibility of quantum adversarial neural networks in independently selecting models for generators and discriminators [18,70,71], the encoder and decoder of the newly designed QVAE can independently choose between quantum or classical models.

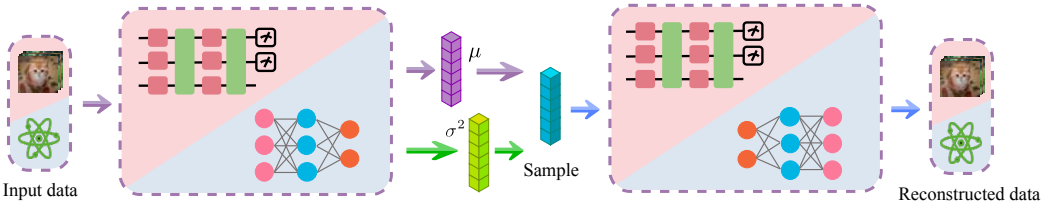

Figure 1. Model architecture diagram of QVAE used to train the concept generator in CD-QNN. This figure illustrates the architecture of the Quantum Variational Autoencoder (QVAE) used to generate human-interpretable concepts in CD-QNN. The purple and green vectors represent the mean and variance outputs from the decoder, respectively. These are used to sample a latent vector, depicted in blue, which serves as a one-dimensional vector. Each element of this vector corresponds to a distinct concept, which is used in the subsequent model stages.

When a quantum neural network (QNN) is selected as the encoder and a deep neural network (DNN) as the decoder, the QVAE functions as a quantum-classical hybrid model. To clearly distinguish between the VAE's encoder and the process of loading classical data into quantum states, the latter is referred to as the *dataloader*.

The encoder utilizes a hardware-efficient ansatz, which enhances the expressivity of the quantum circuit through a series of parameterized unitary transformations [72]. This approach employs a layered structure, where each layer consists of parameterized rotation gates (based on Pauli X, Y, and Z matrices) and entanglement gates (such as CNOT gates), enabling efficient quantum computation while ensuring compatibility with current quantum hardware limitations [57]. The adjustable parameters within these rotation gates allow the quantum circuit to adapt and optimize for specific quantum tasks during the learning process. Quantum state measurements and quantum information processing are primarily conducted using orthogonal measurements on a computational basis. These measurements are crucial for the effective conversion and processing of quantum-classical information required for downstream tasks [73].

Mathematically, the quantum encoder can be represented as follows. First, a dataloader $U_{\boldsymbol{x}}(\boldsymbol{x})$ encodes classical data $\boldsymbol{x}$ into a quantum state $|\psi(\boldsymbol{x})\rangle$:

$$|\psi(\boldsymbol{x})\rangle = U_{\boldsymbol{x}}(\boldsymbol{x})\,|0\rangle^{\otimes n}\,. \tag{1}$$

This encoding can typically be achieved using rotation gate dataloaders, amplitude dataloaders, or data re-uploading dataloaders [74–76]. For instance, using a rotation gate dataloader, the classical data $\boldsymbol{x}$ can be mapped to a quantum state as follows:

$$U_{\boldsymbol{x}}(\boldsymbol{x})\,|0\rangle^{\otimes n} = \bigotimes_{i=1}^{n} R_y(x_i)\,|0\rangle\,, \tag{2}$$

where $R_y(x_i) = e^{-ix_iY/2}$ represents the rotation around the Y-axis by angle x_i, and $|0\rangle^{\otimes n}$ is the initial state of n qubits, all set to $|0\rangle$.

The quantum circuit is then parameterized as

$$|\phi(\boldsymbol{x};\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|\psi(\boldsymbol{x})\rangle\,, \tag{3}$$

where $U(\boldsymbol{\theta})$ represents the hardware-efficient ansatz with parameters $\boldsymbol{\theta}$. Specifically, $U(\boldsymbol{\theta})$ can be decomposed into a sequence of parameterized single-qubit rotation gates and entangling gates, as follows:

$$U(\boldsymbol{\theta}) = \prod_{l=1}^{L} U_l(\boldsymbol{\theta_l}) \tag{4}$$

$$U_l(\boldsymbol{\theta_l}) = \prod_{q=1}^{n} U_{l,q}(\boldsymbol{\theta_{l,q}}) U_{ENT} \tag{5}$$

$$U_{l,q}(\boldsymbol{\theta_{l,q}}) = R_z(\theta_{l,q}^{(1)}) R_x(\theta_{l,q}^{(2)}) R_z(\theta_{l,q}^{(3)}) \tag{6}$$

$$U_{ENT} = \prod_{q=1}^{n} \mathrm{CNOT}(q, (q+1) \mod n), \tag{7}$$

where $R_x(\theta)$ and $R_z(\theta)$ are rotation gates around the X and Z axes, respectively, applied to qubit q in layer l, and $\mathrm{CNOT}(q, (q+1) \mod n)$ represents a controlled-NOT gate acting on qubits q and $(q+1) \mod n$.

Finally, the measurement operation collapses the quantum state to a classical vector $\boldsymbol{z} = [z_1, z_2, \ldots, z_n]$, as follows:

$$z_i = \langle \phi(\boldsymbol{x}; \boldsymbol{\theta}) | \hat{Z}_i | \phi(\boldsymbol{x}; \boldsymbol{\theta}) \rangle, \tag{8}$$

where $\hat{Z}_i$ is the Pauli-Z operator acting on the i-th qubit, and z_i represents the expectation value of the measurement outcome for qubit i.

Unlike traditional autoencoders that map inputs directly to fixed points in the latent space, variational autoencoders (VAEs) map inputs to probability distributions in the latent space. Following the quantum model output, two classical fully connected layers estimate the mean and variance of the probability distribution, as follows:

$$\mu = \mathrm{FC}_1(\boldsymbol{z}), \quad \sigma = \mathrm{FC}_2(\boldsymbol{z}), \tag{9}$$

where FC_1 and FC_2 are fully connected layers. The reparameterization trick is then applied to ensure differentiability:

$$\tilde{\boldsymbol{z}} = \mu + \sigma \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{10}$$

Subsequently, a DNN serves as the decoder. The decoder network, represented as $d(\cdot)$, maps the latent variable $\tilde{\boldsymbol{z}}$ back to the original input space, as follows:

$$\hat{\boldsymbol{x}} = d(\tilde{\boldsymbol{z}}). \tag{11}$$

The overall loss function for training the QVAE combines reconstruction loss and a regularization term to enforce a prior distribution on the latent space, as follows:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z})), \tag{12}$$

where D_{KL} represents the Kullback–Leibler divergence.

Conversion Between Quantum and Classical Data

It should be noted that the conversion between quantum and classical data inevitably results in resource consumption and potential information loss. For input data originally in the form of quantum states, it can be directly processed by the quantum encoder. Classical data can be converted to quantum states using dataloaders. In practical applications, selecting the appropriate QVAE configuration to balance computational efficiency and accuracy is crucial. By simply modifying the encoder, it can be adapted to function as a DNN. Similarly, the decoder can be transformed into a QNN. This flexibility allows the concept generator to handle both quantum and classical data, achieving seamless integration and enhancing the model's versatility.

3.2. Sparse and Disentangled Representations in QVAE

Typically, the divergence of the encoder's output from a standard Gaussian distribution is measured using the KL divergence, which is then minimized. To encourage the encoder to generate sparse latent space variables, an L1 regularization term can be added to the autoencoder's loss function. This approach reduces redundancy between features and aims to decouple concepts as much as possible, as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \lambda \sum_{i=1}^{n} |z_i|. \tag{13}$$

Merely enforcing the activation of a subset of the input data's dimensions does not guarantee the independence of concepts. Each concept may still represent a complex and intertwined set of features, complicating the interpretation of each individual concept learned by the model. To address this issue more effectively, the loss function of β-VAE [77] is adopted as a framework for learning disentangled concepts. In the β-VAE model, an adjustable hyperparameter β introduces stronger constraints on the latent space by increasing the penalty on KL divergence. As the value of β increases, the latent space is encouraged to approach a unit Gaussian distribution. Given the independence of dimensions in a unit Gaussian distribution, this mechanism also promotes the disentanglement of latent factors, as follows:

$$\mathcal{L} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \beta D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z})). \tag{14}$$

By carefully adjusting β, CD-QNN ensures that the latent space is sufficiently disentangled to produce interpretable concepts while preserving important information and maintaining high reconstruction fidelity. As demonstrated in our experiments, increasing β improves disentanglement without significantly degrading reconstruction quality or predictive performance. In addition, sparsity is selectively applied to activate only the most relevant concepts, reducing redundancy and focusing the model on essential features. This approach aligns with human cognitive processing, making the model more interpretable without sacrificing expressiveness. Through our ablation studies, we show that even with sparsity constraints, CD-QNN maintains high classification accuracy, confirming that critical quantum information is retained.

3.3. Prototype Interpretation

Utilizing the properties of the QVAE, it is possible to generate continuously varying data prototypes. Unlike the original inputs of the training dataset, these prototypes can freely navigate within the dimensional space of the concepts. By analyzing the reconstructed data from these continuous variations, a more comprehensive understanding of the concept's significance can be attained, thereby enabling the exploration and visualization of the specific content of the concepts. This method not only deepens our grasp of the concepts themselves but also enhances our comprehension of how the model discerns and differentiates these concepts. Consequently, this approach offers new insights into the intrinsic structure of data and the decision-making logic of the model.

3.4. Mitigating KL Divergence Vanishing

During the training of QVAE, an issue known as the KL divergence vanishing problem was identified. This phenomenon occurs when the Kullback–Leibler (KL) divergence term, which serves as the regularization component, approaches zero throughout the training process. The KL divergence encourages the latent representation to conform to a prior distribution, typically a standard normal distribution. When the KL divergence diminishes, it suggests that the model may neglect the structure of the latent space, focusing solely on minimizing reconstruction error. This oversight results in a lack of meaningful structure within the latent space, thereby adversely impacting the model's performance and generalization capabilities.

To mitigate this issue, additional configurations were incorporated into the QVAE model, as suggested by [78]. Specifically, Batch Normalization layers were integrated to standardize the mean and variance of the latent space, thereby stabilizing and accelerating the training process. Subsequently, the mean and variance were adjusted using the following scaling layer:

$$\mu = \sqrt{\tau + (1-\tau)\cdot \text{sigmoid}(\theta)}\mu, \tag{15}$$

$$\sigma = \sqrt{(1-\tau)\cdot \text{sigmoid}(-\theta)}\sigma \tag{16}$$

where τ is a hyperparameter ranging between 0 and 1, and θ is a trainable parameter. This approach enhances the model's responsiveness to the KL divergence by modifying the latent space scale, thereby addressing the KL divergence vanishing problem. Subsequent experiments demonstrated that incorporating this scaling layer effectively improves the model's training speed, albeit at the cost of a slight increase in QVAE's reconstruction loss.

4. CD-QNN Architectural Design

The CD-QNN model is composed of three fundamental components: a concept generator c, a feature extractor θ, and a feature integrator g. The comprehensive architecture is depicted in Figure 2. In this structure, the concept generator compresses the raw data into a sparse conceptual representation, which can be realized through the encoder of a QVAE. This encoder converts raw data into a conceptual representation comprehensible to humans. The feature extractor, typically realized by a DNN, is tasked with extracting pertinent information from the raw data to ensure adequate expressivity. The feature integrator amalgamates various concepts into the final output. To preserve the model's interpretability, the feature integrator should be as simplistic as possible. This simplicity aids in understanding each concept's contribution to the final output.

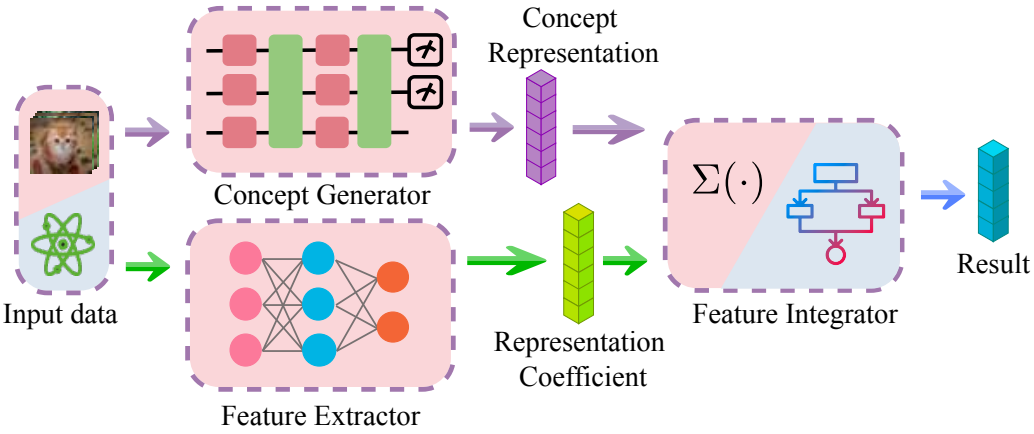

Figure 2. This figure illustrates the architecture of CD-QNN. The purple vectors represent the concept representations, while the green vectors indicate the representation coefficients. Together, these elements are used by the feature integrator to make the final prediction.

Achieving a balance between the algorithm's interpretability and expressivity is crucial. The deep learning model's expressivity refers to its ability to represent and approximate complex functions. This capability is influenced by factors such as the number of parameters, network depth, width, and the choice of activation functions. Quantitative analysis of expressivity includes evaluating parameter count and capacity, the Vapnik–Chervonenkis (VC) dimension [79], and empirical studies on generalization error versus training error [80]. Theoretical approaches, such as the Universal Approximation Theorem [81], and practical methods, including pruning and analyzing information bottlenecks [82], provide insights into a model's expressivity. Recent studies also highlight the potential of quantum-inspired enhancements for generative models, where quantum correlations

provide a powerful resource for improved expressivity [83]. Moreover, the expressivity of variational quantum algorithms has been analyzed using advanced tools in statistical learning theory, revealing that the number of quantum gates and measurement observables upper bound expressivity, with implications for the trainability and generalization of QNNs [84].

This equilibrium between interpretability and expressivity is well-reflected in the selection of three components. The choice of an appropriate model is contingent on the relative importance of interpretability and expressivity. The flexibility of the CD-QNN design lies in its modularity. Each component—the concept generator, feature extractor, and feature integrator—can be independently selected and configured based on specific application requirements. This modular approach allows for tailored adjustments to optimize the balance between representational capacity and interpretability. For instance, in applications where interpretability is paramount, simpler models for the feature integrator and concept generator can be chosen. Conversely, for tasks demanding higher expressivity, more complex models can be employed, ensuring the model remains robust and versatile across diverse scenarios. This design philosophy underscores the adaptability of CD-QNN, making it a powerful framework for integrating quantum and classical computing paradigms in machine learning.

The ensuing discussion will delve into the design methodologies of the model architecture, commencing with the most elementary linear models and gradually advancing towards more intricate constructs. Linear models possess the inherent advantage of interpretability. By methodically abstracting and complicating them, the model's expressivity can be enhanced. The challenge in this process lies in retaining the model's essential interpretability while augmenting its expressivity.

4.1. Linear Models

Consider a set of input features $x_1, \ldots, x_n$ and their corresponding parameters $\theta_1, \ldots, \theta_n$. The prediction of a linear model can be concisely represented as

$$f(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x} = \sum_{i=1}^{n} \theta_i x_i. \tag{17}$$

For simplicity, the bias term is temporarily disregarded in this model. This model is considered highly interpretable due to the following conditions:

1. There exists a direct and explicit relationship between the input features and the model's decision-making process.
2. Each parameter θ_i quantitatively reflects the influence of its corresponding feature x_i on the prediction outcome, with the sign of θ_i indicating whether this influence is positive or negative.
3. The overall prediction output of the model is a linear aggregation of the influences of each feature, ensuring the clarity of each feature's impact on the prediction. There is no interaction between features, thus avoiding any ambiguity in interpretation.

Building upon this foundation, we will incrementally extend the linear model to explore how to preserve the integrity of interpretability mechanisms as the model's complexity escalates, aiming to enhance the model's expressivity while maintaining the transparency of the decision-making process.

4.2. Feature Extractor

To augment the expressivity of the linear model while preserving its overall structure, the model's coefficients can be dynamically adjusted based on the input $\boldsymbol{x}$. This adjustment can be mathematically represented as $f(\boldsymbol{x}) = \boldsymbol{\theta}(\boldsymbol{x})^T \boldsymbol{x}$, where $\boldsymbol{\theta}$ is derived from a model space Θ, which encompasses a range of models, from conventional regression models to DNNs. In the absence of additional constraints, this dynamic coefficient model can achieve

functionalities akin to those of DNNs. For multi-class classification tasks, $\boldsymbol{\theta}(\boldsymbol{x})$ is a $k \times n_{cl}$ matrix, where k denotes the input dimension and n_{cl} denotes the number of classes.

To ensure interpretative stability, it is imperative to guarantee that the variations in coefficients $\boldsymbol{\theta}(\boldsymbol{x})$ and $\boldsymbol{\theta}(\boldsymbol{x}')$ for similar inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$ are minimal. By incorporating regularization terms, such as enforcing $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_0) \approx \boldsymbol{\theta}(\boldsymbol{x}_0)$ in the vicinity of $\boldsymbol{x}_0$, the model can locally approximate a linear model with stable coefficient vectors $\boldsymbol{\theta}(\boldsymbol{x}_0)$. This design ensures that the modifications in coefficients $\theta_i(\boldsymbol{x})$ remain comprehensible, allowing them to dynamically adjust according to variations in input $\boldsymbol{x}$, but in a relatively gradual manner.

This regularization terms serve to control the fluctuations in coefficients steadily, guiding the model to effectively extract pertinent information from complex data, thus preventing overfitting while maintaining interpretability. The meticulous selection and design of regularization techniques are crucial in balancing model interpretability and performance, influencing the model's behavior at individual points $\boldsymbol{x}_i$ as well as its generalization capability across the entire input space.

4.3. Concept Generator

To transition from raw data to human-understandable concepts, the encoder of a QVAE is employed to derive these *concept representations*, denoted as $c(\boldsymbol{x})$. By utilizing these methodologies, an interpretable model has been developed, as follows:

$$f(\boldsymbol{x}) = \boldsymbol{\theta}(\boldsymbol{x})^T \boldsymbol{x} = \sum_{i=1}^{k} \theta(\boldsymbol{x})_i c(\boldsymbol{x})_i, \tag{18}$$

where each $c(\boldsymbol{x})_i$ is a numerical scalar representing the degree of presence of a specific idea in the input $\boldsymbol{x}$. These corresponding coefficients are referred to as *representation coefficients*. The *concept contribution* is obtained by multiplying the concept representation with the representation coefficient, determining the contribution of the concept to the class prediction. If a sample has a negative representation coefficient for a certain concept, it indicates a weak expression of that concept in the sample; if the corresponding concept representation is also negative, it implies that a weak representation of that concept is needed for the sample to belong to the current class. Then the product of these two values results in a positive concept contribution, indicating that the concept positively contributes to the class determination.

For example, considering the idea of brushstroke thickness, the value of $c(\boldsymbol{x})_i$ can range from -1 to 1, where -1 indicates extremely thin brushstrokes and 1 indicates extremely thick brushstrokes. The identification of the strength of these ideas, obtained through the encoder, and their corresponding coefficients in the samples is achieved through a learning process. This process ensures that each representation accurately reflects the nuances of the input data.

4.4. Feature Integrator

A feature integrator g is introduced to thoroughly process the transformed features and generated concepts. Specifically, the final form of the model can be expressed as

$$f(\boldsymbol{x}) = g(\theta(\boldsymbol{x})_1 c(\boldsymbol{x})_1, \ldots, \theta(\boldsymbol{x})_k c(\boldsymbol{x})_k), \tag{19}$$

where g is a function that retains the model's interpretability attributes.

Linear models are an intuitive and widely utilized choice for feature integrators due to their interpretability and simplicity. By linearly combining various features, the contribution of each feature to the final result can be directly observed. Generalized linear models extend linear models by allowing the response variable's distribution to belong to the exponential family and linking the linear predictor to the expected response through a link function [34]. These models perform well when dealing with data exhibiting specific distribution characteristics while maintaining a degree of interpretability. Decision trees

represent another viable option. They construct a tree-like structure to partition the feature space into different regions and make decisions within each region [35,36]. The advantage of decision trees lies in their clear structure, which simplifies the decision process. The splitting conditions at each node and the output results at leaf nodes possess explicit physical meanings, aiding in the explanation and analysis of the model.

In practical applications, the appropriate feature integrator can be selected based on the data characteristics and model requirements. By making an informed choice, the prediction accuracy and generalization capability of the model can be enhanced while ensuring its interpretability.

5. Training Strategy

Building on the previous analysis and design, a two-stage training strategy is proposed to balance the model's pursuit of interpretability and classification performance. Since both the concept generator and feature extractor are functions of the input data, training these two components simultaneously would conflate the degree of expression of concepts (latent space representation of the concept generator) with the relevance coefficients of concepts in the model's judgment (output of the feature extractor). Therefore, a *two-stage* training process is essential. Empirical evidence indicates that concurrent training of the concept generator and feature extractor can lead to model collapse.

In the first stage, the focus is on training the concept generator to acquire the latent space representations of the input data. The second stage involves training the feature extractor and feature integrator while keeping the parameters of the concept generator fixed. If a linear model is selected as the feature integrator, optimization is unnecessary.

5.1. Concept Generator Loss

The objective of optimizing the concept generator is not only to accurately reconstruct the initial input with the generated concepts but also to ensure the orthogonality of the concepts so that the latent variables in each dimension are as independent as possible. To meet this requirement, a concept loss $\mathcal{C}_c(\boldsymbol{x})$ is introduced. Within the β-VAE framework, this loss function is mathematically expressed as

$$\mathcal{C}_c(\phi, \psi; \boldsymbol{x}, \beta) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\psi(\boldsymbol{x}|\boldsymbol{z})] - \beta D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z})), \tag{20}$$

where ϕ and ψ denote the parameter sets of the encoder and decoder, respectively. The first term represents the reconstruction error, measuring the model's accuracy in reconstructing the input data through latent variables, while the second term is the KL divergence regularization term, measuring the deviation of the latent variable distribution from its prior distribution, with β being a hyperparameter controlling the balance between the two terms.

KL (Kullback–Leibler) divergence measures the difference between two probability distributions. In VAE, the goal is to learn a latent space distribution that approximates a standard normal distribution as closely as possible. The formula for KL divergence in this context is

$$D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z})) = \frac{1}{2} \sum_{j=1}^{J} (\mu_j^2 + \sigma_j^2 - \log(\sigma_j^2) - 1), \tag{21}$$

where $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is the distribution of latent variables $\boldsymbol{z}$ output by the encoder, $p(\boldsymbol{z})$ is the desired standard normal distribution, and μ_j and σ_j are the mean and variance of the latent variables. This formula is derived by calculating the expected values and logarithmic differences in the two distributions. In practice, the KL divergence of each sample is summed and averaged to obtain the KL divergence for the entire batch.

5.2. Classification Loss

To ensure the model's accuracy in handling real datasets, a classification loss $\mathcal{C}_\theta(\boldsymbol{x}, y)$ is introduced, reflecting the consistency between the model's predictions and the true labels. In classification problems, the cross-entropy loss function is commonly used to evaluate

model performance. Assuming the model's output $f(\boldsymbol{x})$ is the predicted probability distribution for each class and y is the true label distribution, usually in one-hot encoding form, the mathematical expression of the cross-entropy loss function is

$$\mathcal{C}_\theta(\boldsymbol{x}, y) = -\sum_i y_i \log(f_i(\boldsymbol{x})), \tag{22}$$

where y_i indicates the presence of the true label in class i and $f_i(\boldsymbol{x})$ represents the model's predicted probability for class i. For binary classification problems, this loss function can be further simplified to

$$\mathcal{C}_\theta(\boldsymbol{x}, y) = -[y \log(p) + (1-y) \log(1-p)] \tag{23}$$

where p is the model's predicted probability of the positive class and y is the binary true label (0 or 1).

5.3. Stability of Interpretations

To ensure the stability of the generated interpretations, constraints on the model's continuity and local stability are imposed. Assuming θ_i is a constant parameter and treating the function f as a function of $c(\boldsymbol{x})$, i.e., $f(\boldsymbol{x}) = g(c(\boldsymbol{x}))$, and let $\boldsymbol{z} = c(\boldsymbol{x})$. Applying the chain rule, we obtain $\nabla_x f = \nabla_z f \cdot J_c(\boldsymbol{x})$, where $J_c(\boldsymbol{x})$ represents the Jacobian matrix of c with respect to $\boldsymbol{x}$. At a data point $\boldsymbol{x}_0$, we want $\theta(\boldsymbol{x}_0)$ to approximate the derivative of f with respect to the concept vector $c(\boldsymbol{x})$ at least locally, i.e.:

$$\|\nabla_x f(\boldsymbol{x}) - \theta(\boldsymbol{x})^T J_c(\boldsymbol{x})\| \approx 0. \tag{24}$$

Based on this, the difference can be included as a regularization term $\mathcal{R}(\boldsymbol{x})$ in the following loss function:

$$\mathcal{R}(\boldsymbol{x}) := \|\nabla_x f(\boldsymbol{x}) - \theta(\boldsymbol{x})^T J_c(\boldsymbol{x})\|. \tag{25}$$

It should be noted that involving the calculation of second-order derivatives will significantly increase the complexity of model training. Empirical studies have shown that in certain scenarios, the interpretations themselves exhibit good stability, so the decision to introduce this regularization term should be balanced against training resources and the stability performance in the specific scenario.

6. Experiments

In this section, we present a comprehensive evaluation of the Concept-Driven Quantum Neural Network (CD-QNN) model through a series of meticulously designed experiments. The DIGIT dataset is utilized to train, validate, and test the model, employing various configurations to assess performance. We conduct ablation studies to investigate the impact of different parameters and components on the model's classification accuracy, reconstruction error, and interpretability. By examining multiple experimental setups, we aim to highlight the strengths and limitations of the CD-QNN, providing insights into the effectiveness of the two-stage optimization strategy and the role of quantum-classical hybrid models in enhancing machine learning frameworks. This section is structured to present the preprocessing steps, training methodologies, detailed results from ablation studies, and in-depth analyses of classification accuracy, reconstruction error, and prototype interpretation.

6.1. Experiment Setup

The DIGIT dataset was preprocessed and partitioned into training, validation, and test sets in proportions of 72%, 8%, and 20%, respectively. During the training phase, the Adam optimizer was employed with an initial learning rate of 0.001, β parameters of $(0.9, 0.999)$, ϵ set at 1×10^{-8}, without weight decay, and AMSGrad was not utilized [85,86].

The training process continued for multiple epochs until the model's performance met the desired criteria or the stopping conditions were met.

As detailed in Section 3.1, the QVAE model was trained using a quantum encoder and a classical decoder, with the quantum encoder subsequently employed as the concept generator. The specific configuration of the QVAE is outlined in Table 1. The binary cross-entropy loss function was applied, thus normalization operations were omitted during data preprocessing. The loss function comprises both reconstruction loss and KL divergence, with their balance managed by adjusting the hyperparameter β. This adjustment facilitates a trade-off between reconstruction accuracy and the smoothness of the latent space. It is critical that the reduction method for both components of the loss function remains consistent, either using the sum or mean, to avoid instability during training.

Table 1. Configuration of the concept generator in CD-QNN.

Parameter	Value
Number of qubits	6
Number of layers	10
Dataloader	Amplitude
Measurement operators	30
Training epochs	10
β	1

In the classical decoder, LeakyReLU was employed for activation functions to mitigate the vanishing gradient problem associated with the negative half-axis of ReLU [87]. The LeakyReLU function is defined as

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ \alpha x & \text{if } x \leq 0, \end{cases} \tag{26}$$

where α is a small positive value, set at 0.2 for this experiment. LeakyReLU introduces a slight slope in the negative region, effectively addressing the issue of zero gradients in ReLU activation functions. The classical decoder configuration ensures the output remains within the $[0, 1]$ range by incorporating a Sigmoid activation function in its final layer.

The feature extractor comprises two convolutional neural network layers, each utilizing $[10, 20]$ 3×3 convolutional kernels, followed by two fully connected layers containing $[160, 80]$ neurons, respectively. To enhance the generalization capability of the feature extractor, dropout layers with a parameter of 0.2 were included during training. The feature integrator employs a simple summation function without any trainable parameters.

The aforementioned configuration is not optimized. Therefore, a series of ablation studies were designed to further explore the impact of various modules and parameters on the CD-QNN model's performance. The following aspects were considered:

- Adjusting the β value allows observation of the balance between reconstruction accuracy and latent space smoothness. Increasing β enhances the constraint on the latent space, which may reduce reconstruction accuracy slightly but results in a smoother latent space.
- Modifying the number of measurements or layers in the QNN affects model performance. Reducing measurements can complicate latent space representation extraction, whereas increasing network layers enhances learning capacity but also adds training complexity.
- Introducing a scaling layer adjusts feature scales to improve model generalization and stability. Omitting the scaling layer may cause feature values to be excessively large or small, impacting model convergence speed and performance.
- Employing different dataloaders, such as rotation gate and amplitude dataloaders, alters the model's expressivity and learning dynamics, thereby influencing final performance.

- The choice between the two-stage optimization strategy proposed herein and the simultaneous optimization of the concept generator and feature extractor significantly impacts the training process.

Based on these considerations, nine sets of experiments were conducted, each altering only one or two configurations, as summarized in Table 2.

Table 2. Configurations of the ablation experiments for CD-QNN.

Index	Description
1	Baseline configuration
2	Increase β parameter in QVAE to 4
3	Reduce the number of measurements in the quantum concept generator to 6
4	Increase the number of layers in the quantum concept generator to 30
5	Without scaling layer
6	Use rotation gate dataloaders
7	Train concept generator simultaneously during feature extractor training
8	Increase the number of layers in the QNN to 30, and increase the number of training epochs to 30
9	Use rotation gate dataloaders and without scaling layer

6.2. Classification Accuracy

The classification accuracy of CD-QNN models was assessed, and the outcomes are depicted in Figure 3. This figure illustrates the accuracy trajectory throughout the training process, with the horizontal axis representing the number of training epochs and the vertical axis indicating accuracy. Upon comparing the results across various configurations, several conclusions can be inferred:

- Group 7 reveals that employing a mixed training strategy, where the concept generator is trained concurrently with the feature extractor, enhances the model's accuracy in the initial phases. Nevertheless, as training advances, the model eventually deteriorates. Experiments involving the concept generator with rotation gate dataloaders produced similar outcomes, demonstrating that irrespective of the model architecture, the mixed training strategy ultimately results in model failure. This underscores the significance of the two-stage training strategy, wherein the feature extractor is trained first, followed by the concept generator, to maintain model stability and efficacy.
- Omitting the scaling layer in the latent space influences the model's training speed. Specifically, the rate of accuracy improvement during training is diminished, potentially because the scaling layer assists in adjusting the feature space distribution, thereby expediting model convergence. Although the convergence rate is reduced, the CD-QNN without the scaling layer can still achieve over 95% accuracy with an increased number of training iterations.
- Aside from the aforementioned factors, modifications in other configurations exert relatively minor impacts on model accuracy, generally achieving approximately 95% accuracy. It is noteworthy that subsequent interpretability analyses can disclose significant differences in interpretability even between two models with identical accuracy. This finding further highlights the limitation of relying solely on a single metric to evaluate model performance and the necessity to consider various aspects, particularly the model's internal decision-making mechanisms.

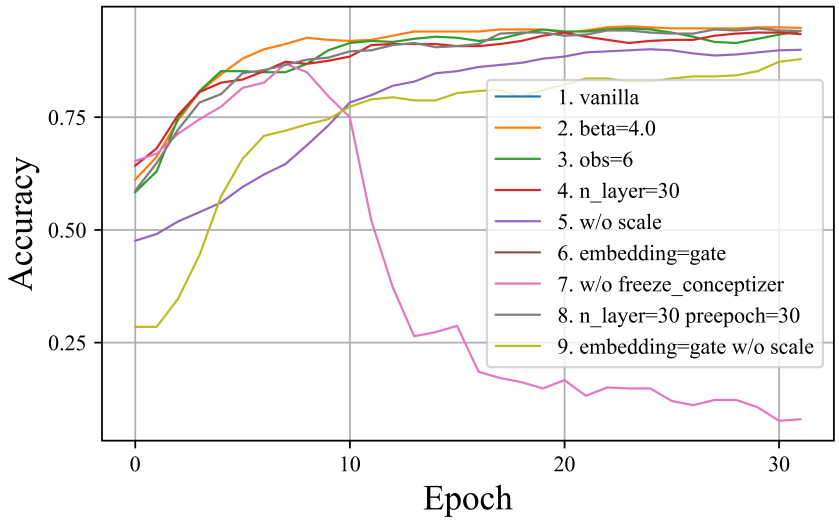

Figure 3. Comparison of classification accuracy for CD-QNN under different configurations.

6.3. Reconstruction Error

To further assess model performance, the reconstructions of 10 samples from Group 2 are presented in Figure 4. For a more detailed analysis, the reconstruction loss of various configurations was compared after the final training iteration. Reconstruction loss was quantified using mean squared error (MSE), as illustrated in Figure 5. Comparing these nine groups, the following conclusions were derived:

- None of the models exhibited overfitting, indicating they maintained robust generalization throughout the training process.
- Comparing Groups 1, 4, and 8, it was observed that increasing the number of layers in the quantum network enhances the model's expressivity, albeit at the cost of increased training resources and time.
- The comparison between Groups 1 and 2 revealed that increasing the weight coefficient β of the reconstruction error may lead to higher reconstruction loss, necessitating a balance in practical applications.
- The comparison between Groups 1 and 3 demonstrated that a 10-layer QNN, with only 180 trainable parameters, can effectively compress a 64-dimensional vector to 6 dimensions (through 6 measurements) while achieving good reconstruction accuracy.
- The comparisons among Groups 1, 5, 6, and 9 indicated that regardless of using rotation gate dataloaders or amplitude dataloaders, removing the scaling layer in the latent space can reduce reconstruction loss.
- The comparison between Groups 1 and 6 showed that both rotation gate dataloaders and amplitude dataloaders enable the QNN to effectively extract information.
- The comparison between Groups 1 and 7 indicated that training the concept generator concurrently with the feature extractor reduces reconstruction loss due to the increased number of iterations. However, the model eventually collapses, resulting in blurred reconstructed images.

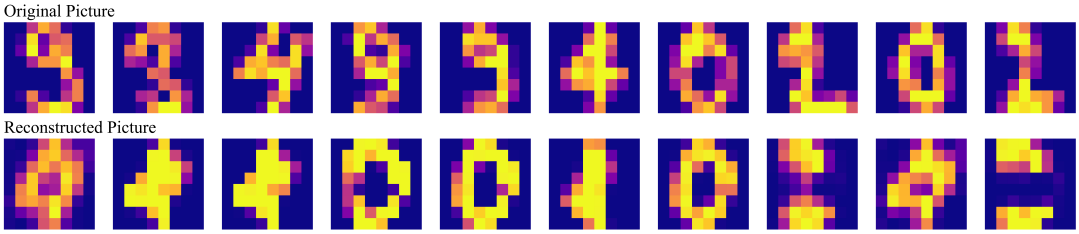

Figure 4. Reconstruction of images from the validation set in Group 2 (beta = 4).

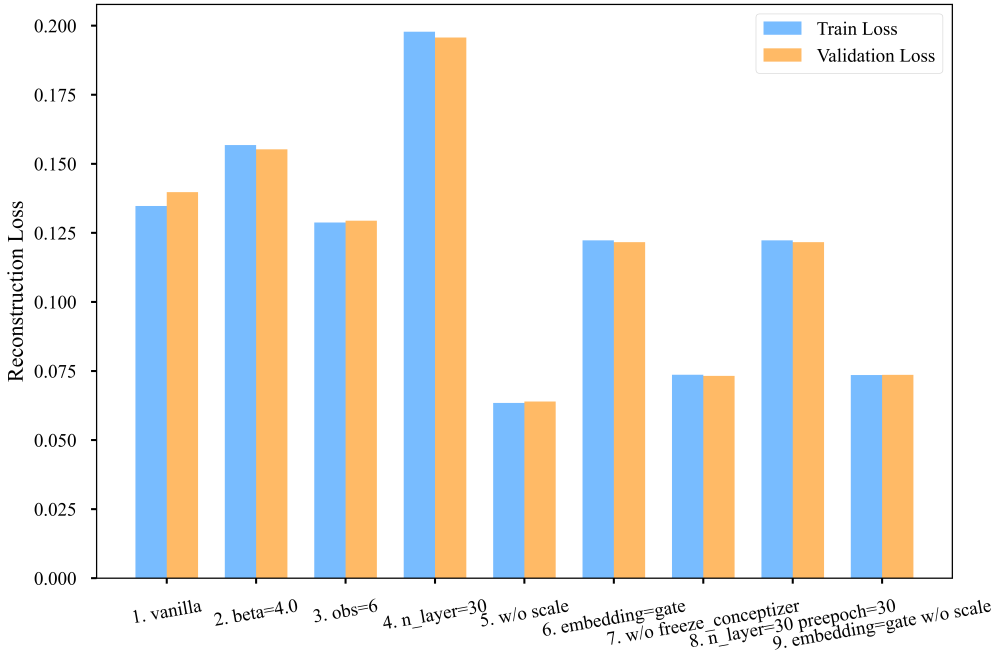

Figure 5. Comparison of reconstruction loss after the final training iteration.

6.4. Prototype Interpretation

The Group 2 configuration, which exhibited the highest classification accuracy, was selected for an in-depth case study. As depicted in Figure 6, for a sample predicted as the digit 9, the concept representations for six concepts generated by the concept generator were plotted, along with the corresponding representation coefficients obtained through the feature extractor.

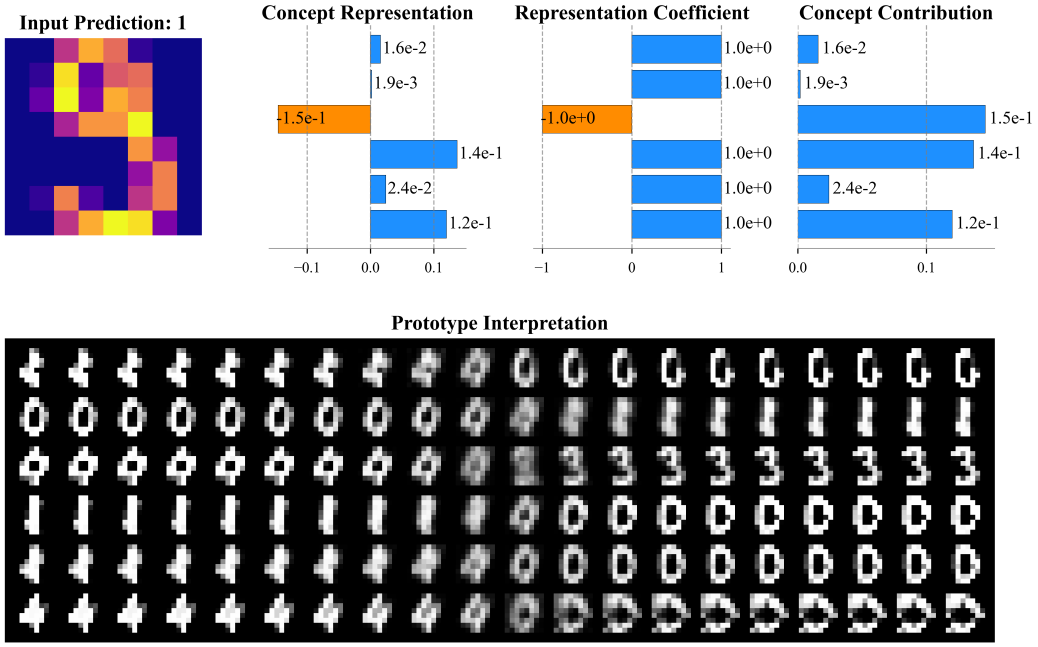

Figure 6. Prototype interpretation of an image from the validation set in Group 2 (beta = 4).

Given that continuous variation in the latent space of QVAE is meaningful, the operation of a concept dimension can be demonstrated by incrementally adjusting its value and decoding the reconstructed image. For instance, in this sample interpretation, the first concept represents a transition from the digit 4 to 6. The stronger the concept representation,

the closer the sample resembles the digit 4. The second concept depicts the transformation of a zero gradually breaking into a 3, and then further reducing the lower semicircle to form a 2.

While encouraging concept diversity, some overlap between concepts was observed. This overlap arises mainly due to the complexity of the data and inherent correlations among features in the dataset. Although the β-VAE framework encourages disentanglement by adjusting the β parameter, achieving complete independence among latent variables can be challenging, especially with highly interdependent features such as handwritten digits. This overlap reduces the clarity of individual contributions, making it more difficult to attribute specific effects to individual concepts in the model's decision-making process.

Additionally, certain concept interpretations remain challenging for human understanding. Some concepts may not correspond to easily interpretable transformations or features, which can hinder the user's ability to understand and trust the model's explanations. This suggests that when designing interpretable models, both the disentanglement and comprehensibility of the concepts should be considered to better facilitate human understanding of the model's decision-making process.

The interpretability of quantum machine learning models is influenced by several factors, including the training extent of the concept generator, the number of layers in the QNN, and the dataloader used. Careful adjustment and optimization of these factors can significantly enhance model interpretability, thereby increasing its reliability in practical applications.

7. Discussion and Conclusions

The design of the Concept-Driven Quantum Neural Network (CD-QNN) model can be comprehensively understood through a two-tier hybrid framework. At the concept generator level, a quantum-classical hybrid model, specifically a QVAE, is employed. Additionally, when the concept generator is viewed as a quantum model, the feature extractor and feature integrator form another layer of a quantum-classical hybrid model at a higher level. This design philosophy harnesses the complementary strengths of quantum and classical computing, thereby aiming to enhance overall performance.

CD-QNN exhibits considerable flexibility and configurability, allowing it to adapt to various input data types and task scenarios by selecting suitable components and configurations to achieve optimal performance. This adaptability enables CD-QNN to meet diverse application requirements, providing a robust tool for future research on the integration of quantum and classical computing.

The CD-QNN model signifies a paradigm shift in QNN design by underscoring the importance of interpretability. CD-QNN integrates the potent representational capabilities of QNNs with the interpretability of self-explanatory models by mapping input data to a human-understandable concept space and making decisions based on these concepts. This methodology offers intuitive explanations of the model's decision-making process, thereby enhancing the model's transparency and credibility.

The collaborative mechanisms among the feature extractor, concept generator, and feature integrator are designed to augment and balance the model's expressivity and interpretability. Emphasis is placed on concept disentanglement, with discussed methods aimed at effectively achieving the disentanglement and representation of concepts, thereby more effectively capturing key information in the data.

The loss function design underscores the significance of two-stage optimization in enhancing the effectiveness of explanations while balancing model prediction performance and interpretability. By considering classification loss, reconstruction loss, and the stability of explanations comprehensively, CD-QNN can maintain high predictive accuracy while offering clear and meaningful model explanations.

As quantum systems scale, maintaining interpretability becomes increasingly challenging. CD-QNN addresses this by focusing on extracting the most critical concepts and limiting the number of concept dimensions through dimensionality reduction techniques like

QVAE. Additionally, CD-QNN employs hierarchical approaches for very high-dimensional systems, where higher-level concepts are derived from simpler ones. Visualization tools such as t-SNE and PCA further aid in making high-dimensional concept spaces interpretable. By integrating domain-specific knowledge, the model aligns its decisions with recognizable phenomena, ensuring that interpretability remains within human cognitive limits even as system complexity grows. Future work will explore these strategies in larger quantum systems to validate their effectiveness at scale.

As the demand for model interpretability continues to increase, CD-QNN and its derivative technologies are anticipated to play a pivotal role in future quantum artificial intelligence research and applications. By providing intuitive explanations and maintaining high predictive accuracy, CD-QNN offers crucial support for the development of more reliable and interpretable quantum intelligent systems.

Author Contributions: Conceptualization, J.T. and W.Y.; methodology, J.T. and W.Y.; software, J.T.; validation, J.T. and W.Y.; formal analysis, J.T. and W.Y.; investigation, J.T.; resources, W.Y.; data curation, J.T.; writing—original draft preparation, J.T. and W.Y.; writing—review and editing, J.T. and W.Y.; visualization, J.T.; supervision, W.Y.; project administration, W.Y.; funding acquisition, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (Nos. 62372459, 62376282 and 91948303).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Shor, P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* **1997**, *26*, 1484–1509. [CrossRef]
2. Deutsch, D.; Jozsat, R. Rapid Solution of Problems by Quantum Computation. *Proc. R. Soc. London. Ser. A Math. Phys. Sci.* **1992**, *439*, 553–558.
3. Grover, L.K. A Fast Quantum Mechanical Algorithm for Database Search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing—STOC '96, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219. [CrossRef]
4. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [CrossRef]
5. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [CrossRef]
6. Otgonbaatar, S.; Datcu, M. Classification of Remote Sensing Images with Parameterized Quantum Gates. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
7. Riedel, M.; Cavallaro, G.; Benediktsson, J.A. Practice and Experience in Using Parallel and Scalable Machine Learning in Remote Sensing from HPC over Cloud to Quantum Computing. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 1571–1574. [CrossRef]
8. Sebastianelli, A.; Zaidenberg, D.A.; Spiller, D.; Le Saux, B.; Ullo, S.L. On Circuit-Based Hybrid Quantum Neural Networks for Remote Sensing Imagery Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *15*, 565–580. [CrossRef]
9. Zaidenberg, D.A.; Sebastianelli, A.; Spiller, D.; Le Saux, B.; Ullo, S.L. Advantages and Bottlenecks of Quantum Machine Learning for Remote Sensing. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 5680–5683. [CrossRef]
10. Farhi, E.; Neven, H. Classification with Quantum Neural Networks on near Term Processors. *arXiv* **2018**, arXiv:1802.06002.
11. McClean, J.R.; Boixo, S.; Smelyanskiy, V.N.; Babbush, R.; Neven, H. Barren Plateaus in Quantum Neural Network Training Landscapes. *Nat. Commun.* **2018**, *9*, 4812. [CrossRef]
12. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum Machine Learning. *Nature* **2017**, *549*, 195–202. [CrossRef] [PubMed]
13. Schuld, M.; Sinayskiy, I.; Petruccione, F. The Quest for a Quantum Neural Network. *Quantum Inf. Process.* **2014**, *13*, 2567–2586. [CrossRef]
14. Farhi, E.; Goldstone, J.; Gutmann, S. A Quantum Approximate Optimization Algorithm. *arXiv* **2014**, arXiv:1411.4028.

15. Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.H.; Zhou, X.Q.; Love, P.J.; Aspuru-Guzik, A.; O'Brien, J.L. A Variational Eigenvalue Solver on a Photonic Quantum Processor. *Nat. Commun.* **2014**, *5*, 4213. [CrossRef] [PubMed]
16. Kapoor, A.; Wiebe, N.; Svore, K. Quantum Perceptron Models. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Curran Associates, Inc.: Red Hook, NY, USA, 2016; Volume 29.
17. Cong, I.; Choi, S.; Lukin, M.D. Quantum Convolutional Neural Networks. *Nat. Phys.* **2019**, *15*, 1273–1278. [CrossRef]
18. Lloyd, S.; Weedbrook, C. Quantum Generative Adversarial Learning. *Phys. Rev. Lett.* **2018**, *121*, 040502. [CrossRef]
19. Romero, J.; Olson, J.P.; Aspuru-Guzik, A. Quantum Autoencoders for Efficient Compression of Quantum Data. *Quantum Sci. Technol.* **2017**, *2*, 045001. [CrossRef]
20. Doshi-Velez, F.; Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv* **2017**, arXiv:1702.08608.
21. Lipton, Z.C. The Mythos of Model Interpretability. *Commun. ACM* **2018**, *61*, 36–43. [CrossRef]
22. Adadi, A.; Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. [CrossRef]
23. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *arXiv* **2019**, arXiv:1910.10045.
24. Molnar, C. *Interpretable Machine Learning*; Lulu.com: Morrisville, NC, USA, 2020.
25. Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C.J.; Müller, K.R. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. *Proc. IEEE* **2021**, *109*, 247–278. [CrossRef]
26. Tim, M. Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artif. Intell.* **2019**, *267*, 1–38. [CrossRef]
27. Rudin, C. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef] [PubMed]
28. Alvarez Melis, D.; Jaakkola, T. Towards Robust Interpretability with Self-Explaining Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.
29. Koh, P.W.; Nguyen, T.; Tang, Y.S.; Mussmann, S.; Pierson, E.; Kim, B.; Liang, P. Concept Bottleneck Models. In Proceedings of the 37th International Conference on Machine Learning (PMLR), Virtual, 13–18 July 2020; pp. 5338–5348.
30. Chen, Z.; Bei, Y.; Rudin, C. Concept Whitening for Interpretable Image Recognition. *Nat. Mach. Intell.* **2020**, *2*, 772–782. [CrossRef]
31. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should i Trust You?" Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144. [CrossRef]
32. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-Cam: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
33. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
34. Hastie, T.J. Generalized Additive Models. In *Statistical Models in S*; Routledge: New York, NY, USA, 2017; pp. 249–307.
35. Breiman, L. *Classification and Regression Trees*; Routledge: New York, NY, USA, 2017.
36. Quinlan, J.R. *C4. 5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.
37. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
38. Samek, W.; Montavon, G.; Vedaldi, A.; Hansen, L.K.; Müller, K.R. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Springer Nature: Berlin, Germany, 2019; Volume 11700.
39. Holzinger, A. From Machine Learning to Explainable AI. In Proceedings of the 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA), Kosice, Slovakia, 23–25 August 2018; pp. 55–66. [CrossRef]
40. Lipton, Z.C. The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability Is Both Important and Slippery. *Queue* **2018**, *16*, 31–57. [CrossRef]
41. Bau, D.; Zhu, J.Y.; Strobelt, H.; Lapedriza, A.; Zhou, B.; Torralba, A. Understanding the Role of Individual Units in a Deep Neural Network. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 30071–30078. [CrossRef]
42. Burge, I.; Barbeau, M.; Garcia-Alfaro, J. A Quantum Algorithm for Shapley Value Estimation. *arXiv* **2023**, arXiv:2301.04727.
43. Heese, R.; Gerlach, T.; Mücke, S.; Müller, S.; Jakobs, M.; Piatkowski, N. Explaining Quantum Circuits with Shapley Values: Towards Explainable Quantum Machine Learning. *arXiv* **2023**, arXiv:2301.09138.
44. Mercaldo, F.; Ciaramella, G.; Iadarola, G.; Storto, M.; Martinelli, F.; Santone, A. Towards Explainable Quantum Machine Learning for Mobile Malware Detection and Classification. *Appl. Sci.* **2022**, *12*, 12025. [CrossRef]
45. Pira, L.; Ferrie, C. On the Interpretability of Quantum Neural Networks. *arXiv* **2024**, arXiv:2308.11098. [CrossRef]
46. Steinmüller, P.; Schulz, T.; Graf, F.; Herr, D. eXplainable AI for Quantum Machine Learning. *arXiv* **2022**, arXiv:2211.01441.
47. Gupta, M.; Cotter, A.; Pfeifer, J.; Voevodski, K.; Canini, K.; Mangylov, A.; Moczydlowski, W.; Van Esbroeck, A. Monotonic Calibrated Interpolated Look-up Tables. *J. Mach. Learn. Res.* **2016**, *17*, 1–47.
48. Lou, Y.; Caruana, R.; Gehrke, J.; Hooker, G. Accurate Intelligible Models with Pairwise Interactions. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 623–631. [CrossRef]

49. Cowan, N. The Magical Mystery Four: How Is Working Memory Capacity Limited, and Why? *Curr. Dir. Psychol. Sci.* **2010**, *19*, 51–57. [CrossRef]
50. Ciliberto, C.; Herbster, M.; Ialongo, A.D.; Pontil, M.; Rocchetto, A.; Severini, S.; Wossnig, L. Quantum Machine Learning: A Classical Perspective. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **2018**, *474*, 20170551. [CrossRef] [PubMed]
51. Wang, X.; Du, Y.; Luo, Y.; Tao, D. Towards Understanding the Power of Quantum Kernels in the NISQ Era. *Quantum* **2021**, *5*, 531. [CrossRef]
52. Qian, Y.; Wang, X.; Du, Y.; Wu, X.; Tao, D. The Dilemma of Quantum Neural Networks. *arXiv* **2021**, arXiv:2106.04975. [CrossRef] [PubMed]
53. Benedetti, M.; Lloyd, E.; Sack, S.; Fiorentini, M. Parameterized Quantum Circuits as Machine Learning Models. *Quantum Sci. Technol.* **2019**, *4*, 043001. [CrossRef]
54. McClean, J.R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The Theory of Variational Hybrid Quantum-Classical Algorithms. *New J. Phys.* **2016**, *18*, 023023. [CrossRef]
55. Schuld, M.; Sinayskiy, I.; Petruccione, F. Simulating a Perceptron on a Quantum Computer. *Phys. Lett. A* **2015**, *379*, 660–663. [CrossRef]
56. Henderson, M.; Shakya, S.; Pradhan, S.; Cook, T. Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits. *Quantum Mach. Intell.* **2020**, *2*, 2. [CrossRef]
57. Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised Learning with Quantum-Enhanced Feature Spaces. *Nature* **2019**, *567*, 209–212. [CrossRef] [PubMed]
58. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv* **2014**, arXiv:1312.6034.
59. Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; Wattenberg, M. SmoothGrad: Removing Noise by Adding Noise. *arXiv* **2017**, arXiv:1706.03825.
60. Lundberg, S.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. *arXiv* **2017**, arXiv:1705.07874. [CrossRef]
61. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
62. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893. [CrossRef]
63. Lowe, D. Object Recognition from Local Scale-Invariant Features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157. [CrossRef]
64. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef]
65. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2022**, arXiv:1312.6114.
66. Khoshaman, A.; Vinci, W.; Denis, B.; Andriyash, E.; Sadeghi, H.; Amin, M.H. Quantum Variational Autoencoder. *Quantum Sci. Technol.* **2018**, *4*, 014001. [CrossRef]
67. LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; Huang, F. A Tutorial on Energy-Based Learning. In *Predicting Structured Data*; MIT Press: Cambridge, MA, USA, 2006; Volume 1.
68. Amin, M.H.; Andriyash, E.; Rolfe, J.; Kulchytskyy, B.; Melko, R. Quantum Boltzmann Machine. *Phys. Rev. X* **2018**, *8*, 021050. [CrossRef]
69. Zoufal, C.; Lucchi, A.; Woerner, S. Variational Quantum Boltzmann Machines. *Quantum Mach. Intell.* **2021**, *3*, 7. [CrossRef]
70. Tian, J.; Sun, X.; Du, Y.; Zhao, S.; Liu, Q.; Zhang, K.; Yi, W.; Huang, W.; Wang, C.; Wu, X.; et al. Recent Advances for Quantum Neural Networks in Generative Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12321–12340. [CrossRef] [PubMed]
71. Dallaire-Demers, P.L.; Killoran, N. Quantum Generative Adversarial Networks. *Phys. Rev. A* **2018**, *98*, 012324. [CrossRef]
72. Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J.M.; Gambetta, J.M. Hardware-Efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets. *Nature* **2017**, *549*, 242–246. [CrossRef]
73. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational Quantum Algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]
74. Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum Circuit Learning. *Phys. Rev. A* **2018**, *98*, 32309. [CrossRef]
75. Pérez-Salinas, A.; Cervera-Lierta, A.; Gil-Fuster, E.; Latorre, J.I. Data Re-Uploading for a Universal Quantum Classifier. *Quantum* **2020**, *4*, 226. [CrossRef]
76. Schuld, M.; Petruccione, F. *Supervised Learning with Quantum Computers*; Quantum Science and Technology; Springer International Publishing: Cham, Switzerland, 2018. [CrossRef]
77. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.P.; Glorot, X.; Botvinick, M.M.; Mohamed, S.; Lerchner, A. Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017.
78. Zhu, Q.; Su, J.; Bi, W.; Liu, X.; Ma, X.; Li, X.; Wu, D. A Batch Normalized Inference Network Keeps the KL Vanishing Away. *arXiv* **2020**, arXiv:2004.12585.
79. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: New York, NY, USA, 2013.

80. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding Deep Learning (Still) Requires Rethinking Generalization. *Commun. ACM* **2021**, *64*, 107–115. [CrossRef]
81. Hornik, K. Approximation Capabilities of Multilayer Feedforward Networks. *Neural Netw.* **1991**, *4*, 251–257. [CrossRef]
82. Tishby, N.; Zaslavsky, N. Deep Learning and the Information Bottleneck Principle. In Proceedings of the 2015 IEEE Information Theory Workshop (ITW), Jerusalem, Israel, 26 April–1 May 2015; pp. 1–5. [CrossRef]
83. Gao, X.; Anschuetz, E.R.; Wang, S.T.; Cirac, J.I.; Lukin, M.D. Enhancing Generative Models via Quantum Correlations. *Phys. Rev. X* **2022**, *12*, 021037. [CrossRef]
84. Du, Y.; Tu, Z.; Yuan, X.; Tao, D. Efficient Measure for the Expressivity of Variational Quantum Algorithms. *Phys. Rev. Lett.* **2022**, *128*, 080506. [CrossRef] [PubMed]
85. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.
86. Reddi, S.J.; Kale, S.; Kumar, S. On the Convergence of Adam and Beyond. *arXiv* **2019**, arXiv:1904.09237.
87. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In Proceedings of the 30th International Conference on Machine Learning (ICML 2013), Atlanta, GA, USA, 16–21 June 2013; Volume 30, p. 3.