

# QRF: Implicit Neural Representations with Quantum Radiance Fields

Yuan-Fu Yang  
National Tsing Hua University  
yfyangd@gmail.com

Min Sun  
National Tsing Hua University  
summin@ee.nthu.edu.tw

## Abstract

*Photorealistic rendering of real-world scenes is a tremendous challenge with a wide range of applications, including MR (Mixed Reality), and VR (Mixed Reality). Neural networks, which have long been investigated in the context of solving differential equations, have previously been introduced as implicit representations for Photorealistic rendering. However, realistic rendering using classic computing is challenging because it requires time-consuming optical ray marching, and suffer computational bottlenecks due to the curse of dimensionality. In this paper, we propose Quantum Radiance Fields (QRF), which integrate the quantum circuit, quantum activation function, and quantum volume rendering for implicit scene representation. The results indicate that QRF not only takes advantage of the merits of quantum computing technology such as high speed, fast convergence, and high parallelism, but also ensure high quality of volume rendering.*

## 1. Introduction

**Neural Scene Representations.** Traditional 3D computer vision pipelines use multi-view stereo algorithms to estimate sparse point clouds, camera poses, and texture meshes from 2D input views. However, re-rendering these scene representations does not achieve photorealistic image quality. In contrast to these explicit scene representations, implicit scene representations produce significantly higher quality renderings and can be supervised directly with 3D data by using neural networks.

Nevertheless, current neural networks consisting of MLPs are incapable of modeling signals with fine detail, and cannot accurately model high-frequency information and higher-order derivatives even with dense supervision. In addition, realistic rendering of real-world scenes using classic computer graphics techniques is challenging because it requires the difficult step of capturing detailed appearance and geometric models. Existing methods in practice often show blurry renderings due to limited network capacity. Synthesizing high-resolution imagery from these representations often requires time-consuming optical ray marching, and suffer computational bottlenecks due to the curse of dimensionality.



Figure 1: Volume rendering of Drums with training 50k iterations. Compared with NeRF baseline [13] and AutoInt [17], quantum radiance fields has faster convergence, higher rendering efficiency, and higher rendering quality under the quantum integration.

**Quantum Neural Networks.** The field of artificial neural networks has benefited greatly from recent developments in quantum computers in recent years. In particular, quantum neural networks (QNNs), a class of quantum algorithms which exploit qubits for creating trainable neural networks. Quantum computing can provide potentially exponential speedups due to their ability to perform massively parallel computations on the superposition of quantum states. The quantum neural networks for implicit representations that are capable of overcoming the computational challenges faced by conventional techniques performed on classical computers.

In this study, the neural rendering is performed with quantum circuits (as shown in Figure 2). The parameters calculation run on quantum computers, whose computation speeds are many times higher than supercomputers. In addition, we proposed a Quantum Volume Rendering based on quantum integration, which has been implemented on real quantum hardware Borealis. Our proposed Quantum Volume Rendering performs the fastest among various rendering tasks and offers speed-related advantages over implementing conventional integration on digital computers, such as Monte Carlo integration, as it involves computing exponentials. To summarize, the contributions of our work include:

- (1) The first QNNs-based system capable of rendering photorealistic novel views, hundreds of times faster than conventional neural networks.
- (2) We present Quantum Radiance Fields (QRF) that consists of a set of quantum implicit fields, where for each quantum circuit, encoding circuit are learned to encode local properties for high-quality rendering.
- (3) We leverage the quantum activation function, which can better represent details in signals than classical

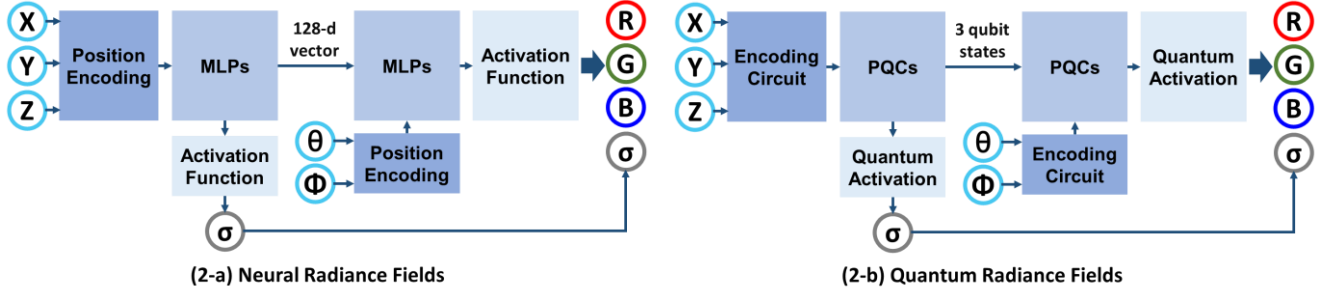


Figure 2: (2-a) NeRF architecture. Given a 3D position ( $x, y, z$ ), viewing direction ( $\theta, \phi$ ), NeRF produces static and transient colors ( $r, g, b$ ) and transparency values ( $\sigma$ ). (2-b): our QRF architecture replaces the same task with encoding circuit, parameterized quantum circuits, and quantum activation.

activation function for implicit neural representations.

(4) We propose the Quantum Volume Rendering, a quantum algorithm for numerical integration, which is fundamentally better than classical Monte Carlo integration in volume rendering.

The remainder of this paper is organized as follows. In Section 2, we introduce the neural scene representations, neural radiance fields, and quantum neural networks. In Section 3, the implicit neural representations with quantum radiance fields are introduced. In Section 4, we present results on 2D image regression and 3D scene reconstruction. For each task, we demonstrate the benefits of using circuit based QRF. Section 5 summarizes this work and briefly discusses possible future extensions.

## 2. Related Work

**Neural Scene Representation.** To model objects in a scene, many different scene geometry representations have been proposed. They can be divided into explicit and implicit representations. Explicit scene representations describe scenes as a collection of geometric primitives, and the output it produce can be classified into voxel-based [1][2], point-based [3][4], and mesh-based representations [5][6]. While explicit scene representations enable rapid generation of novel views, they are fundamentally limited by the internal resolution of their representations, which can lead to blurry outputs for high-frequency content.

To circumvent the above problem, many works have explored the potential of implicit neural scene representations directly infer outputs from a continuous input space. In contrast to explicit neural scene representations, implicit neural scene representations promise 3D structure-aware, continuous, memory-efficient representations of shape parts, objects, or scenes [7][8][9]. These representations use neural network to implicitly define objects or scenes, and can be supervised directly using 3D data, such as point clouds, or with 2D multi-view images [10][11][12].

**Neural Radiance Fields.** Recent work on Neural Radiation Fields (NeRF) has shown how neural network

can be used to learn an implicit volumetric representation of the scene and encode complex 3D environments that can be rendered realistically from novel viewpoints [13]. However, NeRF needs to sample a large number of points along the ray for color accumulation to achieve high quality rendering.

Numerous works were developed with the purpose of speeding up NeRF. Neural Sparse Voxel Fields (NSVF) [15] speed up NeRF’s rendering using classical techniques like empty space skipping and early ray termination. MetaNeRF [16] proposed to apply standard meta-learning algorithms to learn the initial weight parameters of the MLPs. A meta-learned weight initialization leads to faster convergence and allows better reconstruction quality from fewer supervised views during test-time optimization. AutoInt [17] introduced an automatic integration framework that learns closed-form integral solutions that reduce the number of evaluations along the ray when raymarching through a NeRF. DOnERF [18] speed up inference by reducing the number of required samples along the ray. FastNeRF [19] proposed a graphics-inspired factorization that can be compactly cached and subsequently queried to compute pixel values in rendered images. KiloNeRF [20] represents a scene by thousands of small MLPs instead of a single large-capacity MLP representing the entire scene, so smaller and faster evaluation MLPs can be used. DS-NeRF [21] leveraged depth as an additional source of supervision to regularize the geometry learned by NeRF and improve the training of NeRF. Cheng Sun and Min Sun et al. [22] proposed a super-fast convergence approach to reconstructing the per-scene radiance field from a set of images that capture the scene with known poses. Their method directly optimizes the voxel grid and achieves super-fast convergence in per-scene optimization, reducing training time from hours to 15 minutes with comparable quality to NeRF.

**Quantum Neural Networks.** QNNs is a relatively new field that blends the computational advantages brought by quantum computing and advances beyond classical computation [23]. QNNs not only bring more efficient

algorithm performance, but are also able to find the global minimum in the sought solution with higher probability [24]. The main principles of quantum computing are those inherited from quantum physics, such as superposition, entanglement, and interference. A qubit system can hold multiple bits of information simultaneously, and thus also enables massive parallelism [25].

In this paper, we present Quantum Radiance Fields (QRF), which integrates the quantum circuit and quantum activation function with implicit representation. Then, a quantum computing assisted generative training process followed by supervised discriminative training is used to train the QRF model.

### 3. Method

To achieve real-time movie-quality rendering of compact neural representations for generated content, we employ the QNNs-based neural raymarching scheme of NeRF [13]. NeRF uses MLP to encode densities and colors at any continuous 3D position in the scene. We are inspired by DNeRF [18] and represent the scene using a compact local sampling strategy. This strategy enables the raymarching-based neural representation to consider only important samples around the surface region, further reducing the usage of Qubit on simulated environments and quantum computers. We review the baseline model of NeRF in Section 3.1, describe our Quantum Radiation Fields in Section 3.2, introduce Quantum Activation Function in Section 3.3 and Quantum Volume Rendering in Section 3.4.

#### 3.1. Baseline Model

In the neural radiation field, the scene is represented by a neural network  $f_\theta$  with parameter  $\theta_n$  to capture the volume 3D representation. NeRF's neural network  $f_\theta: (p, d) \rightarrow (c, \sigma)$  maps 3D position  $p \in \mathbb{R}^3$  and light direction  $d \in \mathbb{R}^2$  to color value  $c$  and transparency  $\sigma$ . The architecture of  $f_\theta$  is chosen such that only the color  $c$  depends on the viewing direction  $d$ . This encourages the learning of consistent geometry. In the deterministic preprocessing step,  $x$  and  $d$  are transformed via a positional encoding  $\gamma$ , which promotes the learning of high frequency details. To render a single image pixel, a ray is cast from the center of the camera, through that pixel and into the scene. We denote the direction of this ray as  $d$ . Multiple 3D positions  $(p_1, \dots, p_k)$  are then sampled along the ray between the near and far boundaries defined by the camera parameters. The neural network  $f_\theta$  is evaluated at each position  $p_i$  and ray direction  $d$  to produce color  $c_i$  and transparency  $\sigma_i$ . These intermediate outputs are then integrated as follows to produce the final pixel color  $\hat{c}$ :

$$\hat{c} = \sum_i^K T_i (1 - e^{(-\sigma_i \delta_i)}) c_i \quad (1)$$

where  $T_i = e^{-\sum_{j=i}^{i-1} \sigma_j \delta_j}$  is the transmittance and  $\delta_j = (p_{i+1} - p_i)$  is the distance between samples. Since  $f_\theta$  is dependent on ray direction, NeRF can model viewpoint-dependent effects such as specular reflections, a key dimension in which NeRF improves traditional 3D reconstruction methods.

#### 3.2. Quantum Radiance Fields

The quantum radiance fields is implemented by various quantum circuits built into the continuous-variable architecture. It consists of three consecutive parts (as shown in Figure 2). An encoding circuit encodes the classical data into the states of the qubits, followed by a parameterized quantum circuits (PQCs), which is used to transform these states to their optimal location on the Hilbert space. Finally, quantum activation is used to make nonlinear mapping to the input and add some nonlinear factors to neural networks so that neural networks can better represent the color and transparency of each position along the ray.

##### 3.2.1 Encoding Circuit

The encoding circuit is used to encode the classical data into the physical states of Hilbert space for quantum computing [26], which is critical to the success of quantum neural networks. Quantum encoding can be thought of as loading a data point  $x \in X$  from memory into a quantum state so that it can be processed by a QNN. The loading is accomplished by encoding from the set  $X$  to the  $n$ -qubit quantum state  $D_n$ . Many QNN papers [27][28][29] proposed wavefunction encoding with  $n = \log_2 2N$ . This provides an exponential space savings at the cost of an exponential increasing in time. That is, a quantum state of  $\log_2 2N$  qubits can represent a data point with  $N$  features, but such a quantum state takes time  $O(2^n)$  to prepare. Some recent authors [30][31][32] have considered an angle encoding which can efficiently encode classical data into quantum state.

**Angle Encoding.** Angle encoding makes use of rotation gates to encode classical information  $x_k \in \mathbb{R}^N$  without any normalization condition. Angle encoding can be constructed using a single rotation with angle  $\theta_k$  (normalized to be in  $[-\pi, \pi]$ ) for each qubit, and can therefore encode  $N$  features with  $N$  qubits. Angle encoding consists in the following transformation:

$$S_k |0\rangle = \bigotimes_{k=0}^N \cos(\theta_k) |0\rangle + \sin(\theta_k) |1\rangle \quad (2)$$

where the circuit starts with the  $|0\rangle$  state, encodes a data point  $x_k$  using a circuit  $S_k$ .  $N$  is the number of qubits, which used for encoding is equal to the dimension of vector  $x_k$ . Angle encoding can be very easily constructed and has a depth of only 1. The main advantage of angle encoding is that it is very efficient in terms of operations: Only a

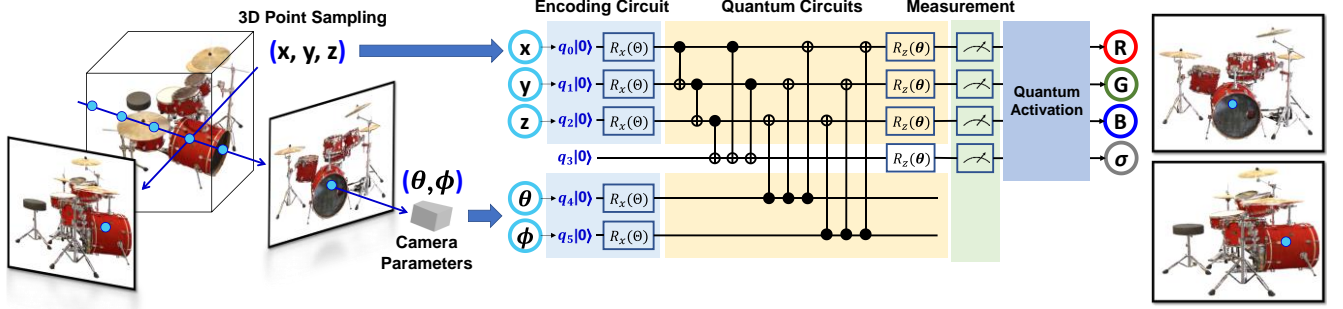


Figure 3: Quantum Radiance Fields (QRF) with encoding circuits and quantum circuits produces colors (r, g, b) and transparency values ( $\sigma$ ) given a 3D position ( $x, y, z$ ) and viewing direction ( $\theta, \phi$ ). Similar to the NeRF architecture, QRF enforces that the predicted  $\sigma$  is independent of view direction. Note that this schematic is a simplified quantum circuit with only 4 rotation gates around the z axis.

constant number of parallel operations are needed regardless of how many data values need to be encoded. This is not optimal from a qubit point of view since each input vector component requires one qubit [52].

**Dense Angle Encoding.** Angle encoding can be slightly generalized to encode two features per qubit by exploiting the relative phase degree of freedom [53]. We refer to this as the dense angle encoding and include a definition below:

$$|x\rangle = \bigotimes_{k=1}^{N/2} \cos(\theta_{2k}) |0\rangle + e^{2\pi i x_{2k}} \sin(\pi \theta_{2k}) |1\rangle \quad (3)$$

where  $x$  is a feature vector  $x = [x_1, \dots, x_N]^T \in \mathbb{R}^N$ , the dense angle encoding maps  $x \rightarrow E(x)$ . Dense encoding is derived by extending the above formula into two features using relative phase degrees of freedom. It exploits the additional property of relative phase qubits to encode  $N$  data points using only  $N/2$  qubits.

### 3.2.2 Parametrized Quantum Circuits.

Parametrized Quantum Circuit is composed of a set of parameterized single and controlled single qubit gates. The parameters are iteratively optimized by a classical optimizer to attain a desired input-output relationship. A block-diagonal approximation to the Fubini-Study metric tensor of a PQC can be evaluated on quantum hardware. In general, an  $n$  qubits PQC can be written as:

$$u(\hat{\theta})|\varphi_0\rangle = \left( \prod_{\ell=1}^k W_{\ell} u_{\ell}(\theta_{\ell}) \right) |\varphi_0\rangle \quad (4)$$

where  $\varphi_0$  is the initial quantum state,  $m$  is the maximum circuit depth,  $W_{\ell}$  is the non-parameterized quantum gate at  $\ell$ -th layer,  $u_{\ell}(\theta_{\ell})$  is the parametrized quantum gate with parameters  $\{\theta_0, \theta_1, \dots, \theta_k\}$  at  $\ell$ -th layer, which is a sequence consisting of parameterized qubit gates. Herein, the form of  $u_{\ell}(\theta_{\ell})$  is variable and accords with any physical constraint such as highly limited connectivity between physical qubits.

To achieve better entanglement of the qubits before appending nonlinear operations, the  $n$  qubits PQC has  $n$  repeated layers in our model. In order to provide

computational speedup by orchestrating constructive and destructive interference of the amplitudes in quantum computing, we constructed  $m$  rotation gates on the  $n$  qubits PQC as our basic quantum circuit, which can be written as:

$$\left( \prod_{\ell=1}^n \left( \bigotimes_{j=0}^m CNOT_{i,i+1} R(\theta_{i+n \times j}) \right) \right) \quad (5)$$

where  $CNOT_{i,i+1}$  represents  $CNOT$  gate as the control qubit.  $R(\theta_{i+n \times j})$  represents the rotation gate along each of the X, Y, and Z-axis.  $\theta_{i+n \times j}$  is adjustable parameter of rotation gates  $R$ . With the Pauli matrix, we can define single-qubit rotation along each of the X, Y, and Z-axis as:

$$R_x(\theta) = e^{-i\frac{\theta}{2}\sigma_x} = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \quad (6)$$

$$R_y(\theta) = e^{-i\frac{\theta}{2}\sigma_y} = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \quad (7)$$

$$R_z(\theta) = e^{-i\frac{\theta}{2}\sigma_z} = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \quad (8)$$

where  $\{\sigma_x, \sigma_y, \sigma_z\}$  is Pauli matrices. The operation of  $R(\theta)$  can be modified by changing parameters  $\theta$ . Thus, the output state can be optimized to approximate the wanted state. By optimizing the parameters, the general PQC tries to approximate arbitrary states so that it can be used for different specific molecules. The goal of PQC is to solve an optimization problem encoded into a cost function:

$$\theta^* = \underset{\theta \in \mathcal{C}}{\operatorname{argmin}} (\langle \psi(\theta) | H | \psi(\theta) \rangle) \quad (9)$$

where  $H$  is the Hamiltonian with the ground energy to seek. As parameters  $\theta$  are continuous, many gradient-based optimization algorithms can be used to find the optimal ones. Figure 3 shows an example of Parametrized Quantum Circuit with  $n=6$  and  $m=4$ . Four qubits use the rotation gate  $R(\theta)$  by the angle  $\theta$  around z-axis on the Hilbert space, and  $CNOT$  gate is used for 2 specific qubits.

### 3.3. Quantum Activation Function

Recent implicit neural representations are built on ReLU-based multilayer perceptron. These architectures lack the capacity to represent the fine details in the underlying signal, and they often do not represent the derivative of the target signal well. This is partly due to the fact that ReLU networks are piecewise linear, and their second derivatives are zero everywhere, so they cannot model the information contained in the higher-order derivatives of natural signals. To address these limitations, we leverage quantum activation functions, which can better represent details in signals than ReLU-MLPs for implicit neural representations.

We apply a multi-step quantum approach by selecting the ReLU's solution for positive values  $R(z)_{ReLU}$ , and the LReLU's solution for negative values  $R(z)_{LReLU}$  [54]. By applying the quantum principle of entanglement, the tensor product of the two candidate Hilbert state spaces from  $H_{ReLU}$  and  $H_{LReLU}$  was performed as:

$$H_{ReLU} \otimes H_{LReLU} \quad (10)$$

The quantum entanglement in Eq. 10 allows to overcome the limitation of the ReLU being dying for negative inputs. The resulting state in the blended system is described by:

$$|\varphi\rangle_{ReLU} \otimes |\varphi\rangle_{LReLU} \quad (11)$$

In an entangled or inseparable state, the formulation of product states of Quantum ReLU (QReLU) can be generalized as:

$$|\varphi\rangle_{QReLU} = \sum_{ReLU, LReLU} |0|1\rangle_{ReLU} \otimes |0|1\rangle_{LReLU} \quad (12)$$

where keeping output for positive values in the QReLU, but with the added novelty of the entangled solution for negative values. This fits complicated signals, such as natural images and 3D shapes, and their derivatives robustly.

### 3.4. Quantum Volume Rendering

Conventional implicit neural representations represent a scene as an implicit function  $F_\theta(p, v) \rightarrow (c, \omega)$ , where  $\theta$  are parameters of an underlying neural network [13]. It evaluate a volume rendering integral to compute the color of camera ray  $p(z) = p_0 + z \cdot v$  as:

$$c(p_0, v) = \int_0^\infty \omega(p(z)) \cdot c(p(z), v) dz \quad (13)$$

where  $\int_0^\infty \omega(p(z)) dz = 1$ ,  $c$  is the scene color,  $w$  is the probability density at spatial location  $p$  and ray direction  $v$ .  $c(p_0, v)$  describes the scene color  $c$  and its probability density  $\omega$  at spatial location  $p$  and ray direction  $v$ .

Volume rendering methods estimate the integral

$c(p_0, v)$  by densely sampling points on each camera ray and accumulating the colors and densities of the sampled points into a 2D image as:

$$c(p_0, v) \approx \sum_{i=n}^N \left( \prod_{j=1}^{i-1} \alpha(z_j, \Delta_i) \right) \cdot (1 - \alpha(z_i, \Delta_i)) \cdot c(p(z_i), v) \quad (14)$$

where  $\alpha(z_i, \Delta_i) = \exp(-\sigma(p(z_i), v))$ , and  $\{\sigma(p(z_i))\}_{i=1}^N$  are the colors and the volume densities of the sampled points.

Although volume rendering offer unprecedented image quality, they are also extremely slow and memory inefficient. This is because volume rendering methods need to sample a large number of points along the rays for color accumulation to achieve high quality rendering. Previous works only considered non-empty voxels for raymarching and reduce the number of samples per ray [15][18]. However, for scenes with high depth and complexity, this works will result in longer evaluation time and lower rendering quality.

Volume rendering is essentially a numerical integration problem in each pixel, which is commonly done by Monte Carlo integration on classical computers. In this paper, we propose the quantum ray tracing, a quantum algorithm for numerical integration that is fundamentally superior to classical Monte Carlo integration. Furthermore, we apply Grover's search [33] to design clever algorithms to take full advantage of quantum parallelism. Given a ray tracing oracle that implements the following transformations:

$$O_f(pixel, channel): \sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{j=0}^{N-1} x_j |j\rangle |f(j)\rangle \quad (15)$$

where  $pixel$  and  $channel$  (R, G or B) are classical parameters,  $j$  plays the role of ray identity, and  $f(j)$  is a real number that stands for the ray energy. In the rest of this paper  $f$  is specified as the function that maps ray  $j$  to ray energy. The oracle can trace  $N = 2n$  paths simultaneously, and the final color we hope to write to the corresponding pixel and channel is the average of those energies  $\frac{1}{N} \sum_{j=0}^{N-1} f(j)$ . Suppose those real numbers  $f(j)$  are stored in a fixed-point format with integer bit length  $b_0$  and total bit length  $b$ , we can transfer the estimation problem of Eq. 15 into quantum counting by constructing a Boolean function:

$$g(j, k) = \begin{cases} 1, & f(j) \geq 2^{b_0-b_k} \\ 0, & f(j) < 2^{b_0-b_k} \end{cases} \quad (16)$$

where  $k = (0, 1, \dots, 2^b - 1)$ . The phase oracle  $O_g$  for  $g$  in Grover's search [55] as:

$$O_g: \sum_{j,k} |j\rangle |k\rangle \rightarrow \sum_{j,k} (-1)^{g(j,k)} |j\rangle |k\rangle \quad (17)$$



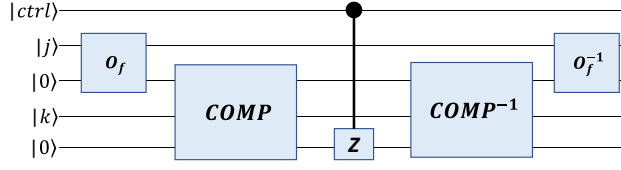


Figure 4: The construction of controlled- $O_g$

To construct  $O_g$ , we need a comparison gate that performs the comparison operation  $COMP$  on the two integers  $2^{b-b_0}f(j)$  and  $k$ ,

$$COMP: \sum_{j,k} |f(j)\rangle |k\rangle \rightarrow \sum_{j,k} |f(j)\rangle |k\rangle |g(j,k)\rangle \quad (18)$$

The  $O_g$  gate can be constructed as Figure 4. The quantity  $\sum_{j,k} g(j,k)$  can be estimated by quantum counting algorithm. In the paper we assume one call to  $O_f$  quantum ray tracing takes the same samples as tracing one path in classical numerical integration. We evaluate the cost of classical path tracing by the number of ray paths  $N_c$ , as the noise comes mostly from the Monte Carlo integration. And in quantum ray tracing, the time cost is evaluated by the number of queries  $N_q$  to the ray tracing oracle  $O_f$ . The quantum integration has a convergence rate of  $O(1/N_q)$ , hence has a quadratic speedup over classical Monte Carlo integration with convergence rate of  $O(1/\sqrt{N_c})$ .

## 4. Results

### 4.1. Task

**2D Image Regression.** We train a QNNs to regress from 2D input pixel coordinates to the corresponding RGB values of an image (as shown in Figure 5). We consider two different distributions  $\mathcal{H}$ : face images (CelebA [34]) and natural images (Div2K [35]). Given a sampled image  $h \sim \mathcal{H}$ , we resize all images to  $256 \times 256$  as observations for network weights  $\theta$  in the optimization inner loop. At each inner loop step, the entire image is reconstructed and used to compute the loss. We then compare the classical MLPs and QNNs over these two distributions.

**3D Scene Representation.** The goal of 3D scene

representation is to generate a novel view of the scene from a set of reference images. We validate our QRFs through an extensive series of ablation studies and comparisons to recent techniques for accelerating NeRF. We evaluate our method on three inward-facing datasets:

- (1) Synthetic-NeRF [13]: The Synthetic-NeRF dataset consists of 360-degree views of complex objects in 8 scenes, where each scene has a central object with 100 inward facing cameras distributed randomly on the upper hemisphere. The images are  $800 \times 800$  with provided ground truth camera poses.
- (2) Synthetic-NSVF [15]: The Synthetic-NSVF contains 8 objects synthesized by NSVF. Strictly following the settings of NSVF, we set the image resolution to  $800 \times 800$  pixels and let each scene have 100 views for training and 200 views for testing.
- (3) Tanks & Temples [36]: The Tanks and Temples is a real-world dataset containing 5 scenes of real objects captured by an inward-facing camera surrounding the scene. Each scene contains between 152-384 images of size  $1920 \times 1080$ .

### 4.2. Implementation

The principal baseline for our experiments is NeRF [13]. We report the results of the original NeRF implementation, as well as the reimplementation in Jax (JaxNeRF) [14]. We also compare two older methods, Scene Representation Network (SRN) [7] and Neural Volume [8], as well as five recent papers introducing NeRF accelerations, Neural Sparse Voxel Field (NSVF) [15], AutoInt [17], FastNeRF [19], KioNeRF [20] and Depth-supervised NeRF (DS-NeRF) [21]. To evaluate QRF, we focus on two competing requirements of scene representations: quality of the generated images, and efficiency of the image generation. To quantify the rendering quality, we rely on three metrics: (1) Peak Signal to Noise Ratio (PSNR): A classic metric to measure the corruption of a signal. (2) Structural Similarity Index Measure (SSIM) [37]: A perceptual image quality assessment based on the degradation of structural information. (3) Learned Perceptual Image Patch Similarity (LPIPS) [38]: A perceptual metric based on the deep features of a trained network that is more consistent with human judgement.

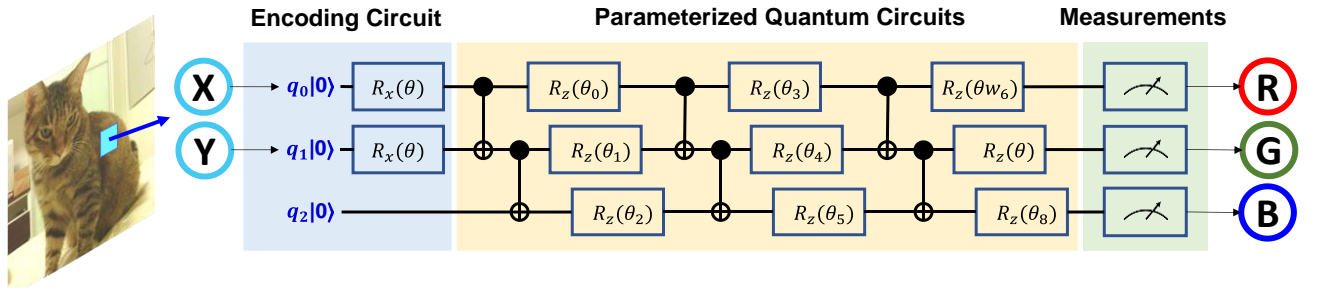


Figure 5: Quantum Implicit Neural Representations on 2D image regression with Encoding Circuit and PQS ( $n=3$ ,  $m=3$ )

where higher PSNR, SSIM and lower LPIPS is most desirable. We train both the NeRF baseline and QRF for a high number of iterations to find the limits of the representation capabilities of the respective architectures. We train each model for 350k iterations. For the classical scene representation, we train it in 48 hours using 2 Tesla V100 GPUs. For our 2D quantum representations and QRF model, we train it with Borealis, a photonic quantum processor from Xanadu that can be programmed and entangled. The inference time performance is measured on a Tesla V100 for classical NeRF and a Borealis for QRF.

### 4.3. Ablation Study

We use Div2K dataset [35] and the Drums scene of Synthetic-NeRF [13] to conduct the ablation experiments. The strategy of Activation Function, Encoding Circuit, and Quantum Circuit are discussed in this section.

**Activation Function.** We first compare the performance of the classical activation function and the quantum activation function in the 2D image regression and 3D scene representation task. We use Rectified Linear Unit (ReLU), Exponential Linear Unit (ELU), Smooth ReLU (SoftPlus), SIREN [9] as the classical activation function, and QReLU as quantum activation function. For 2D image regression, we train an MLP with 4 layers/256 channels and apply sigmoid activation to the output for each task. For 3D scene representation, we train baseline NeRF with 6 layers/256 channels and apply positional encoding to the input coordinates. Table 1 shows that when the QReLU function serves as the activation function for each task, it outperforms the state-of-the-art activation function, which indicates that the quantum activation function has better convergence.

**Encoding Circuit and Quantum Circuit.** We validate our QRF on our quantum system using various encoding circuits and quantum circuits. We use circuit-5/6/16/17 provided by Sukin Sim et al. [39], which has better expressive ability and entanglement ability, as the quantum circuit baseline. In addition, we adopt general qubit encoding, wavefunction encoding, angle encoding, and dense angle encoding as our encoding circuit strategy. More details about our quantum circuit and encoding circuit can be found in the supplemental.

The ablation results yield many significant findings (as show in Table 2). First, circuit-5 and circuit-6 are the fully connected graph arrangement of qubits which led to both favorable expressibility and entangling capability. Therefore, the network model with circuit-5 and circuit-6 has a high PSNR regardless of the encoding circuit used. Second, circuit-5 and circuit-16 with controlled Z-rotation (CRz) gates outperform circuit-6 and circuit-17, respectively. This is because the CRz operations in the entangling block commute with each other and thus the effective unitary operation comprised of CRz gates can be

expressed using unique generator terms that are fewer than the number of parameters for these gates.

Table 1: Ablation Study of activation function from the Div2K and Synthetic-NeRF dataset.

Task		2D Image Regression			3D Scene Representation		
Datasets		Div2K Dataset [35]			Synthetic-NeRF [13]		
Evaluate Metrics		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Classical AF	ReLU	32.89	0.961	0.044	24.85	0.812	0.208
	ELU	23.12	0.380	0.258	-	-	-
	Softplus	19.37	0.273	0.482	-	-	-
	SIREN	36.92	0.970	0.021	26.44	0.907	0.179
Quantum AF	QReLU	38.43	0.971	0.016	27.12	0.919	0.168

Table 2: Ablation Study of encoding circuit and quantum circuit from the Synthetic-NeRF dataset.

Evaluate Metrics	Encoding Circuit	Quantum Circuit			
		5	6	16	17
PSNR	General Qubit Encoding	26.52	26.33	25.52	24.82
	Wavefunction Encoding	26.58	26.39	25.58	24.88
	Angle Encoding	26.98	26.80	25.99	25.29
	Dense Angle Encoding	27.33	27.15	26.33	25.64
SSIM	General Qubit Encoding	0.893	0.875	0.833	0.809
	Wavefunction Encoding	0.880	0.885	0.838	0.820
	Angle Encoding	0.911	0.900	0.860	0.832
	Dense Angle Encoding	0.921	0.915	0.873	0.838
LPIPS	General Qubit Encoding	0.183	0.183	0.195	0.211
	Wavefunction Encoding	0.182	0.189	0.195	0.208
	Angle Encoding	0.175	0.176	0.188	0.206
	Dense Angle Encoding	0.169	0.176	0.183	0.199

### 4.4. Experiment Results

**2D Image Regression.** We first compare the performance between classical MLPs and QNNs model in the task of 2D image regression. According to the ablation study result, we use QReLU as the activation function, dense angle encoding as the encoding circuit, and circuit-5 as our parameterized quantum circuit. Table 3 shows that our proposed quantum model has significant advantages in each evaluate metric.

Table 3: Quantitative comparisons for 2D image regression. Compared with classical MLPs, our proposed QNNs model outperforms in the rendering quality.

Datasets	CelebA [33]			Div2K Dataset [34]		
Evaluate Metrics	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Classical MLPs	30.67	0.945	0.112	32.89	0.961	0.044
QNNs	33.71	0.963	0.038	36.36	0.969	0.027

**3D Scene Representation.** We also adopt same strategy (QReLU, dense angle encoding, circuit-5) as QNNs to

construct the QRF architecture. As shown in Table 4, we find that QRF inference is over 2000 times faster than NeRF, and faster than other methods except FastNeRF. It is because that FastNeRF can compactly cached and subsequently queried to compute the pixel values in the rendered image by factorization approach. Furthermore, QRF performed the best among all image quality metrics. Notably, our method does not address the training speed issue. We propose to improve the training speed of NeRF models by finding initialization through Meta Learning [16], or reconstructing the per-scene radiance field by Direct Voxel Grid Optimization NeRF [22].

## 5. Conclusion

In this paper, we presented Quantum Radiance Fields, a novel extension to NeRF that enables the rendering of photorealistic images using quantum computing. As a result, our method not only renders much faster, but can also deliver higher quality images. Moreover, the presented acceleration quantum strategy might also apply more broadly to other methods. We hope this paper can be a demonstration that quantum computing has the potential to provide satisfactory solutions for scene representation and volume rendering.

Table 4: Quantitative results on each scene from the Synthetic-NeRF [13], Synthetic-NSVF [15], and Tanks and Temples [36]. We highlight the top 3 results in each column are color coded as **Top 1**, **Top 2** and **Top 3**.

Dataset Evaluate Metrics	Synthetic-NeRF [13]				Synthetic-NSVF [15]				Tanks and Temples [36]			
	PSNR	SSIM	LPIPS	FPS	PSNR	SSIM	LPIPS	FPS	PSNR	SSIM	LPIPS	FPS
SRN [7]	22.26	0.846	0.170	0.909	24.33	0.882	0.141	1.304	24.10	0.847	0.251	0.250
Neural Volumes [8]	26.05	0.893	0.160	3.330	25.83	0.892	0.124	4.778	23.70	0.834	0.260	1.000
NeRF [13]	31.01	0.947	0.081	0.023	30.81	0.952	0.043	0.033	25.78	0.864	0.198	0.007
JaxNeRF [14]	31.69	0.953	0.049	0.045	31.49	0.958	0.026	0.065	27.94	0.904	0.168	0.013
NSVF [15]	<b>31.75</b>	<b>0.953</b>	<b>0.047</b>	0.815	<b>35.18</b>	<b>0.979</b>	<b>0.015</b>	0.095	<b>28.42</b>	<b>0.907</b>	<b>0.153</b>	0.163
AutoInt [17]	25.55	0.911	0.170	0.380	26.63	0.916	0.090	0.545	22.28	0.766	0.278	0.116
DoNeRF [18]	<b>32.50</b>	<b>0.957</b>	<b>0.037</b>	5.635	32.29	0.962	0.027	8.085	27.02	0.805	0.174	1.715
FastNeRF [19]	29.97	0.941	0.053	<b>172.42</b>	29.78	0.946	0.083	<b>224.71</b>	24.92	0.792	0.213	<b>47.67</b>
KioNeRF [20]	31.02	0.950	0.051	<b>38.46</b>	<b>33.37</b>	<b>0.970</b>	<b>0.020</b>	<b>55.07</b>	<b>28.41</b>	<b>0.910</b>	<b>0.091</b>	<b>11.68</b>
QRF	<b>32.65</b>	<b>0.960</b>	<b>0.029</b>	<b>47.26</b>	<b>35.44</b>	<b>0.980</b>	<b>0.014</b>	<b>67.70</b>	<b>29.65</b>	<b>0.820</b>	<b>0.085</b>	<b>14.36</b>

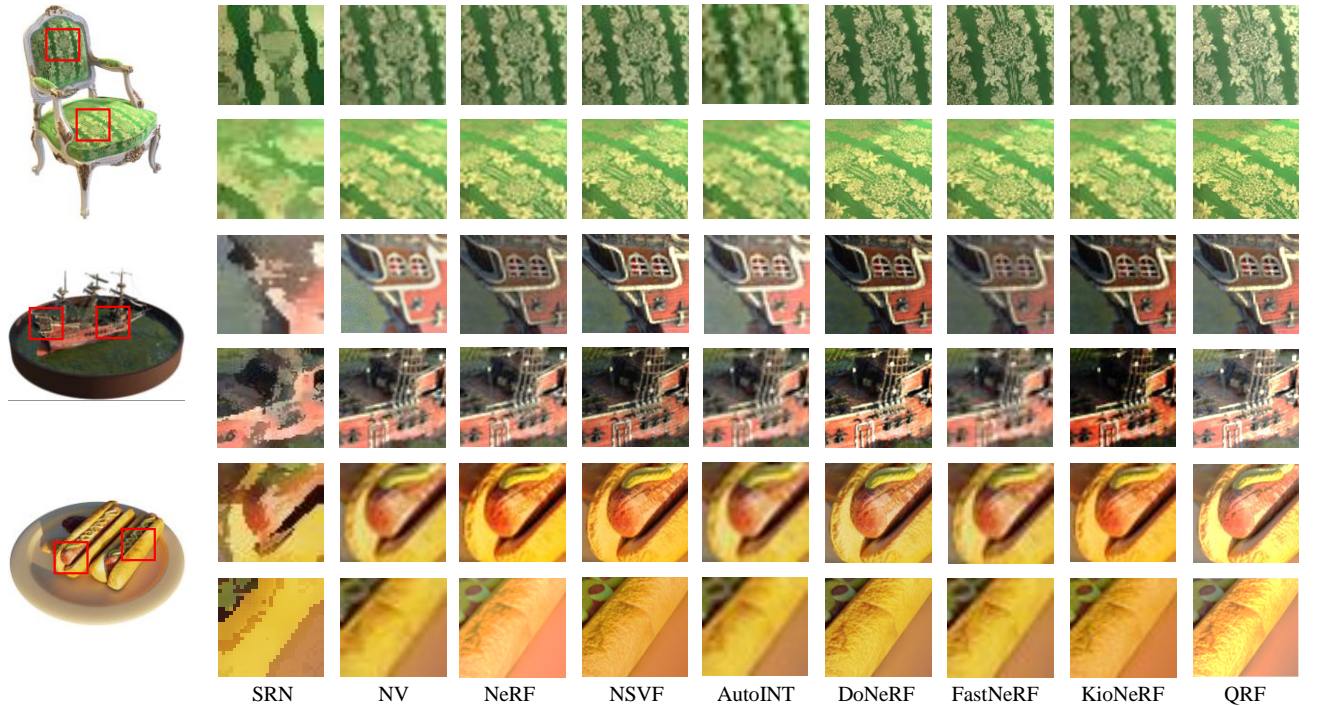


Figure 6: Qualitative comparisons on Synthetic-NeRF [13]. We compare classical implicit representation, NeRF, NeRF accelerations, and our proposed method. On this dataset, we find that our method better recovers the fine details in the scene. The results are similar in other datasets, please refer to our supplementary for more details.



## References

- [1] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3D ShapeNets: A deep representation for volumetric shapes", *Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [2] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T. Freeman, and Joshua B. Tenenbaum, "Learning shape priors for single-view 3D completion and reconstruction", *European Conference on Computer Vision (ECCV)*, 2018.
- [3] Haoqiang Fan, Hao Su, and Leonidas Guibas, "A point set generation network for 3D object reconstruction from a single image", *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [4] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation", *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [5] Kan Guo, Dongqing Zou and Xiaowu Chen, "3D mesh labeling via deep convolutional neural networks", *ACM Trans. on Graphics (SIGGRAPH)*, 2015.
- [6] Pengyu Wang, Yuan Gan, Panpan Shui, Fenggen Yu, Yan Zhang, Songle Chen, and Zhengxing Sun, "3D shape segmentation via shape fully convolutional networks", *Computers & Graphics*, 2017.
- [7] Vincent Sitzmann, Michael Zollhöfer and Gordon Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations", *Neural Information Processing Systems (NeurIPS)*, 2019.
- [8] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann and Yaser Sheikh, "Neural volumes: Learning dynamic renderable volumes from images", *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 65-65, 2019.
- [9] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell and Gordon Wetzstein, "Implicit neural representations with periodic activation functions", *Neural Information Processing Systems (NeurIPS)*, 2020.
- [10] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa and Hao Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization", *International Conference on Computer Vision (ICCV)*, 2019.
- [11] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss and Andreas Geiger, "Texture fields: Learning texture representations in function space", *International Conference on Computer Vision (ICCV)*, 2019.
- [12] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, et al., "Multiview neural surface reconstruction by disentangling geometry and appearance", *Neural Information Processing Systems (NeurIPS)*, 2020.
- [13] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi and Ren Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis", *European Conference on Computer Vision (ECCV)*, 2020.
- [14] Boyang Deng, Jonathan T. Barron, and Pratul P. Srinivasan, "JaxNeRF: an efficient JAX implementation of NeRF", 2020.
- [15] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua and Christian Theobalt, "Neural sparse voxel fields", *Neural Information Processing Systems (NeurIPS)*, 2020.
- [16] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng, "Learned Initializations for Optimizing Coordinate-Based Neural Representations", *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [17] David B. Lindell, Julien N. P. Martel, and Gordon Wetzstein, "AutoInt: Automatic Integration for Fast Neural Volume Rendering", *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [18] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravyarthy R. Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger, "DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks", *Eurographics Symposium on Rendering Techniques (EGSR)*, 2021.
- [19] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin, "FastNeRF: High-Fidelity Neural Rendering at 200FPS", *International Conference on Computer Vision (ICCV)*, 2021.
- [20] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger, "KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs", *International Conference on Computer Vision (ICCV)*, 2021.
- [21] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan, "Depth-supervised NeRF: Fewer Views and Faster Training for Free", *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [22] Cheng Sun, Min Sun, and Hwann-Tzong Chen, "Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction", *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [23] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven, "Characterizing Quantum Supremacy in Near-Term Devices", *Nature Physics*, vol. 14, no. 6, pp. 595-600, 2018.
- [24] Carlo Ciliberto, Mark Herbster, Alessandro Davide Ialongo, Massimiliano Pontil, Andrea Rocchetto, Simone Severini, and Leonard Wossnig, "Quantum machine learning: a classical perspective", *The Royal Society*, 2017.
- [25] Istvan Barabasi, Charles C. Tappert, Daniel Evans, and Avery M. Leider, "Quantum Computing and Deep Learning Working Together to Solve Optimization Problems", *International Conference on Computational Science and Computational Intelligence (CSCI)*, 2019.
- [26] Maria Schuld, "Supervised quantum machine learning models are kernel methods", *arXiv:2101.11020*, 2021.
- [27] Iordanis Kerenidis, and Alessandro Luongo, "Quantum classification of the MNIST dataset via Slow Feature Analysis", *arXiv:1805.08837*, 2018.
- [28] Danial Dervovic, Mark Herbster, Peter Mountney, Simone Severini, Nairi Usher, and Leonard Wossnig, "Quantum linear systems algorithms: a primer", *arXiv:1802.08227*, 2018.
- [29] Zhikuan Zhao, Jack K. Fitzsimons, Patrick Rebentrost, Vedran Dunjko, and Joseph F. Fitzsimons, "Smooth input preparation for quantum and quantum-inspired machine learning", *arXiv:1804.00281*, 2018.

- [30] E. Miles Stoudenmire, and David J. Schwab, "Supervised Learning with Tensor Networks", *Neural Information Processing Systems (NeurIPS)*, 2016.
- [31] Edward Grant, Marcello Benedetti, Shuxiang Cao, Andrew Hallam, Joshua Lockhart, Vid Stojevic, Andrew G. Green, and Simone Severini, "Hierarchical quantum classifiers", *npj Quantum Information*, vol. 4, pp. 1–8, 2018.
- [32] Shuxiang Cao, Leonard Wossnig, Brian Vlastakis, Peter Leek, and Edward Grant, "Cost function embedding and dataset encoding for machine learning with parameterized quantum circuits", *arXiv:1910.03902*, 2019.
- [33] Lov K. Grover, "Quantum Mechanics Helps in Searching for a Needle in a Haystack", *Physical Review Letters*, 1997.
- [34] Ziwei Liu, Ping Luo, Xiaogang Wang and Xiaoou Tang, "Deep learning face attributes in the wild", *International Conference on Computer Vision (ICCV)*, 2015.
- [35] Andrey Ignatov, Radu Timofte, et al., "PIRM challenge on perceptual image enhancement on smartphones", *European Conference on Computer Vision (ECCV) Workshops*, 2018.
- [36] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction", *ACM Trans. on Graphics*, 2017.
- [37] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh and Eero P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", *IEEE Transactions on Image Processing*, 2004.
- [38] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman and Oliver Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric", *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [39] Sukin Sim, Peter D. Johnson, and Alan Aspuru-Guzik, "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms", *arXiv:1905.10876*, 2019.