Q-NeRF: Neural Radiance Fields on a Simulated Gate-based Quantum Computer

Anonymous Author(s)
Affiliation
Address
email

Abstract

Quantum Neural Networks have shown theoretical and empirical promise in improving model compactness and convergence speed over classical deep networks. Meanwhile, novel view synthesis has seen major advances with Neural Radiance Fields (NeRFs), albeit at the cost of large models and intensive training. In this work, we introduce QNeRF, the first hybrid quantum-classical model for 3D scene representation and novel-view synthesis. QNeRF leverages parameterized quantum circuits to encode spatial and view-dependent information via quantum superposition and entanglement. We present two variants. The first model fully exploits quantum amplitudes to increase representational capabilities. The second, based on a novel dual-branch quantum embedding, introduces a task-informed inductive bias that drastically reduces the complexity of quantum state preparation, ensuring scalability and potential hardware compatibility. In our experiments, we show that, when trained on images of low resolutions, QNeRF matches or outperforms classical NeRF baselines while using less than half the number of parameters and converging significantly faster. These results suggest that Quantum Machine Learning can serve as a competitive alternative for continuous signal representation in vision and graphics.

1 Introduction

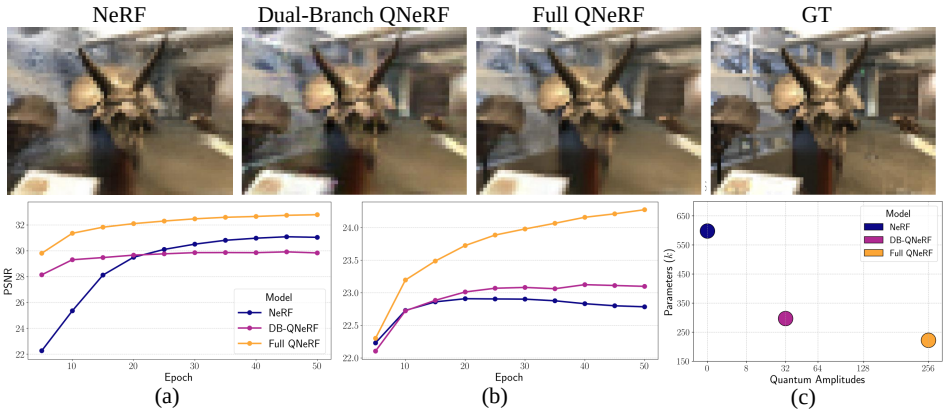

Figure 1: Top: Visualization of renderings for the scene *Trex*. Bottom: Average PSRN on testing during training for Blender dataset (a) and LLFF (b). On (c), we compare number of parameters (on y axis) with the number of amplitudes required in the quantum embedding (on x), a number related to the complexity of the quantum state preparation.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Neural Radiance Fields (NeRFs) have revolutionised novel view synthesis by modelling 3D scenes as a continuous volumetric function implicitly parametrized—in their simplest form—by a multi-layer perceptron (MLP) [1, 2, 3]. By encoding scene geometry and appearance through radiance and density fields, NeRFs achieve photorealistic rendering of unseen views in complex environments. These methods have found applications in areas such as 3D scenes reconstruction [4, 5, 6], human modelling [7, 8, 9], and image processing [10, 11]. Despite their impressive generalisation across novel viewpoints, the performance of NeRF-based frameworks remains heavily dependent on the underlying learning model, particularly in terms of the training efficiency and convergence speed.

While increasing the capacity of classical neural networks remains a standard strategy for improving performance, a more principled alternative lies in rethinking the foundations of learning models. This work investigates Quantum Neural Networks (QNNs) as a novel, previously unexplored and highly efficient framework in the context of novel view synthesis.

Quantum Machine Learning (QML) has recently emerged as an alternative paradigm to classical Machine Learning with theoretical promises and hopes to enhance characteristics of the ML models [12, 13]. Following early theoretical results demonstrating advantages in expressivity and learnability [14, 15, 16], the field is increasingly shifting towards practical implementations. In particular, QNNs, the quantum analogues of neural networks, have shown evidence of quantum advantage with respect to both model compactness and training efficiency in specific learning settings [17, 18, 19].

Building on these developments, we introduce a **Quantum Neural Radiance Field** (**QNeRF**), a hybrid model designed to harness the expressive power of parameterized quantum circuits (PQCs) for 3D scene representation (Figure 1). Inspired by QVFs [20], our model exploits quantum superposition and entanglement to learn complex representations more efficiently than classical neural networks. Meanwhile, output rendering and parameter optimisation are carried out classically, making the overall framework a *hybrid* quantum-classical model. We propose two architectural variants: (1) A **Full QNeRF**, which maximally exploits quantum resources to enhance representational capacity, and (2) A **Dual-Branch QNeRF**, which encodes spatial and view-dependent coordinates as separate intermediate quantum states, leading to improved scalability and potential better compatibility with near-term quantum hardware.

We experimentally demonstrate on simulated hardware that for low-resolution scenes the proposed QNeRF model achieves superior or comparable performance to classical (vanilla) NeRF architectures while requiring less than half the number of parameters and exhibiting significantly faster convergence. These improvements suggest that quantum neural architectures can serve as competitive alternatives for continuous signal representation in volumetric novel-view synthesis. In summary, our work has the following technical contributions:

- QNeRF, a hybrid quantum-classical model for radiance field learning that matches or exceeds the performance of the classical NeRF baseline, while using fewer than half the parameters and achieving significantly faster convergence. This is the first NeRF-based model designed with architectural compatibility for execution on quantum hardware.
- We introduce a dual-branch embedding that encodes spatial and view-dependent features as separate quantum states, imposing a task-informed inductive bias while simplifying quantum state preparation. Furthermore, we incorporate an *output scaling* strategy that mitigates exponential concentration effects [21], a key challenge in QML.

To encourage further research on this topic, we release the full implementation of our model along with all experimental code[1].

2 Related Work

2.1 Classical view synthesis

Neural networks have long been applied to image-based view synthesis [22, 23, 24]. However, a major breakthrough in the field was the introduction of Neural Radiance Fields (NeRF) [1], which proposed learning a continuous volumetric scene representation from posed images. NeRF defines

[1]The source code will be released after the first round of review. We submit a demo in the supplementary material.

a 5D mapping from spatial location and viewing direction to colour and density, enabling high-fidelity novel view synthesis. The success of NeRF set a new standard for rendering quality and inspired a substantial body of subsequent research [25, 3, 26, 27, 28]. Numerous extensions have aimed to improve rendering performance, generalisation, and robustness under specific conditions, such as Mip-NeRF [2], which improved performance at multiple resolutions, and instant-NGP [29], developed to dramatically reduce training time. It is worth noting that many of these NeRF variants retain the core architectural element of Multi-Layer Perceptrons to learn the radiance field, and could therefore benefit from a hybrid classical-quantum architecture such as the one we propose in this work. However, since our proposed quantum architecture enhances the "building block" of NeRF, these variants are only partially related to this work.

2.2 Quantum-enhanced Computer Vision

Quantum Computer Vision (QCV) has recently emerged as a subfield of quantum machine learning, focusing on leveraging quantum computing for tasks in image processing, graphics, and 3D reconstruction. Early work in this domain predominantly relied on quantum annealers [30, 31, 32, 33, 34, 35]. More recently, attention has shifted toward gate-based quantum computing due to its greater generality and potential for quantum advantage. Gate-based approaches offer increased flexibility in designing parameterized quantum circuits, enabling variational models that are trainable via classical optimisation. Several QCV applications have been proposed using gate-based architectures, including quantum convolutional neural networks [36, 37, 38], quantum generative adversarial networks [39, 40], quantum autoencoders [41], and other hybrid architectures [42, 18, 19].

Of particular relevance to this work are two prior contributions. The first is [43], which applies a hybrid architecture to visual data using a "sandwich" structure, where quantum layers are embedded between classical feature extractors and regressors. While effective, this configuration makes it difficult to disentangle the contribution of the quantum component from the overall model performance. In contrast, our architecture ensures that the quantum circuit processes the full encoded input and that the quantum embedding plays a direct role in model expressivity, allowing a cleaner analysis of the quantum contribution.

The second is the Quantum Visual Field (QVF) model [20] , which this work builds upon and is inspired by. In this work, the authors proposed a hybrid architecture for visual data using amplitude embedding and variational quantum circuits. While QVFs demonstrated the viability of using quantum embeddings for visual field representation, our work addresses a fundamentally different task and introduces several key innovations. First, our proposed **Full QNeRF** model is directly designed for the task of novel view synthesis. Moreover, components such as the quantum encoding and output scaling are designed respectively to reduce the complexity of the amplitude embedding, and to mitigate the *exponential concentration* [21] phenomenon. Finally, we propose a **Dual-Branch QNeRF** that independently processes positional and view-dependent features. This design not only reduces exponentially the gate complexity depending on quantum state preparation [44], but also introduces a task-aligned inductive bias that reflects the underlying factorization in view synthesis, where positional and directional information contribute differently to the generative process.

3 Quantum Machine Learning

In this section, we provide the necessary background on Quantum Machine Learning (QML). We defer notation details to Appendix B. Comprehensive introductions to Quantum Computing and QML can be found in [45] and [46], respectively.

Quantum Machine Learning arises from the intersection of Quantum Computing with Machine Learning and Deep Learning. The field encompasses a broad variety of models, methodologies, and objectives [47, 48]. One of the central goals of QML is to design models that exploit quantum phenomena such as entanglement and superposition to achieve an "advantage" in terms of computational resources, representational power, or learning performance when compared to purely classical methods [13]. A prominent class of models within QML is that of Quantum Neural Networks (QNNs). In [14], it was proved that QNNs can achieve higher effective dimensions than comparable classical neural networks, which translates into faster convergence during training. Following this theoretical result, experimental results from both general [17] and vision-specific tasks [18, 19] show that QNNs can achieve the same level or better performances than classical Neural Networks with

fewer parameters and with increased convergence speed. Given these properties, it is natural to investigate the application of QNNs to high-dimensional vision tasks, where efficient training and generalization are particularly critical due to the high dimensionality and complexity of the models required.

3.1 Quantum Neural Networks

In Quantum Machine Learning, the term Quantum Neural Network is often used interchangeably with Parameterized Quantum Circuit (PQC), as the two notions are closely related [49]. A QNN typically consists of a sequence of quantum gates whose operations depend on the classical inputs $\mathbf{x}$ and on a set of free parameters θ, which are optimized during the training process.

Given an initial state $|\phi\rangle$ (e.g. the state $|0\rangle^{\otimes N}$, where N is the size of the quantum system), the application of a parameterized quantum circuit P results in the state $P(\theta;\mathbf{x})|\phi\rangle$. In many architectures, the circuit P can be naturally decomposed into two stages: a data encoding stage and a variational (trainable) stage. The encoding is typically achieved through a fixed set of gates that map the classical input into a quantum state (often referred to as the *embedding* or *feature map*), denoted by $S(\mathbf{x})$. The variational stage, represented by $V(\theta)$, then acts on the embedded state. Thus, the final state of the circuit can be described as $V(\theta)S(\mathbf{x})|\phi\rangle = V(\theta)|\phi_S(\mathbf{x})\rangle$, where $|\phi_S(\mathbf{x})\rangle$ denotes the quantum state obtained by embedding the input $\mathbf{x}$. The structure $V(\theta)$ typically constitutes a variational ansatz, designed to be expressive enough to represent the target function during learning.

3.2 Noisy Intermediate-Scale Quantum

Current quantum hardware operates in what is referred to as the *Noisy Intermediate-Scale Quantum* (NISQ) era [50]. In this regime, quantum devices are composed of tens to hundreds of qubits, but they are subject to non-negligible levels of noise and decoherence, which significantly limit their computational capabilities. As a consequence, it is of fundamental importance to design quantum circuits that minimize both the number of quantum gates and the overall circuit depth. Excessive gate count or depth can result in an accumulation of errors that rapidly degrade the fidelity of the computation, making the outcomes unreliable. This limitation motivates the development of shallow, hardware-efficient quantum circuits and encourages a careful trade-off between expressivity and robustness to noise.

4 Quantum Neural Radiance Fields (Q-NeRFs)

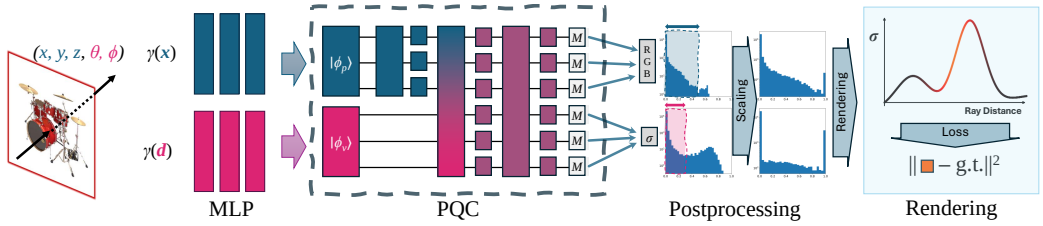

Figure 2: Scheme of the proposed Dual-Branch QNeRF. Positional and view-dependent coordinates (shown in teal and magenta, respectively) are first encoded into quantum amplitudes through two separate classical MLPs, processed by a PQC (dashed box), and converted again to classical information through a parity-based measurement. The output is then processed with a scaling layer, and employed to reconstruct the view, which is finally used to compute the loss with regard to the ground-truth image.

We describe our proposed method **Quantum Neural Radiance Field** (QNeRF), consisting of a hybrid quantum-classical model designed to efficiently represent scenes and generate novel views. An overview is provided in Figure 2. A brief background on Neural Radiance Fields is provided in Appendix A.

We start by describing how to encode classical coordinates into a quantum state. This description is given in two variants: one for the *Full* encoding, which allows to leverage all the properties of the quantum system, and one for the *Dual-Branch* encoding, which reduces the expressivity of the model to drastically increase scalability and compatibility with current quantum devices. Then, we present

the quantum circuit design choices. Finally, we describe postprocessing operations, consisting of a parity measurement to extract information, and then a "de-concentration" scaling layer that enhances the performance of the model by mitigating exponential concentration [21].

4.1 MLP-based Quantum Embedding

Neural Radiance Field [1] models heavily rely on positional encodings for both spatial and view-dependent information. This technique enriches each input coordinate $x \in \mathbb{R}$ by mapping it to a higher-dimensional space $\mathbb{R}^{2L}$ using the transformation

$$\gamma(x) = \left(\sin(2^0\pi x), \cos(2^0\pi x), \ldots, \sin(2^{L-1}\pi x), \cos(2^{L-1}\pi x)\right). \tag{1}$$

This encoding has been shown to be crucial for mitigating the spectral bias of neural networks, which tend to underrepresent high-frequency components, thereby enabling the representation of fine-grained variations in the data. Even for modest values of L, the dimension of $\gamma(x)$ exceeds the number of qubits available in current quantum devices. Therefore, to obtain a quantum state that represents the structure from the encoded vector, we employ amplitude embedding. Unlike angle embedding or unitary amplitude encoding [51], which scale linearly with the number of features, amplitude embedding offers exponential compression, as a vector with 2^n components is represented using only n qubits.

Full Embedding with All Amplitudes We first describe the *Full* embedding strategy in which all the possible amplitudes are used to encode data, obtaining a more expressive representation. As a first step, the enriched input vector $\gamma(\mathbf{x})$, obtained via positional encoding, is mapped to a normalised vector $\mathcal{M}(\mathbf{x})$ of 2^n *amplitudes*. This requires addressing three key challenges: (i) decoupling the dimensionality of $\mathbf{x}$ from the hardware-constrained number of amplitudes 2^n; (ii) ensuring that $\mathcal{M}(\mathbf{x})$ retains sufficient representational structure for the downstream quantum model; and (iii) satisfying the normalization constraint $\sum_{i\leq 2^n} |\alpha_i|^2 = 1$, where α_i denotes the i-th amplitude. To this end, we employ a lightweight Multi-Layer Perceptron (MLP) with an output layer of size 2^n. A ReLU activation function is applied at the end of the MLP to ensure that all amplitudes are non-negative and increasing sparsity, thereby facilitating quantum state preparation [52]. Finally, the amplitudes are normalised to guarantee compatibility with quantum embedding requirements.

Dual-Branch Embedding Although recent work has improved the efficiency of amplitude state preparation [53] encoding 2^n amplitudes on a quantum device remains a significant challenge [54, 55, 44]. Since the state preparation requires up to $\mathcal{O}(k)$ gates, where k is the number of amplitudes, exploiting the full quantum space can be unfeasible on current NISQ devices [50].

On the other hand, there is no intrinsic reason to treat positional and view-dependent features identically within the quantum embedding. Drawing inspiration from the original NeRF architecture, in which view-dependent features are encoded after positional features have been already processed, we introduce a **dual-branch** that separates and independently processes these components, while exponentially reducing the number of amplitudes required. Formally, we divide the total qubit budget into n_p and n_v qubits, assigned to positional and view-dependent encodings, respectively. Two MLPs are trained to produce 2^{n_p}- and 2^{n_v}-dimensional vectors, which are independently amplitude-encoded and then composed into a tensor product state:

$$|\phi(x)\rangle = |\phi_p(x_p)\rangle \otimes |\phi_v(x_v)\rangle. \tag{2}$$

In the simplest case where $n_p = n_v = n/2$, the total number of amplitudes is reduced to $2^{n/2+1}$, offering an exponential reduction relative to the Full QNeRF approach. Furthermore, the dual-branch strategy also yields an exponential reduction in the number of parameters required for the MLPs. In particular, our Dual-Branch QNeRF model can scale up to more than 15 qubits while maintaining fewer parameters than the classical NeRF model. An in-depth analysis of the number of amplitudes and parameters for both encoding strategies is provided in Appendix C.

4.2 Quantum Circuit Design

Following quantum embedding, the encoded data is processed by a variational quantum circuit with learnable parameters $V(\boldsymbol{\theta})$. In line with prior work on real-valued quantum feature spaces [20], we restrict the variational ansatz to the real subspace of the Hilbert space by employing only R_Y rotations.

This design choice simplifies gradient-based optimisation and increases resilience to hardware noise, without significantly limiting model expressivity for the task considered.

The circuit is composed of two elementary modules: (i) a single-qubit rotational layer consisting of $R_Y(\theta)$ gates applied independently to each qubit, and (ii) a *dense entangling layer* that applies controlled-$R_Y(\theta)$ gates across pairs of qubits. Formally, given disjoint control and target sets of qubits A, B, the dense layer applies a controlled-R_Y gate to each $(q_A, q_B) \in A \times B$. On the other hand, when $A = B$ we apply (q_i, q_j) for each pair with $i < j$. In the Full QNeRF, we build a standard circuit as a sequence of $\ell \geq 1$ blocks, each composed of a dense entangling layer followed by a rotational layer over all n qubits.

In what follows, we restrict our analysis to the circuit used in the Dual-Branch QNeRF. For a graphical representation, we refer to the central block in Figure 2. Starting from the state obtained by the Dual-Branch encoding in Equation 4.1, we process the qubit sets for positional and view-dependent features separately. Initially, a dense entangling layer is applied to the n_p positional qubits, followed by a rotational layer. After these steps, the state of the system can be written as

$$(V_p(\theta) \otimes I_v)\left(|\phi_p(x_p)\rangle \otimes |\phi_v(x_v)\rangle\right) = V_p(\theta)|\phi_p(x_p)\rangle \otimes |\phi_v(x_v)\rangle, \tag{3}$$

where V_p is the operator corresponding to the two positional layers. This design ensures that positional information is first internally processed and entangled, mirroring the sequential encoding strategy used in classical NeRFs. Next, the view-dependent amplitude embedding is introduced, and a second dense entangling layer is applied between the n_p positional qubits and the n_v view-dependent qubits, combining the two feature spaces. A global rotational layer is then applied to all n qubits. By entangling states corresponding to positional and view-dependent features, we ensure that the model can learn the correlation between all the input coordinates. Finally, to further enhance model expressivity, the circuit can optionally be extended with $\ell \geq 0$ additional blocks, each comprising a dense entangling layer followed by a rotational layer over all n qubits.

4.3 Parity-based Measurements

At the end of a quantum circuit, the quantum state is typically converted into classical data via measurement. In this work, we measure the system in the computational basis, corresponding to projective measurements in the eigenbasis of the Pauli-Z operator. To mitigate the effects of barren plateaus [56, 57], we employ local measurements. Specifically, we define a family of single-qubit observables acting on individual qubits as

$$\hat{O}_i = \mathbb{I}^{\otimes(i-1)} \otimes Z \otimes \mathbb{I}^{\otimes(n-i)}, \quad \text{for } i = 1, \ldots, n, \tag{4}$$

where Z denotes the Pauli-Z operator and $\mathbb{I}$ is the identity operator on a single qubit. In practice, we write $\hat{O}_i = Z_i$, indicating a Pauli-Z measurement on the i-th qubit. These local projective measurements yield bitstrings corresponding to independent measurements on each qubit. Local observables are both hardware-efficient and less susceptible to barren plateaus compared to global observables, and have been shown to enhance the trainability of variational quantum algorithms [58, 59]. The output of the circuit is given by the expectation value

$$\mathcal{O}(\mathbf{x}) = \langle \phi_{\text{in}} | V(\boldsymbol{\theta})^\dagger \hat{O} V(\boldsymbol{\theta}) | \phi_{\text{in}} \rangle, \tag{5}$$

where $\mathbf{x}$ is the input (derived from positional and view-dependent features), $V(\boldsymbol{\theta})$ is the parameterized quantum circuit, and the input state $|\phi_{\text{in}}\rangle = |\phi_p\rangle \otimes |\phi_v\rangle$ is prepared via a dual-branch amplitude embedding. The output $\mathcal{O}(\mathbf{x})$ is therefore a vector of n real numbers in the interval $[-1, 1]$.

To obtain outputs suitable for the considered task, we transform $\mathcal{O}(\mathbf{x})$ into a 4-dimensional vector representing RGB colour channels and a volumetric density. This transformation proceeds in two stages. First, for each of the four output components, we select a subset of qubits $\mathcal{C}_i \subset \{1, \ldots, n\}$ and compute the average of their corresponding expectation values, i.e.,

$$\tilde{o}_i(\mathbf{x}) = \frac{1}{|\mathcal{C}_i|} \sum_{j \in \mathcal{C}_i} \mathcal{O}_j(\mathbf{x}), \tag{6}$$

where $\mathcal{O}_j(\mathbf{x})$ is the expectation value associated with qubit j. This step, referred to as *parity averaging*, aggregates information from specific qubit subsets into semantically meaningful outputs. Finally, we clip the resulting values to $[0, 1]$ The resulting vector $\mathbf{o}(\mathbf{x}) = (o_1(\mathbf{x}), \ldots, o_4(\mathbf{x})) \in [0, 1]^4$ can then be used to reconstruct a novel view as described in [1]. The novel view is then used to compute the standard MSE loss by comparing it with the ground truth.

4.4 Output Scaling

Variational quantum circuits are known to suffer from the *exponential concentration* phenomenon [21], which refers to the tendency of their output distributions to concentrate exponentially around their mean as the number of qubits increases. This effect significantly limits the expressive power of such circuits, particularly in representing distributions with high variance. To mitigate the limitations introduced by exponential concentration and improve the trainability of our model, we introduce a learnable scaling factor α_c applied to the output associated with each channel. The final output $\alpha_c \mathcal{O}_c(\mathbf{x})$ is then clipped to $[0, 1]$. This modification widens the range of output values, thereby counteracting the tendency of the circuit outputs to collapse around their mean, and acts in practice as a "de-concentration" layer. Empirically, we found that output scaling plays a crucial role in enhancing model performance, in particular with regard to the predicted density. An example of the effects of output scaling is shown in Figure 3.

5 Experimental Evaluation

To assess the effectiveness of our proposed approach, we conduct experiments on noiseless, simulated quantum hardware across multiple scenes, following the evaluation protocol established in [1]. Given computational constraints, we select a representative subset of eight scenes: four from the Blender dataset, obtained from a synthetic setting, and four from the LLFF dataset [24]. All images used during training and testing are downscaled to reduce training time. This depends on the fact that quantum simulation is computationally expensive, and that currently there are no GPU-based simulation frameworks compatible with our proposed method. More details on the experimental setting, including simulation time are provided in Appendix D.

We evaluate both Full QNeRF and Dual-Branch QNeRF using quantum circuits with 8 qubits. It is important to emphasise that, as this represents the first QML-based approach for novel view synthesis, direct comparison with other baselines is inherently challenging. Classical architectures operate under a fundamentally different computational paradigm and typically require extensive training, which is not currently feasible for either real quantum hardware or classically simulated QML models. To establish a meaningful baseline, we compare our quantum models against the original classical NeRF architecture, trained on the same downscaled images, by matching the number of epochs. We provide a comparison with more recent models in Appendix E, obtaining coherent observations. However, we caution that such comparisons should be interpreted with care, as these models benefit from large-scale optimization and higher parameter counts. Integrating analogous techniques into quantum-based models remains an open direction for future work.

A summary of the results is provided in Tables 1a and 1b. Learning curves for each scene are provided in Appendix F, Figure 8. Qualitative results are provided in Appendix F, Figures 6 and 7.

5.1 Model Configuration

The encoding sizes for positional and view-dependent features are set to 10 and 4, respectively, following the original NeRF model. In both quantum models, the encoding MLPs consist of three fully connected layers with a hidden dimension of 256, and output equal to the number of amplitudes, corresponding to 2^8 for the Full QNeRF, and 2^5 for the Dual-Branch QNeRF. Both ansatz are constructed as described in Section 4.2 with $\ell = 1$. A detailed breakdown of the number of classical and quantum parameters is provided in Appendix C. Initial values for the quantum parameters are chosen using *identity initialisation* [60], a common initialisation strategy to mitigate barren plateaus. Finally, the parity measurement assigns 2 qubits for each output channel.

5.2 Blender Dataset

For the Blender dataset, we selected the scenes *materials*, *ficus*, *lego*, and *drums*. Each image was downscaled via average pooling to a resolution of 50×50 pixels. All models were trained on a subset of 100 training images and evaluated on a validation set of 200 samples. Complete quantitative results are reported in Table 1a. The average testing performance during training is visualised in Figure 1 (a).

After 10 training epochs, both quantum models consistently outperform the classical NeRF baseline, with the only exception being the *lego* scene, where DB-QNeRF underperforms but is still surpassed

by FQNeRF. On average, the PSNR gain after 10 epochs is approximately 4 points for DB-QNeRF and 6 points for FQNeRF, highlighting the fast convergence and expressive power of quantum-encoded features in the early training phase. Following full training, the classical NeRF eventually exceeds DB-QNeRF in performance, suggesting that the limited number of encoded amplitudes constrains its capacity to fully capture scene complexity. In contrast, FQNeRF consistently outperforms the classical baseline across all scenes, with particularly significant improvements on more detailed cases such as *materials* and *drums*.

Model	NISQ	*Materials*	*Ficus*	*Lego*	*Drums*	*Average*
Full	Low	**33.44 ± 0.25** **35.34 ± 0.12**	**30.31 ± 0.16** **31.29 ± 0.12**	**32.96 ± 0.19** **34.93 ± 0.12**	**28.71 ± 0.08** **29.60 ± 0.06**	**31.36 ± 0.17** **32.79 ± 0.11**
DB	High	30.32 ± 0.23 31.22 ± 0.32	29.44 ± 0.13 29.63 ± 0.24	30.53 ± 0.24 31.50 ± 0.27	26.94 ± 0.20 27.01 ± 0.16	29.31 ± 0.20 29.84 ± 0.25
NeRF	-	20.48 ± 6.81 32.47 ± 0.54	23.34 ± 6.27 31.02 ± 0.21	32.71 ± 0.49 33.59 ± 0.18	24.26 ± 0.18 26.85 ± 0.12	25.19 ± 3.44 30.98 ± 0.26

(a) Blender

Model	NISQ	*Horns*	*Fern*	*Trex*	*Room*	Average
Full	Low	**22.45 ± 0.54** **23.70 ± 0.64**	22.11 ± 0.22 **23.32 ± 0.49**	**22.59 ± 1.03** **23.01 ± 1.48**	**26.68 ± 0.64** **28.25 ± 0.55**	**23.46 ± 0.61** **24.57 ± 0.79**
DB	High	21.40 ± 0.43 22.19 ± 0.46	**22.16 ± 0.35** 22.02 ± 0.40	21.92 ± 0.50 21.99 ± 0.40	25.42 ± 0.52 26.20 ± 0.46	22.72 ± 0.45 23.10 ± 0.43
NeRF	-	20.80 ± 0.64 21.02 ± 0.60	21.95 ± 0.34 21.73 ± 0.41	22.10 ± 0.41 21.60 ± 0.29	26.07 ± 0.48 26.80 ± 0.50	22.73 ± 0.47 22.79 ± 0.45

(b) LLFF

Table 1: PSNR performance (mean ± standard deviation) of each model on Blender scenes at 10 (first row) and 50 (second row) training epochs. Quantum models are annotated with their compatibility with Noisy Intermediate-Scale Quantum (NISQ) hardware. For each row, the best and second-best results are highlighted in bold and underlined, respectively.

5.3 LLFF dataset

For the LLFF dataset, we selected the scenes *horns*, *fern*, *trex*, and *room*. Each image was downscaled up to a resolution of 126×95. For each run, the set of images was split in training and testing set in a 80-20 split. Complete quantitative results are reported in Table 1b. The average testing performance during training is visualised in Figure 1 (b).

FQNeRF consistently outperforms both DB-QNeRF and the classical NeRF model on all scenes, at both early (10 epochs) and later (50 epochs) stages of training. Notably, for the *horns*, *fern*, and *trex* scenes, FQNeRF achieves higher PSNR after just 10 epochs than the final PSNR reached by NeRF, demonstrating faster convergence and superior performance with fewer parameters. In contrast, DB-QNeRF shows performance on par with the classical model on average. However, unlike NeRF—which displays a slight PSNR drop after full training on *fern* and *trex*, possibly due to overfitting, DB-QNeRF maintains stable performance, ultimately achieving a marginally higher final PSNR and indicating better generalisation.

5.4 Effects of Output Scaling

We investigate the impact of output scaling on the scenes *horns* and *trex*. Specifically, we train both Full QNeRF and Dual-Branch QNeRF without the scaling (or de-concentration) layer, while keeping all other hyperparameters fixed. As shown in Table 2, removing output scaling leads to a noticeable performance drop in both models. In particular, the absence of scaling results in an average PSNR decrease of approximately 0.7 for the Full model and more than 1.7 for the Dual-Branch variant on *horns*, and of 0.3 and 0.65 on *trex*. This indicates that the Dual-Branch architecture is more reliant on

output scaling to achieve optimal performance. As shown in Figure 3, the effects of scaling for the Dual-Branch QNeRF are evident, as reconstruction without scaling leads to blurrier results.

5.5 Increasing the Number of Qubits

We evaluate the performance of the Dual-Branch QNeRF model as the number of qubits increases, on the scenes *horns* and *trex*. As shown in Figure 4, we observe an improvement in PSNR as the number of qubits grows. Since the total number of trainable parameters remains nearly constant across configurations (ranging from $293k$ to $301k$ for 6 to 10 qubits), the performance gain can be attributed to the increased number of encoded amplitudes. This suggests that leveraging a larger quantum state space contributes positively to reconstruction quality.

Model	*Horns*	*Trex*
Full	23.01 ± 0.69	22.73 ± 0.92
DB	20.44 ± 0.48	21.36 ± 0.49

Table 2: PSNR performance of quantum models without output scaling on the scenes *horns* and *trex*.

(a) No scaling (22.65) (b) Scaling (25.18)

Figure 3: Qualitative comparison (with PSNR) of DB model, with and without scaling.

6 Discussion and Limitations

The experimental results indicate that the Full QNeRF model consistently outperforms the classical NeRF baseline in both accuracy and convergence speed. On the other hand, DB-QNeRF allows for either increased convergence speed or enhanced performance.

While these results are encouraging, it is important to note that the experiments were conducted in a simulated environment. Real-world quantum hardware may introduce challenges such as noise and decoherence. Nonetheless, our design choices—such as shallow circuit depth, constrained expressibility, and use of the real-positive subspace—help mitigate these effects and enhance robustness. A potential challenge lies in the scalability of amplitude embedding for state preparation. Although our setting of DB-QNeRF is within the capabilities of current hardware and state preparation routines [61] (requiring encoding just 32 amplitudes), it remains to be studied how model performance and feasibility evolve as the number of qubits increases. We provide a preliminary analysis of this scalability aspect in Appendix C, showing that the number of amplitudes and classical parameters remain feasible up to 16 qubits. We believe that a deeper investigation into exact or approximate state preparation [52] on real hardware would be a valuable direction for future work.

7 Conclusion

This work introduces two novel hybrid classical-quantum models, Full QNeRF and Dual-Branch QNeRF, designed for the task of novel view synthesis. The Full QNeRF leverages the full capacity of quantum state encoding, achieving enhanced performance at the cost of higher resource requirements. In contrast, the Dual-Branch QNeRF adopts a more hardware-feasible architecture with favourable scalability properties, making it more suitable for near-term quantum hardware. Our experiments demonstrate that both models provide significant benefits over the classical NeRF baseline. Specifically, they enable either improved reconstruction quality or substantially faster convergence, often both, while using less than half the number of parameters compared to the classical model.

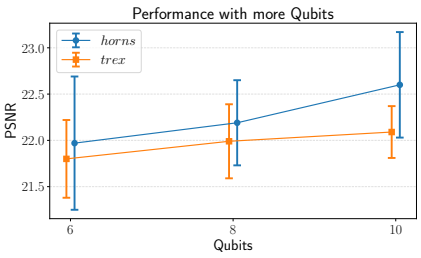

Figure 4: Evaluation (PSNR with standard deviation) of the proposed DB-QNeRF model for 6, 8, and 10 qubits.

In summary, we present the first quantum machine learning models tailored to a complex computer vision task, showing that quantum-enhanced representations can deliver competitive or superior performance with reduced parameter counts. Importantly, the DB-QNeRF model remains within the capabilities of current or near-term quantum hardware, marking a step forward toward practical quantum advantage in vision-based learning.

References

[1] 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. In *ECCV*, 2020.

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021.

[3] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. 10 2022.

[4] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF$--$: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[5] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *International Conference on Computer Vision*, 2021.

[6] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. *arXiv preprint arXiv:2103.15595*, 2021.

[7] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021.

[8] Sida Peng, Chen Geng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Implicit neural representations with structured latent codes for human body modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[9] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J. Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. *2020 International Conference on 3D Vision (3DV)*, pages 333–344, 2020.

[10] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021.

[11] Nianyi Li, Simron Thapa, Cameron Whyte, Albert W. Reed, Suren Jayasuriya, and Jinwei Ye. Unsupervised non-rigid image distortion removal via grid deformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2522–2532, October 2021.

[12] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, September 2017.

[13] M. Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J. Coles. Challenges and Opportunities in Quantum Machine Learning. *Science*, 2:567–576, 2022.

[14] Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computat. Sci.*, 1:403–409, 2021.

[15] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17(9):1013–1017, 2021.

[16] N. Pirnay, S. Jerbi, J. P. Seifert, and J. Eisert. An unconditional distribution learning advantage with shallow quantum circuits, 2024.

[17] Ricardo Simoes, Patrick Huber, Nicola Meier, Nikita Smailov, Rudolf Fuchslin, and Kurt Stockinger. Experimental evaluation of quantum machine learning algorithms. *IEEE Access*, PP:1–1, 01 2023.

[18] El Amine Cherrat, Iordanis Kerenidis, Natansh Mathur, Jonas Landman, Martin Strahm, and Yun Yvonna Li. Quantum Vision Transformers. *Quantum*, 8:1265, February 2024.

[19] Jonas Landman, Natansh Mathur, Yun Yvonna Li, Martin Strahm, Skander Kazdaghli, Anupam Prakash, and Iordanis Kerenidis. Quantum Methods for Neural Networks and Application to Medical Image Classification. *Quantum*, 6:881, December 2022.

[20] Anonymous et al. Quantum-inspired multi-dimensional visual fields with learnable energy representations, 2025.

[21] Supanut Thanasilp, Samson Wang, M. Cerezo, and Zoë Holmes. Exponential concentration in quantum kernel methods. *Nature Communications*, 15(1), 6 2024.

[22] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. DeepView: View Synthesis With Learned Gradient Descent . In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2362–2371, Los Alamitos, CA, USA, June 2019. IEEE Computer Society.

[23] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019.

[24] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[25] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022.

[26] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P. Srinivasan, Richard Szeliski, Jonathan T. Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. *arXiv*, 2023.

[27] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[28] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. *CVPR*, 2022.

[29] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[30] Marcel Seelbach Benkner, Zorah Lähner, Vladislav Golyanik, Christof Wunderlich, Christian Theobalt, and Michael Moeller. Q-match: Iterative shape matching via quantum annealing. In *International Conference on Computer Vision (ICCV)*, 2021.

[31] Shahrokh Heidari, Michael J. Dinneen, and Patrice Delmas. Quantum annealing for computer vision minimization problems. *Future Generation Computer Systems*, 160:54–64, 2024.

[32] Han Choong, Suryansh Kumar, and Luc Gool. Quantum annealing for single image super-resolution. 04 2023.

[33] Tolga Birdal, Vladislav Golyanik, Christian Theobalt, and Leonidas Guibas. Quantum permutation synchronization. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[34] Matteo Farina, Luca Magri, Willi Menapace, Elisa Ricci, Vladislav Golyanik, and Federica Arrigoni. Quantum multi-model fitting. In *Computer Vision and Pattern Recognition (CVPR)*, 2023.

[35] Jan-Nico Zaech, Alexander Liniger, Martin Danelljan, Dengxin Dai, and Luc Gool. Adiabatic quantum computing for multi object tracking. 02 2022.

[36] Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook. Quanvolutional neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence*, 2(1):2, 2020.

[37] Fan Fan, Yilei Shi, Tobias Guggemos, and Xiao Xiang Zhu. Hybrid quantum-classical convolutional neural network model for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12):18145–18159, 2024.

[38] Li Hai, Chen Liang, Hao Yaming, Yu Wenli, and Shi Fengquan. An improved convolutional neural networks: Quantum pseudo-transposed convolutional neural networks. *IEEE Access*, 13:37108–37117, 2025.

[39] He-Liang Huang, Ming Gong, Youwei Zhao, Yulin Wu, Chaoyue Wang, Shaowei Li, Futian Liang, Jin Lin, Yu Xu, Rui Yang, Tongliang Liu, Min-Hsiu Hsieh, Hui Deng, Hao Rong, Cheng-Zhi Peng, Chao-Yang Lu, Yu-Ao Chen, Dacheng Tao, and Jian-Wei Pan. Experimental quantum generative adversarial networks for image generation. *Physical Review Applied*, 16, 08 2021.

[40] Daniel Silver, Aditya Ranjan, Tirthak Patel, Harshitta Gandhi, William Cutler, and Devesh Tiwari. MosaiQ: Quantum Generative Adversarial Networks for Image Generation on NISQ Computers . In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7007–7016, Los Alamitos, CA, USA, October 2023. IEEE Computer Society.

[41] Lakshika Rathi, Edith Tretschk, Christian Theobalt, Rishabh Dabral, and Vladislav Golyanik. 3D-QAE: Fully quantum auto-encoding of 3d point clouds. In *The British Machine Vision Conference (BMVC)*, 2023.

[42] Hrant Gharibyan, Hovnatan Karapetyan, Tigran Sedrakyan, Pero Subasic, Vincent P. Su, Rudy H. Tanin, and Hayk Tepanyan. Quantum image classification: Experiments on utility-scale quantum computers, 2025.

[43] Jiaming Zhao, Wenbo Qiao, Peng Zhang, and Hui Gao. Quantum implicit neural representations. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[44] Xiao-Ming Zhang, Tongyang Li, and Xiao Yuan. Quantum state preparation with optimal circuit depth: Implementations and applications. *Phys. Rev. Lett.*, 129:230504, Nov 2022.

[45] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[46] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56, 09 2014.

[47] David Peral-García, Juan Cruz-Benito, and Francisco José García-Peñalvo. Systematic literature review: Quantum machine learning and its applications. *Computer Science Review*, 51:100619, 2024.

[48] Nimish Mishra, Manik Kapil, Hemant Rakesh, Amit Anand, Nilima Mishra, Aakash Warke, Soumya Sarkar, Sanchayan Dutta, Sabhyata Gupta, Aditya Prasad Dash, Rakshit Gharat, Yagnik Chatterjee, Shuvarati Roy, Shivam Raj, Valay Kumar Jain, Shreeram Bagaria, Smit Chaudhary, Vishwanath Singh, Rituparna Maji, Priyanka Dalei, Bikash K. Behera, Sabyasachi Mukhopadhyay, and Prasanta K. Panigrahi. Quantum machine learning: A review and current status. In Neha Sharma, Amlan Chakrabarti, Valentina Emilia Balas, and Jan Martinovic, editors, *Data Management, Analytics and Innovation*, pages 101–145, Singapore, 2021. Springer Singapore.

[49] Kwok Wan, Oscar Dahlsten, Hlér Kristjánsson, Robert Gardner, and M. Kim. Quantum generalisation of feedforward neural networks. *npj Quantum Information*, 3, 12 2016.

[50] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.

[51] Sonika Johri, Shantanu Debnath, Avinash Mocherla, Alexandros SINGK, Anupam Prakash, Jungsang Kim, and Iordanis Kerenidis. Nearest centroid classification on a trapped ion quantum computer. *npj Quantum Information*, 7, 12 2021.

[52] Kouhei Nakaji, Shumpei Uno, Yohichi Suzuki, Rudy Raymond, Tamiya Onodera, Tomoki Tanaka, Hiroyuki Tezuka, Naoki Mitsuda, and Naoki Yamamoto. Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicators. *Phys. Rev. Res.*, 4:023136, May 2022.

[53] Vittorio Pagni, Sigurd Huber, Michael Epping, and Michael Felderer. Fast Quantum Amplitude Encoding of Typical Classical Data. 3 2025.

[54] Gui-Lu Long and Yang Sun. Efficient scheme for initializing a quantum register with an arbitrary superposed state. *Phys. Rev. A*, 64:014303, Jun 2001.

[55] Martin Plesch and Časlav Brukner. Quantum-state preparation with universal gate decompositions. *Phys. Rev. A*, 83:032302, Mar 2011.

[56] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, 2018.

[57] Martín Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J. Coles, Lukasz Cincio, Jarrod R. McClean, Zoë Holmes, and M. Cerezo. Barren plateaus in variational quantum computing. *Nature Reviews Physics*, 7(4):174–189, 2025.

[58] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1):1791, 2021.

[59] Supanut Thanasilp, Samson Wang, Nhat Anh Nghiem, Patrick Coles, and Marco Cerezo. Subtleties in the trainability of quantum machine learning models. *Quantum Machine Intelligence*, 5(1):21, 2023.

[60] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*, 3:214, December 2019.

[61] Mikko Möttönen, Juha Vartiainen, Ville Bergholm, and Martti Salomaa. Transformation of quantum states using uniformly controlled rotations. *Quantum Information & Computation*, 5:467–473, 09 2005.

[62] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients. *Quantum*, 6:677, March 2022.

[63] M. J. D. Powell. *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, pages 51–67. Springer Netherlands, Dordrecht, 1994.

[64] J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

[65] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[66] Ville Bergholm et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. 11 2018.

[67] Jonas Stein, Jonas Blenninger, David Bucher, Peter J. Eder, Elif Cetiner, Maximilian Zorn, and Claudia Linnhoff-Popien. CUAOA: A Novel CUDA-Accelerated Simulation Framework for the QAOA . In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 280–285, Los Alamitos, CA, USA, September 2024. IEEE Computer Society.

[68] Gabin Schieffer, Stefano Markidis, and Ivy Peng. Harnessing CUDA-Q's MPS for Tensor Network Simulations of Large-Scale Quantum Circuits . In *2025 33rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, pages 94–103, Los Alamitos, CA, USA, March 2025. IEEE Computer Society.

[69] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, Los Alamitos, CA, USA, June 2022. IEEE Computer Society.

[70] Chun Gu. multinerf-pytorch: a pytorch implementation of multinerf, 2023.

[71] Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, Peter Hedman, Ricardo Martin-Brualla, and Jonathan T. Barron. MultiNeRF: A Code Release for Mip-NeRF 360, Ref-NeRF, and RawNeRF, 2022.

[72] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[73] Jiaxiang Tang. Torch-ngp: a pytorch implementation of instant-ngp, 2022. https://github.com/ashawkey/torch-ngp.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In abstract and introduction we claim that our proposed models allow for same or better performance compared to vanilla NeRF baseline, with less parameters and with higher convergence speed. This is in agreement with the experimental results obtained. In the introduction we clearly describe characteristics from the experimental setting (simulated hardware, low resolution scenes). We also clearly show the same in the teaser image (Figure 1).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations of this work in Section 6. Moreover, we made clear that experiments are conducted on simulated hardware and potential scalability issues (with regards to quantum capabilities) with one of the two models proposed. We also presented clearly the dataset preprocessing step (i.e. images downscaling), both in text and in visualizations, where all the images are presented with the resolution used in the experiments. We did not find limitations regarding privacy or fairness.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We did not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We ensured reproducibility to the best of our capabilities, by (1) describing in details models architecture (2) presenting all hyperparameters (3) giving all training details. We plan to release the full code after publication. In the additional material we submit a demo to reproduce the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We plan to public release the full code after publication. In the additional material we provide a demo to reproduce all the experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the details to reproduce the experiments are provided. Details regarding the dataset are given in Section 5, while models configuration and training details are provided in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All the results shown are computed for 5 runs, and presented as mean PSNR $\pm$ standard deviation. The factors of variability (i.e. parameters initialization, and train/test split for the LLFF dataset) are clearly presented in the text.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Time required and other relevant details are given in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We carefully read the code of ethics, finding no concerns.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We did not find direct path to negative applications for the task of novel-view synthesis.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We did not find high risk for misuse on the task of novel-view synthesis.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We used the datasets from the papers [1] and [24], which are cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We did not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human subjects were involved.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: No human subjects were involved.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: No LLMs were involved as any important, original, or non-standard component.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.