# Quantum 3D Visual Grounding: A Step Towards Quantum-inspired AI-Visualization

**Adib Bazgir** [1] **Rama chandra Praneeth Madugula** [2] **Yuwen Zhang** [1]

## Abstract

We introduce an advanced task of quantum 3D visual grounding in RGB images using language descriptions enriched with appearance and geometric information through quantum computing paradigms. In this work, we propose a framework which can enhance the existing classical 3D visual grounding techniques by leveraging the inherent parallelism and high-dimensional processing capabilities of quantum computing. This framework, Quantum3DVG, integrates quantum neural networks, including Quantum CNN (QCNN), Quantum Visual/Depth Encoder (QVDE), Quantum Text-Guided Visual/Depth Adapter (QTGVDA), and Quantum MLP (QMLP), to process both visual features and geometric data. At the heart of the proposed model, QVDE and QCNN encode image patches and depth information as quantum states, allowing for a high-level abstraction and quantum feature extraction. The QTGVDA is then re-envisioned as quantum circuit that refines these quantum states, employing quantum gates to align multi-scale visual and geometric features with textual descriptions. Finally, a quantum MLP is utilized for final object localization and classification.

## 1. Introduction

The interaction between humans and robots can be greatly enhanced if robots can understand language descriptions to find objects in complex, real-world 3D environments. While 2D visual grounding techniques have advanced (Deng et al., 2021; Yang et al., 2022; Zhan et al., 2023), they miss the depth aspect of understanding. Research has delved into using RGB-D and 3D scanning for indoor environments (Chen et al., 2020; Achlioptas et al., 2020), and recent work extends this to outdoor scenarios with LiDAR (Lin et al., 2023). However, these techniques are costly and have limitations.

[1] University of Missouri-Columbia [2] New York University. Correspondence to: Yuwen Zhang <zhangyu@missouri.edu>.

Published at ICML 2024 Workshop on Foundation Models in the Wild. 

While monocular 3D object detection (Huang et al., 2022; Brazil, 2023) is more accessible, it often lacks the rich semantic understanding of language needed for accurate object localization. A recently published work (Zhan et al., 2024) surpasses the constraints resulted from Refs (Huang et al., 2022; Brazil, 2023) by incorporating advanced transformer model equipped with multi-head attention mechanism. Even though this approach well semantically interprets the space and objects, it is still not only suffering from the necessity of extensive computational infrastructure required for training the model, but also lack of accuracy in pattern recognition while encountering high level of truncation and occlusion. The overall frameworks of 3D visual grounding through monocular and RGB-D images as input are displayed in Figure 1. Considering the potential challenges, we herein propose a conceptual framework which can significantly accelerate the training process using the fundamentals of the quantum state, quantum gate, and quantum circuit. This work also attempts to theoretically consider the application of quantum-base vision models improving the pattern recognition capabilities within high intensity level of truncation and occlusion. Finally, the proposed Quantum3DVG model can be integrated with visualization software to enhance the advanced post-processing purposes.

## 2. Related Work

### 2.1. 2D Visual Grounding

The field of grounding language descriptions in visual scenes, also known as referring expression comprehension, has been a major focus in computer vision and natural language processing. This task involves pinpointing a specific area within an image based on a textual description (Hu et al., 2016, Yu et al., 2018, Nagaraja et al., 2016). These descriptions can range from concise phrases (Plummer et al., 2015) to elaborate sentences (Mao et al., 2016). The standard approach typically involves two steps. First, potential object locations are generated, often using pre-trained object detectors or unsupervised methods. Then, the most likely objects are identified by comparing these regions to the description and ranking them based on their similarity (Zhang et al., 2018).

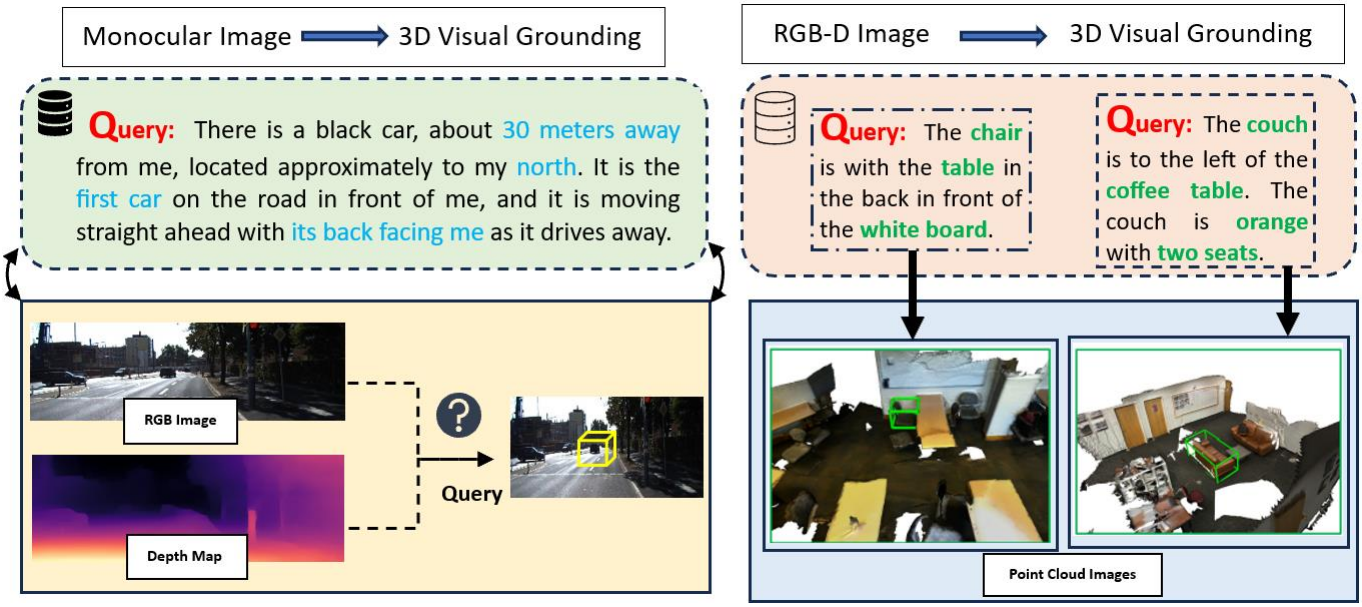



*Figure 1.* The schematic representation of 3D visual grounding associated with input types including monocular and RGB-D images.

A prominent area of research involves understanding the relationships between objects in the scene (Wang et al., 2019; Yang et al., 2019; Hu et al., 2017). For instance, Yang et al. (Yang et al., 2019) leveraged graph attention networks and a modular decomposition approach to establish connections between relationships and language expressions. MAttNet by Yu et al. introduced language-based and visual attention mechanisms to capture contextual information across modalities. While these methods excel at handling 2D vision and language reasoning tasks, they may not be well-suited for visual grounding on point clouds, where understanding 3D geometric relationships remain an under-explored area.

### 2.2. 3D Object Detection

In the realm of 3D scene understanding, pinpointing individual objects from a cloud of 3D data points is a crucial task. Researchers have proposed various detection techniques in recent times, including PointGroup (Jiang et al., 2020) and VoteNet (Qi et al., 2019). A study by (Chen et al., 2022) highlighted a powerful object detector which can have significant impact on the vision language system's overall performance. Their approach utilized PointGroup as the foundation for detection. However, PointGroup and similar backbones (like PointNet (Qi et al., 2017) and its variants) often struggle to differentiate between objects in close proximity. SoftGroup (Vu et al., 2022) recently addressed this limitation, achieving impressive results in the ScanNet (Dai et al., 2017) instance segmentation challenge. Unlike traditional clustering methods like PointGroup, SoftGroup boasts enhanced robustness and flexibility. This is achieved through a "soft assignment" approach, where each data point can partially belong to multiple clusters. This creates a more refined data representation, potentially leading to superior performance.

### 2.3. 3D Visual Grounding

Pinpointing a specific object in a 3D scene based on a textual description is a complex task known as 3D visual grounding. Pioneering datasets like Scanrefer and Referit3D paved the way for research in this area. Similar to its 2D counterpart, early approaches relied on a two-stage pipeline. This involved using pre-trained object detectors, like PointNet++ (Qi et al., 2017), to generate candidate objects and extract features. Later works, like SAT (Yang et al., 2021), incorporated 2D object information to bolster training. InstanceRefer (Yuan et al., 2021) reframed the task as an instance matching problem. To directly interpret intricate and varied descriptions within point clouds, Feng et al. (Feng et al., 2021) proposed a novel approach that constructs three interrelated graphs: a language scene graph, a 3D proposal relation graph, and a 3D visual graph. Transformer-based architectures have also gained traction, with models like 3DVG-Trans (Zhao et al., 2021), TransRefer3D (He et al., 2016), Multi-View Trans (Huang et al., 2022), and LanguageRefer (Roh et al., 2022) demonstrating promising results. D3Net and 3DJCG introduced unified frameworks for both dense captioning and visual grounding tasks. While previous efforts primarily focused on indoor environments with furniture as the target object, recent advancements aim to broaden the application scope. Lin et al. (Lin et al., 2023) introduced a large-scale outdoor scene grounding task that leverages online-captured 2D images and 3D point clouds. However, acquiring visual data through LiDAR or specialized cameras remains expensive and impractical for many scenarios. This motivates our work, which explores 3D visual grounding using single images as input.

## 3. Methodology

As shown in Figure 2, we conceptually propose an end-to-

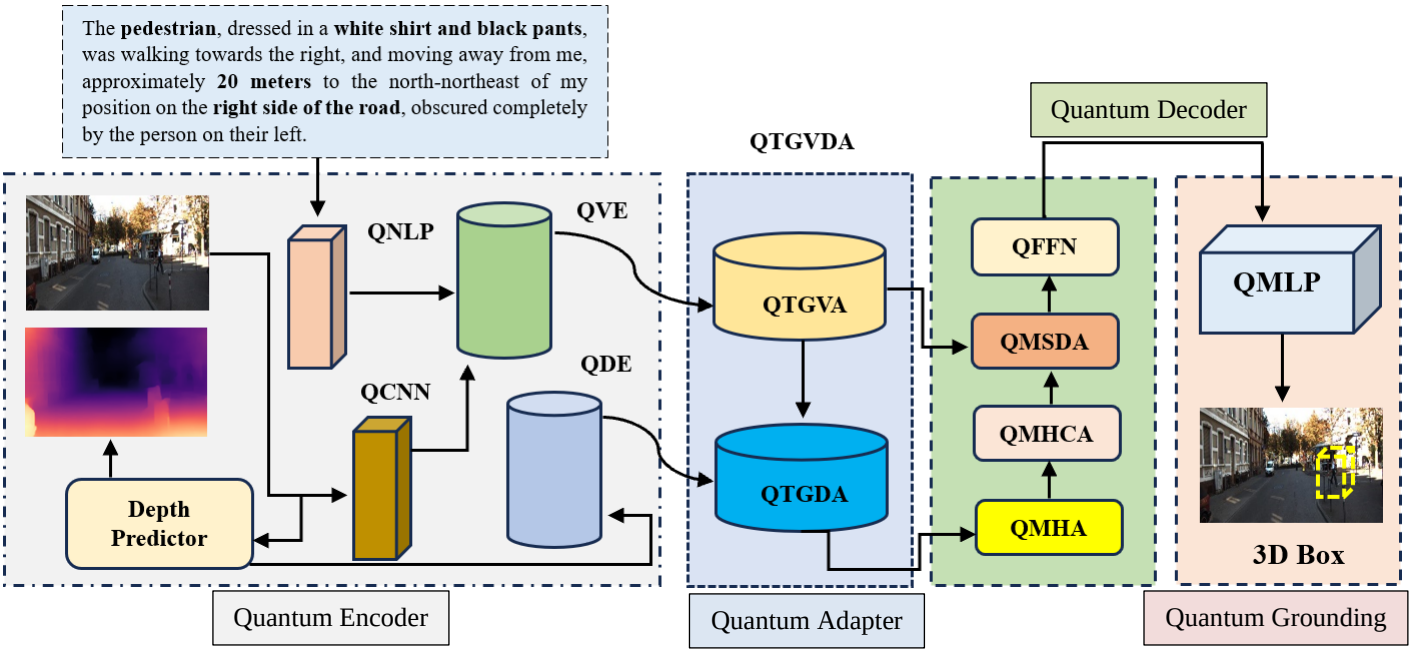



*Figure 2.* The schematic representation of quantum 3D visual grounding containing quantum vision and language models.

end quantum-based framework, Quantum3DVG, which consists of four main modules: 1) the quantum encoder; 2) the quantum adapter; 3) the quantum decoder; 4) the quantum grounding. The quantum encoder partition consists of quantum natural language processing (QNLP), quantum convolutional neural networks (QCNN), quantum vision encoder (QVE), and quantum depth encoder (QDE). For quantum adapter or QTGVDA, it contains quantum text-guided vision adapter (QTGVA) and quantum text-guided depth adapter (QTGDA). In quantum decoder partition, the components are quantum multi-head attention (QMHA), quantum multi-head cross attention (QMHCA), quantum multi-scale deformable attention (QMSDA), and quantum feed forward network (QFFN), followed by quantum multilayer perceptron (QMLP) in quantum grounding partition.

Diving deeper into the complexities and nuances of the Quantum3DVG model, we explore how each component functions within a quantum computational framework to enhance the process of 3D visual grounding using quantum vision and quantum natural language descriptions. The intersection of quantum computing with machine learning here aims to leverage the fundamental aspects of quantum mechanics, such as quantum entanglement and superposition, to process information in ways that could dramatically outstrip the capabilities of classical computational methods. Furthermore, more details concerning each partition has been provided in the following sections.

### 3.1. Quantum Multi-Modality of Encoder

#### 3.1.1. Quantum Visual Encoder (QVE)

The QVE commences with the conversion of classical image data into quantum data. This is not a trivial step—each image patch must be carefully encoded into quantum states without losing essential information. Techniques like amplitude encoding, where pixel intensities are translated into the probability amplitudes of quantum states, or angle encoding, where pixel values determine the rotations in quantum gate operations, are considered. Each encoding scheme has its advantages and trade-offs regarding computational resources and the fidelity of the represented data. After encoding, a sequence of QCNN layers applies quantum convolution operations. These are not direct analogs to classical convolutions but rather innovative quantum operations that consider the multi-level interactions between qubits. These interactions are non-local, a key property that could allow for a more holistic representation of visual features as compared to their classical counterparts. The QVE would employ quantum circuits to encode visual information from image patches. These quantum circuits can be thought of as QMHCA, QMSDA, QMHSA, and QFFN modules seen in Figure 3 (Appendix (A-1)). In the quantum realm, these modules would leverage quantum gates to perform entangled transformations on qubits that represent the image data, effectively enabling simultaneous processing of multiple spatial scales and feature representations. This is an extension of the classical approach, where different filters capture different aspects of the input data.

#### 3.1.2. Quantum Depth Encoder (QDE)

For depth processing, the QDE transforms depth information

into quantum states, which are then manipulated by a series of quantum transformer blocks such as QMHSA and QFFN. These blocks, reminiscent of the self-attention mechanism in classical transformers, perform a quantum version of this operation, where the attention weights are calculated through the interactions of quantum states. The transformer's ability to handle sequences is adapted to the quantum domain, providing a mechanism for interpreting quantum states in a manner that preserves spatial hierarchies and relationships critical for depth perception.

### 3.2. Quantum Text-Guided Adapter

#### 3.2.1. Quantum Text-Guided Visual Adapter (QTGVA)

The QTGVA, as depicted in Figure 4 (Appendix (A-2)), stands out as a particularly intriguing component, taking on the challenge of integrating classical text information into a quantum process. This adapter acts as a translator, using text embeddings as a guide to modulate quantum operations on the visual states. This modulation involves a dynamic adjustment of the quantum gates' parameters based on the language input, allowing the adapter to focus on and enhance the quantum states that correspond to the textual descriptions. The QTGVA within the architecture functions to coalesce visual features with textual cues at the quantum state level. In this quantum construct, the classical Normalization and Addition operations are intrinsically accounted for through the unitary nature of quantum operations, maintaining the requisite normalization of quantum states post-transformation. To incorporate the guidance from textual inputs, a hybrid quantum-classical interaction is employed, wherein classical text data modulates quantum operations, altering the state evolution in a controlled manner.

Furthermore, the Quantum Multi-Scale Deformable Attention (QMSDA) module is designed to replace the classical Multi-Scale Deformable Attention, employing a set of entangling gates that process spatially varying features across different scales. These gates induce a coherent manipulation of qubits corresponding to diverse visual features, simulating the adaptable convolutional process to the geometry of the input. Additionally, a pixel-wise attention mechanism is emulated using a controlled sequence of quantum gates. These gates selectively enhance or suppress features within the quantum states, modulated by pixel-level textual information, akin to the classical pixel-wise attention process. The QMHCA has a complex pattern of entanglements across multiple qubit registers. The QMHCA allows for the concurrent processing of disparate visual feature sets, establishing a holistic understanding of the visual data with respect to the textual descriptions. Lastly, quantum up/downsampling is conceptualized to manipulate the resolution of quantum-encoded feature representations. Further details concerning QTGVA and the rest of baseline models like QTGDA, QD, QMLP, QCNN, and QNLP are elaborated in Appendix (A).

## 4. Experiments (Quantum3DVG vs Classical3DVG)

In this study, we use a hybrid combination of human and vehicle types of datasets for outdoor locations using ChatGPT, Gemini, and Perplexity along with manual manipulation strategies which has been shown in Figure 10. More quantitative details of utilized dataset and further ablation study of each Quantum3DVG model's component can be found in Appendix (B) and Appendix (C). As a part of primary results, the 3D object detection and localization associated with Quantum3DVG (Q3DVG) and Classical3DVG (C3DVG) are visually compared in Figure 11. It can be clearly seen that the Q3DVG model is capable of accurately depicting 3D boxing compared to existing C3DVG model, while it also enjoys numerous advantages such as model training time decrement and clearer object detection and localization. Further qualitative testaments of Q3DVG model are provided in Appendix (D) proving that the quantum model has numerous advantages compared its classical counterpart known as C3DVG.

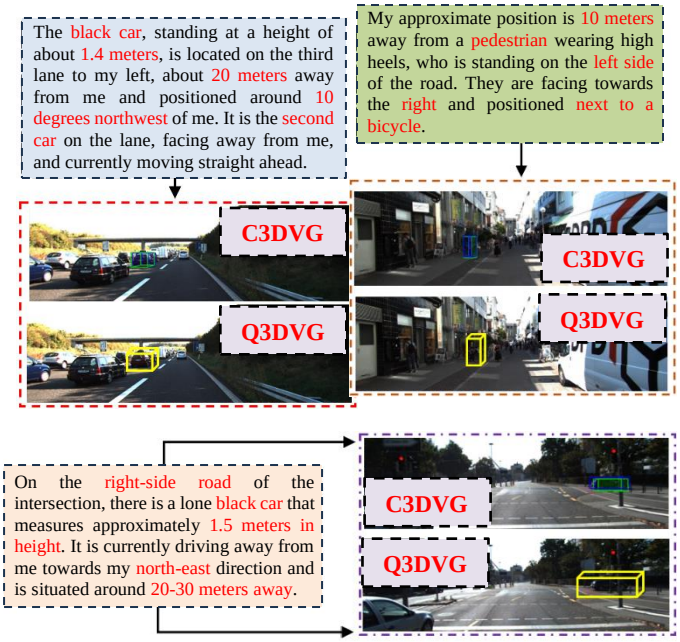



*Figure 11.* Primary results from Q3DVG and C3DVG models.

## 5. Conclusion

This study introduces the Quantum3DVG model, a novel approach that integrates quantum computing into 3D visual grounding. Our preliminary findings highlight the model's improved computational efficiency and robustness compared to classical methods, particularly in vision recognition and rapid feature processing.

## References


Deng, J., Yang, Z., Chen, T., Zhou, W., and Li, H. Transvg: End-to-end visual grounding with transformers. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Yang, L., Xu, Y., Yuan, C., Liu, W., Li, B., and Hu, W. Improving visual grounding with visual-linguistic verification and iterative reasoning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Zhan, Y., Xiong, Z., and Yuan, Y. RSVG: Exploring data and models for visual grounding on remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 61(2): 1-13, 2023.

Chen, D. Z., Chang, A. X., and Nießner, M. Scanrefer: 3d object localization in rgb-d scans using natural language. *In European conference on computer vision*, 2020.

Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., and Guibas, L. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. *In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings*, 2020.

Lin, Z., Peng, X., Cong, P., Hou, Y., Zhu, X., Yang, S., et al. WildRefer: 3D Object Localization in Large-scale Dynamic Scenes with Multi-modal Visual Data and Natural Language. *arXiv preprint arXiv:2304.05645*, 2023.

Huang, K.-C., Wu, T.-H., Su, H.-T., and Hsu, W. H. Monodtr: Monocular 3d object detection with depth-aware transformer. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Brazil, G., Kumar, A., Straub, J., Ravi, N., Johnson, J., and Gkioxari, G. Omni3d: A large benchmark and model for 3d object detection in the wild. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023.

Zhan, Y., Yuan, Y., and Xiong, Z. Mono3dvg: 3d visual grounding in monocular images. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., and Darrell, T. Natural language object retrieval. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., et al. Mattnet: Modular attention network for referring expression comprehension. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

Nagaraja, V. K., Morariu, V. I., and Davis, L. S. Modeling context between objects for referring expression understanding. *In Computer Vision–ECCV 2016: 14th European Conference*, 2016.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *In Proceedings of the IEEE international conference on computer vision*, 2015.

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., and Murphy, K. Generation and comprehension of unambiguous object descriptions. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Zhang, H., Niu, Y., and Chang, S.-F. Grounding referring expressions in images by variational context. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Wang, P., Wu, Q., Cao, J., Shen, C., Gao, L., and Hengel, A. v. d. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Yang, S., Li, G., and Yu, Y. Cross-modal relationship inference for grounding referring expressions. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

Hu, R., Rohrbach, M., Andreas, J., Darrell, T., and Saenko, K. Modeling relationships in referential expressions with compositional modular networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

Yang, S., Li, G., and Yu, Y. Dynamic graph attention for referring expression comprehension. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.-W., and Jia, J. Pointgroup: Dual-set point grouping for 3d instance segmentation. *In Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, 2020.

Qi, C. R., Litany, O., He, K., and Guibas, L. J. Deep hough voting for 3d object detection in point clouds. *In proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Chen, D. Z., Wu, Q., Nießner, M., and Chang, A. X. D 3 net: A unified speaker-listener architecture for 3d dense

captioning and visual grounding. *In European Conference on Computer Vision*, 2022.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

Vu, T., Kim, K., Luu, T. M., Nguyen, T., and Yoo, C. D. Softgroup for 3d instance segmentation on point clouds. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 2017.

Yang, Z., Zhang, S., Wang, L., and Luo, J. Sat: 2d semantics assisted training for 3d visual grounding. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Yuan, Z., Yan, X., Liao, Y., Zhang, R., Wang, S., Li, Z., et al. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Feng, M., Li, Z., Li, Q., Zhang, L., Zhang, X., Zhu, G., et al. Free-form description guided 3d visual graph network for object grounding in point cloud. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Zhao, L., Cai, D., Sheng, L., and Xu, D. 3DVG-Transformer: Relation modeling for visual grounding on point clouds. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Huang, S., Chen, Y., Jia, J., and Wang, L. Multi-view transformer for 3d visual grounding. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Roh, J., Desingh, K., Farhadi, A., and Fox, D. Languagerefer: Spatial-language model for 3d visual grounding. *In Conference on Robot Learning*, 2022.

Cai, D., Zhao, L., Zhang, J., Sheng, L., and Xu, D. 3djcg: A unified framework for joint dense captioning and visual grounding on 3d point clouds. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Liu, H., Lin, A., Han, X., Yang, L., Yu, Y., and Cui, S. Refer-it-in-rgbd: A bottom-up approach for 3d visual grounding in rgbd images. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Pandey, S., Basisth, N. J., Sachan, T., Kumari, N., and Pakray, P. Quantum machine learning for natural language processing application. *Physica A: Statistical Mechanics and its Applications*, 627(3): 129123, 2023.

Friedrich, L. and Maziero, J. Quantum neural network with ensemble learning to mitigate barren plateaus and cost function concentration. *arXiv preprint arXiv:2402.06026*, 2024.

Shao, C. A quantum model for multilayer perceptron. *arXiv preprint arXiv:1808.10561*, 2018.

Kerenidis, I., Landman, J. and Prakash, A. Quantum algorithms for deep convolutional neural networks. *arXiv preprint arXiv:1911.01117*, 2019.

Scala, F., Ceschini, A., Panella, M. and Gerace, D. A general approach to dropout in quantum neural networks. *Advanced Quantum Technologies*, 2023.

# Appendix (A): Additional Model Architecture and Relevant Details

## Appendix (A-1): Quantum Vision Encoder (QVE) and Quantum Depth Encoder (QDE)

As can be observed in Figure 3, the overall schematic of a Quantum Vision Encoder (QVE) Model and a Quantum Depth Encoder (QDE) Model, both of which are components of a larger quantum-based multimodal model are presented. A detailed analytical description of the QVE and QDE schematics are given below:

The QVE Model consists of three main components: QMHCA (Quantum Multi-Head Cross Attention), QMSDA (Quantum Multi-Scale Deformable Attention), and QFFN (Quantum Feed-Forward Network). The QMHCA component likely performs cross-attention operations to capture relationships between different parts of the input data, utilizing multiple attention heads to focus on various aspects of the input simultaneously. This enhances the model's ability to capture diverse and complex patterns. The QMSDA module aggregates data at multiple scales using deformable attention mechanisms, which are essential for capturing information at various levels of detail and allowing the model to adaptively focus on important features across different resolutions. The QFFN component processes the aggregated data using a feed-forward network, likely consisting of several layers of quantum neurons (qubits and quantum gates) that apply transformations to the input data, enhancing its representation before passing it to the output layer. The data flow in the QVE model starts with the input being processed by the QMHCA module to extract relational features. These features are then passed to the QMSDA module for multi-scale deformable aggregation, and finally, the aggregated data is fed into the QFFN module, which transforms it into the output.

The QDE Model, on the other hand, consists of two main components: QMHSA (Quantum Multi-Head Self-Attention) and QFFN (Quantum Feed-Forward Network). The QMHSA component implements self-attention mechanisms, allowing the model to focus on different parts of the input sequence when encoding depth information. With multiple self-attention mechanisms applied simultaneously, the model can better learn intricate dependencies within the input data. Similar to the QFFN in the QVE model, the QFFN in the QDE model processes data using a feed-forward network structure, further transforming the encoded depth information and preparing it for output or further processing. The data flow in the QDE model starts with the input being processed by the QMHSA module, where self-attention mechanisms capture dependencies within the input data. The attended data is then passed to the QFFN module for further transformation, and the output from the QFFN can be directed to other components or used as the final output.

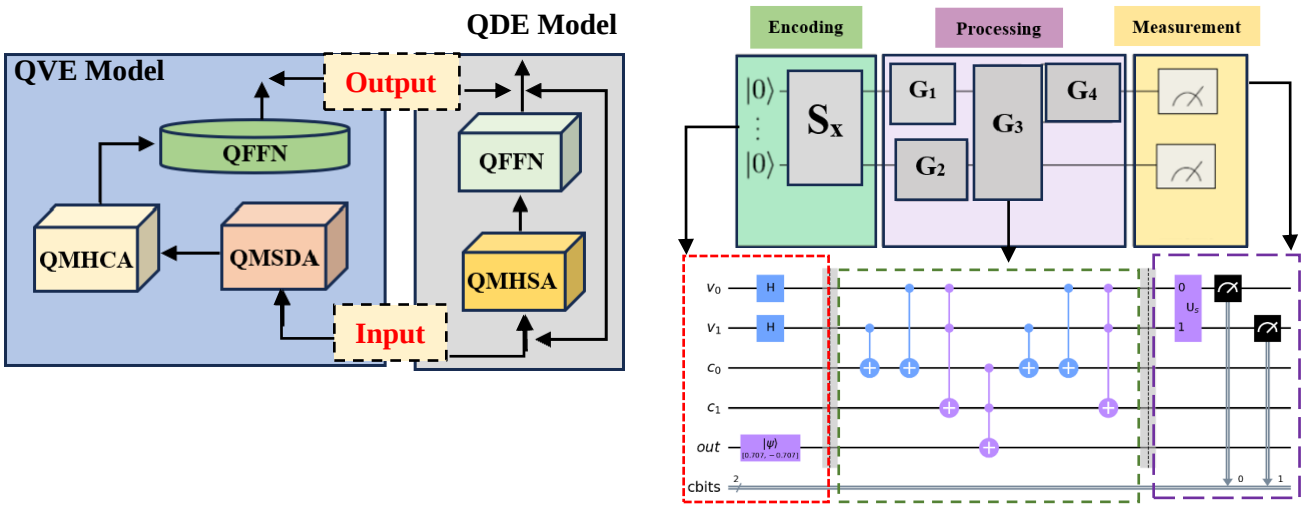



*Figure 3.* The schematic of QVE/QDE and the quantum circuit function.

The quantum circuit as an inherent partition of QVE and QDE components is depicted in the right-side of Figure 3. The circuit is divided into three main sections: Encoding, Processing, and Measurement. In the Encoding section, the process begins with the preparation of initial quantum states, represented by |0⟩ states. A set of operations, denoted as

$S_X$, is applied to these initial states to encode the input data into a quantum state suitable for further processing. The circuit shows the application of Hadamard gates (H) to the quantum bits $v_0$ and $v_1$, creating superpositions of $|0\rangle$ and $|1\rangle$. This step is crucial for generating a superposition state that can be used for parallel processing of information in the quantum circuit.

The Processing section includes multiple quantum gates represented by $G_1$, $G_2$, $G_3$, and $G_4$. These gates likely represent various quantum operations that transform the encoded quantum states, such as entanglement, phase shifts, or more complex unitary transformations. The circuit includes several CNOT gates, which are used to create entanglement between qubits. The control qubit (indicated by a filled circle) influences the state of the target qubit (indicated by $\oplus$). The placement of these CNOT gates in the circuit suggests a pattern of entanglement and conditional operations that form the core of quantum processing. The quantum state $|\psi\rangle$ represents the intermediate state of the quantum bits after undergoing the operations in the processing section. This state encapsulates the complex superpositions and entanglements created by the applied gates. In the Measurement section, measurement operators are included that collapse the quantum state into classical bits. Measurements are typically performed in the computational basis ($|0\rangle$ or $|1\rangle$) and are indicated by meter symbols in the diagram. The final part of the circuit shows the measurement of the quantum bits, transforming the quantum information into classical bits ($c_0$ and $c_1$) that can be read out. The measured values are then interpreted as the output of the quantum circuit.

The overall flow of the quantum circuit starts with the encoding of initial states using Hadamard gates to create superpositions. The superposition states undergo a series of quantum operations in the processing section, involving various gates that perform transformations and create entanglements. The final quantum state, after processing, is measured to obtain classical outputs that represent the result of the quantum computation. This quantum circuit is designed to leverage quantum parallelism and entanglement to process information efficiently, aligning with the goals of the Quantum Vision Encoder (QVE) and Quantum Depth Encoder (QDE) models.

## Appendix (A-2): Quantum Text-Guided Visual Adapter (QTGVA) and Quantum Text-Guided Depth Adapter (QTGDA)

As given in Figure 4, the schematic illustration of two components such as Quantum Text-Guided Visual Adapter (QTGVA) and the Quantum Text-Guided Depth Adapter (QTGDA) is further visualized. The Quantum Text-Guided Visual Adapter (QTGVA) begins with an input signal, which can be a combination of textual and visual data. This input is first processed by the QMHCA (Quantum Multi-Head Cross Attention) module. This module is responsible for integrating information from multiple heads of attention mechanisms, leveraging quantum principles for efficient handling of high-dimensional data and facilitating cross-modal interaction between visual and textual data. Next, the data moves to the QMSDA (Quantum Multi-Scale Deformable Attention) module, which adapts to different scales and deforms the attention mechanism to focus on relevant parts of the data at varying scales. This helps in capturing multi-scale features crucial for understanding complex visual inputs. Following this, the data is further refined by the Q-Attention mechanism, which ensures that the most relevant features are highlighted, and irrelevant noise is minimized. The processed data then undergoes quantum-based up-sampling or down-sampling in the Q-Up/Down Sampling module to match the required dimensions for subsequent processing, maintaining data integrity while resizing. Finally, the QTGV (Quantum Text-Guided Vision) integrates the outputs from the attention mechanisms and the sampling process, ensuring that the visual data is accurately interpreted and transformed in a way guided by the accompanying textual data.

The Quantum Text-Guided Depth Adapter (QTGDA) starts with an input that includes depth information guided by text. This input is first processed by the QMHA (Quantum Multi-Head Attention) module, which integrates depth information through multiple attention heads, designed to handle high-dimensional depth data efficiently using quantum principles. The data then moves to the QMHCA (Quantum Multi-Head Cross Attention) module to facilitate cross-modal interaction, focusing on depth information in conjunction with textual data. Finally, the QTGDA (Quantum Text-Guided Depth Adapter) integrates these processed elements to produce the final depth-related output. In terms of connections and workflow, the data flow in QTGVA involves processing the input through QMHCA and QMSDA, handling cross-attention and multi-scale deformable attention respectively. These outputs are refined by the Q-Attention mechanism and adjusted in scale by the Q-Up/Down Sampling module, with QTGV integrating all these processed data elements to produce the final output. Similarly, in QTGDA, the depth information is processed by QMHA and QMHCA, with QTGDA integrating these processed elements to produce the final depth-related output. The outputs from both QTGVA and QTGDA can be used together to provide a comprehensive understanding of the input data by integrating visual, textual, and depth

information efficiently using quantum principles.

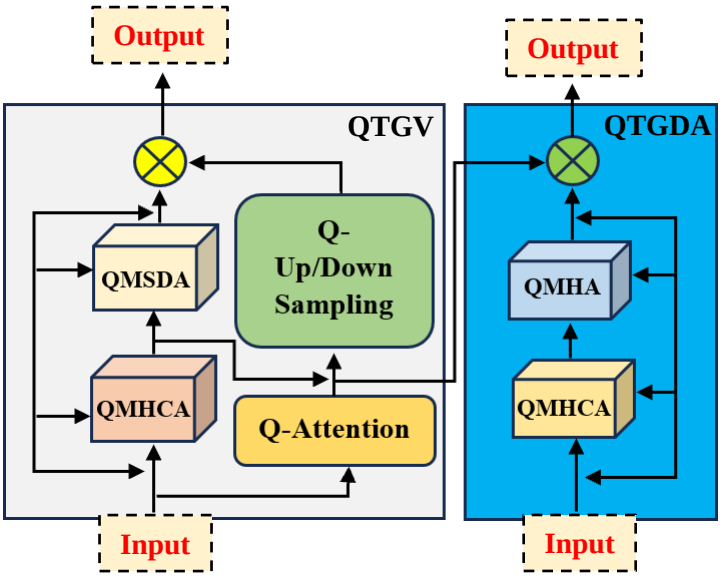



*Figure 4.* The schematic of QTGVA and QTGDA.

## Appendix (A-3): Quantum Decoder (QD)

As provided in Figure 5, the decoder section of the model brings the multi-modal quantum states together. Using quantum circuits that implement a type of quantum attention mechanism in the forms of QMHA, QMHCA, QMSDA, and QFFN, the quantum decoder directs focus within the entangled states, picking out the relevant features for object localization. The quantum parallelism at play here could allow the QD to consider a vast array of feature combinations and relationships simultaneously, a feat unattainable in classical computing.

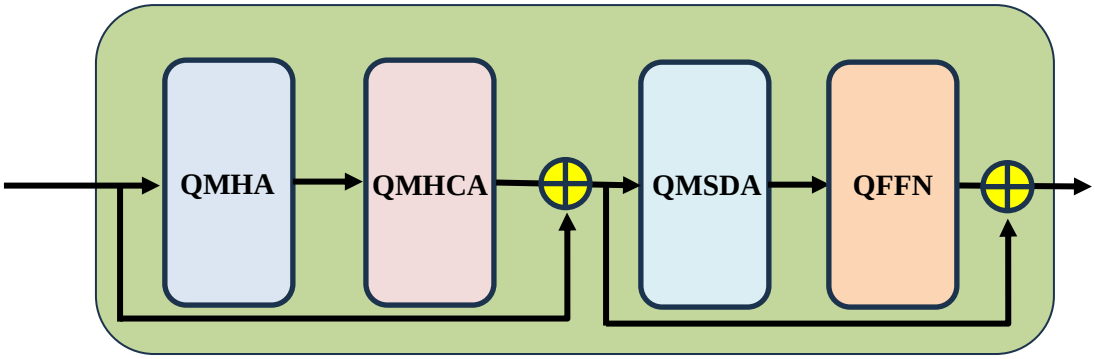



*Figure 5.* The detailed schematic of the QD within its quantum components.

## Appendix (A-4): Quantum MLP (QMLP)

At the culmination of the model, the QMLP translates the refined quantum states from the QD into a form usable for classical interpretation. The QMLP performs this task by applying a series of quantum gates arranged in layers, which non-linearly manipulate the quantum states. The measurement process at the output of these layers collapses the quantum states into classical information, which correlates to the predicted 3D bounding boxes (Friedrich et al., 2024; Shao, 2018).

As shown in Figure 6, it illustrates a comprehensive QMLP architecture specifically designed for this study. The bottom section of Figure 6 visualizes the quantum states using Bloch spheres. Each Bloch sphere represents the state of a qubit, illustrating the probabilities of the qubit being in state $|0\rangle$ or $|1\rangle$, as well as their superpositions. The Bloch spheres offer an intuitive way to understand the state transformations that occur during the processing in QNNs and QMLP. The visualization shows how input data is mapped onto the quantum states, highlighting the quantum encoding and processing steps. The general state of a qubit $|\psi\rangle$ can be described by the equation: $|\psi\rangle = \cos(\theta/2)\,|0\rangle + e^{i\phi}\sin(\theta/2)\,|1\rangle$ where $\theta$ and $\phi$ are the spherical coordinates on the Bloch sphere. The $z$-axis represents the computational basis states $|0\rangle$ and $|1\rangle$, while the $x$- and $y$-axes represent the superpositions of these states, such as $|+\rangle = (|0\rangle + |1\rangle) / 2$ and $|-\rangle = (|0\rangle - |1\rangle) / 2$. Pure states are depicted as points on the surface of the sphere, whereas mixed states, which are probabilistic combinations of pure states, would lie inside the sphere (though these are not depicted in this Figure).

Each Bloch sphere at the bottom of Figure 6 represents the quantum states of the qubits after encoding and processing through the Quantum Neural Networks (QNNs) and the Quantum MLP. Initially, qubits are in a definite state, usually $|0\rangle$, depicted at the north pole of the Bloch sphere. As data is encoded into the quantum states, the qubits move from their initial state to various points on the Bloch sphere, depending on the encoding scheme (angle or amplitude encoding). The encoded state reflects the information from the input data. For instance, if amplitude encoding is used, the probability amplitudes of the qubit states correspond to the input data values.

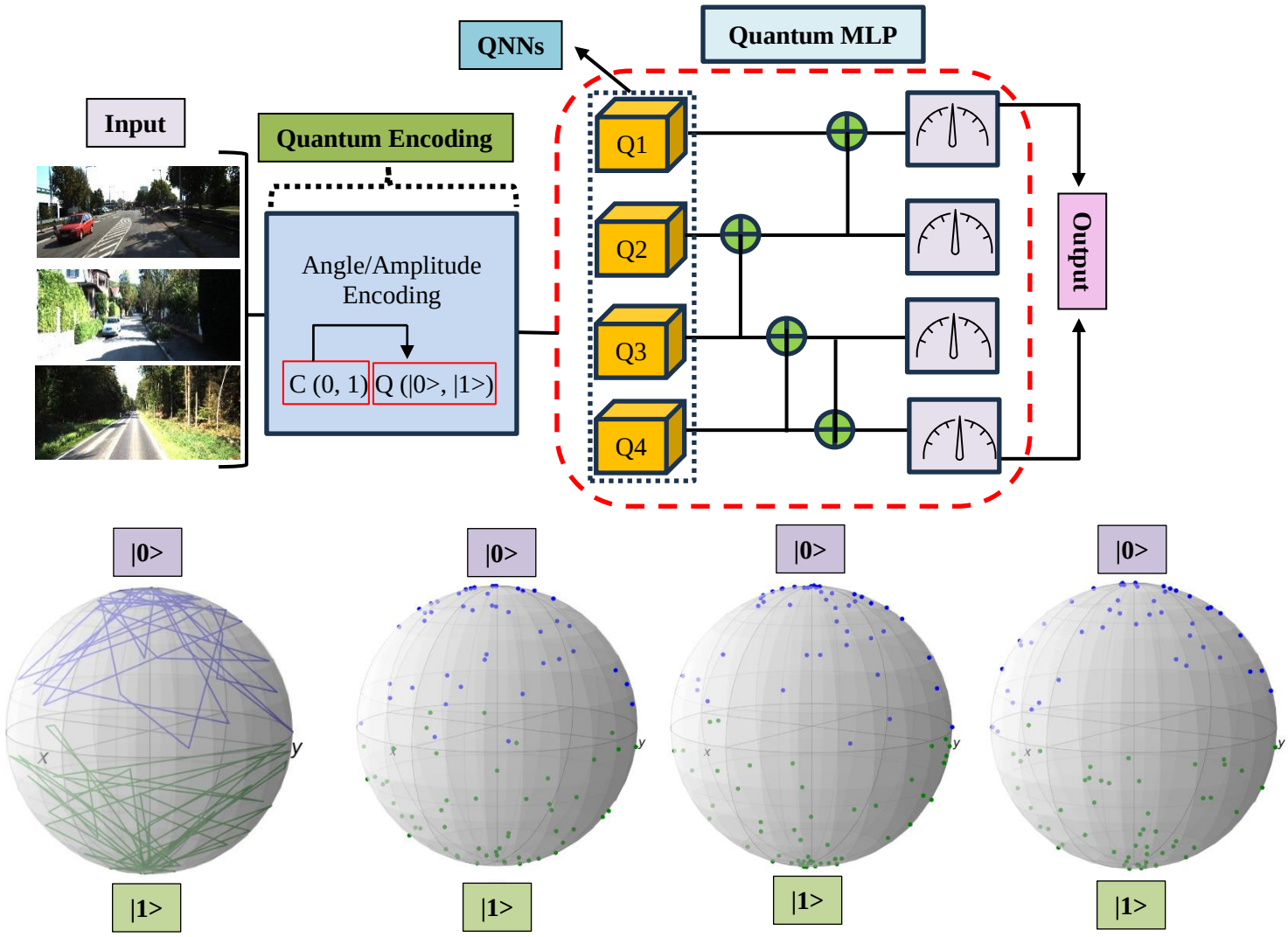



*Figure 6.* The detailed schematic of the QMLP model along with its quantum components and encoded qubits.

## Appendix (A-5): Quantum Convolutional Neural Network (QCNN)

### Quantum CNN Layer:

As shown in Figure 7, the QCNN architecture leverages the quantum computational framework to enhance the processing of complex data. The QCNN comprises several layers, each performing specific operations on the input data. The input to the QCNN is a multispectral image, capturing various parameters. This image undergoes preprocessing, such as normalization and dimension adjustments, to ensure compatibility with the quantum convolutional layers. The first set of layers consists of Quantum Convolutional 1D (QCONV 1D) operations. The first QCONV 1D layer applies a quantum convolutional operation to the input data, extracting basic features from the multispectral image. This layer uses 32 filters with a kernel size of 3x3, a stride of 1, and same padding. The second QCONV 1D layer further refines features extracted by the first layer, capturing more complex patterns. This layer employs 64 filters with the same kernel size, stride, and padding as the first layer. Following the QCONV 1D layers, Quantum ReLU (Q-ReLU) activation functions are applied. The first Q-ReLU activation introduces non-linearity after the first QCONV 1D layer, enabling the network to learn complex data representations. The second Q-ReLU activation follows the second QCONV 1D layer, further enhancing the network's capability to model intricate patterns. Quantum Dropout (Q-Dropout) layers are incorporated to prevent overfitting. The first Q-Dropout layer randomly deactivates 20% of quantum gates during training, ensuring the network generalizes well to unseen data. The second Q-Dropout layer similarly deactivates 20% of quantum gates in deeper layers, reducing overfitting risks.

The core quantum computational block includes a variational ansatz, a parameterized quantum circuit designed to approximate the desired quantum state transformations. This block features specific quantum gates such as $Rx$, $Ry$, and $Rz$, which apply rotations around the respective axes on the Bloch sphere. Entanglement operations, depicted by CNOT gates, create correlations between qubits, enhancing the network's capacity to model complex dependencies. The parameters within the variational ansatz are optimized during training to minimize the loss function. The final layer integrates the processed information, producing output predictions for various parameters (Kerenidis et al., 2019).

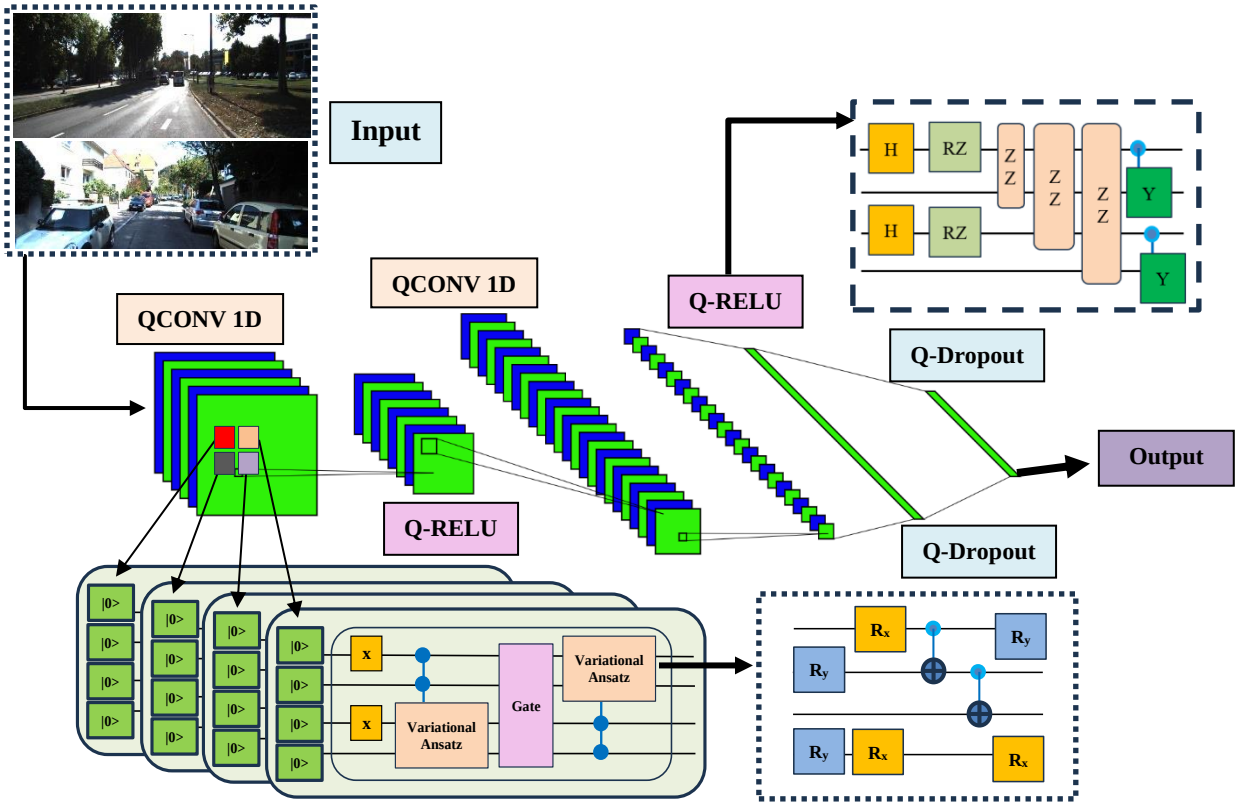



*Figure 7.* The detailed schematic of the QCNN model along with its quantum components.

More specifically, the QCNN layer's key steps and components involved in the quantum forward and backward passes for a QCNN layer are summarized as below, and the detailed discussion is given in Algorithm 1:

1. **Quantum Convolution Product**:

The quantum convolution product is the primary operation, mapping the convolution process from classical CNNs to a quantum framework. It uses a mapping between the convolution of tensors and matrix multiplication, which can be reduced to inner product estimation between vectors.

2. **Inner Product Estimation**:

This involves calculating the inner product between the input and kernel tensors using quantum states. The inner product is estimated using amplitude estimation and median evaluation algorithms to ensure accuracy.

3. **Non-Linearity**:

After the convolution, a non-linear activation function (e.g., Q-ReLU) is applied. This is implemented using quantum circuits to handle the non-linearity, which is essential for the learning capability of neural networks.

4. **Quantum Sampling**:

The output of the quantum convolution is a quantum state representing the result of the convolution product. To retrieve meaningful classical information, quantum sampling techniques are used. This involves conditional rotations and amplitude amplification to enhance the probability of measuring important data points.

5. **Quantum Tomography**:

To convert the quantum state back to a classical form, quantum tomography with $\ell\infty$ norm guarantees is employed. This process ensures that the classical output closely approximates the desired results from the quantum state.

6. **Pooling Operation**:

The pooling operation, which reduces the dimensionality of the data, is integrated into the QCNN structure. This can be performed during the QRAM update phase and includes techniques like maximum pooling or average pooling.

**Quantum Dropout Technique:**

In this section, we present a comprehensive discussion of various quantum dropout strategies, as illustrated in Figure 8. Quantum dropout is an essential technique for regularizing quantum neural networks, akin to classical dropout in conventional neural networks. Each strategy employs a unique approach to dropping gates, thereby affecting the network's overall performance and robustness (Scala et al., 2023). In this study, we employ a hybrid policy of given quantum dropout techniques in Figure 8.

**Canonical dropout** involves dropping a single rotation gate $R_G$ along with all preceding entangling gates $E_G$ that targeted the particular qubit, and all subsequent entangling gates that used that qubit as a control. As shown in Figure 8, single dropped gates are highlighted by circles/rectangles, with arrows indicating the sequence of dropped gates. This dropout strategy minimizes the network's dependency on specific qubits, potentially enhancing model generalization. The dropping probability $p_G$ is employed together with $p_L$ to obtain the overall dropping probability $p = p_G p_L$.

**Canonical-forward dropout** involves dropping a single rotation gate $R_G$ along with all subsequent entangling gates $E_G$ that used that qubit as a control. Illustrated in Figure 8, this method mitigates the influence of future entanglements involving the dropped rotation gate, reducing error propagation through the network. The same dropping probability $p_G$ is utilized.

**Independent dropout** works by dropping a single rotation gate $R_G$ and a single entangling gate $E_G$ independently of each other. Figure 8 demonstrates this approach, applying dropout independently to different types of gates, potentially balancing the influence of rotation and entangling gates on the network. This method employs distinct probabilities $p_R$ for rotation gates and $p_E$ for entangling gates.

**Rotation dropout** involves dropping single rotation gates $R_G$ alone. As shown in Figure 8, this straightforward approach simplifies the quantum circuit by focusing solely on rotation gates, crucial for qubit state manipulation. The probability $p_R$ is used to determine the dropping of rotation gates.

**Entangling dropout** involves dropping single entangling gates $E_G$ alone. Illustrated in Figure 8, this strategy targets entangling gates, pivotal for creating quantum correlations between qubits. Dropping these gates reduces the complexity of the quantum entanglement structure. The probability $p_E$ is used for entangling gate dropout.

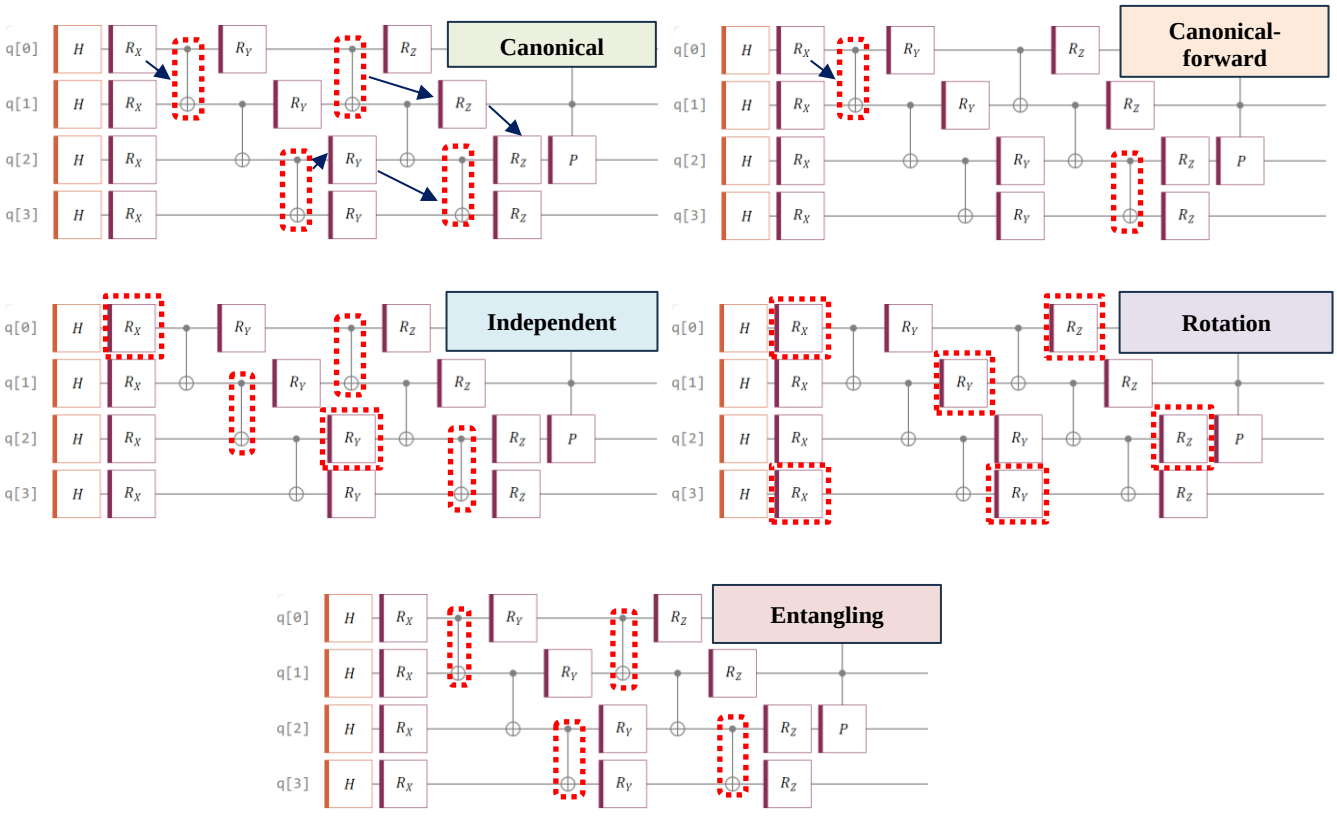
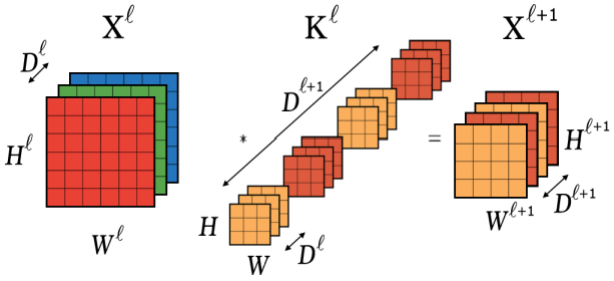



*Figure 8.* Various quantum dropout techniques utilized in our work.

**Algorithm 1 (Combined Algorithm):** QCNN with Quantum Backpropagation

**Input:** Data input matrix $A^{\ell}$, kernel matrix $F^{\ell}$, precision parameters $\epsilon$, $\eta$, and $\delta$, non-linearity function $f$, learning rate $\lambda$.

**Output**: Updated data matrices $A^{\ell+1}$ and kernel matrices $F^{\ell}$.




1. **Forward Pass (Quantum Convolution):**
   - **Inner Product Estimation:**

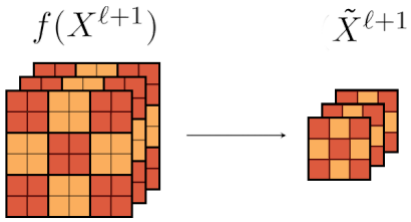
$$\frac{1}{K}\sum_{p,q}|p\rangle|q\rangle \mapsto \frac{1}{K}\sum_{p,q}|p\rangle|q\rangle|\overline{P}_{pq}\rangle|g_{pq}\rangle$$

   - **Non-Linearity**:

$$\frac{1}{K}\sum_{p,q}|p\rangle|q\rangle|f(\overline{Y}^{\ell+1}_{p,q})\rangle|g_{pq}\rangle$$

   - **Quantum Sampling**:

$$\frac{1}{K}\sum_{p,q}\alpha'_{pq}|p\rangle|q\rangle|f(\overline{Y}^{\ell+1}_{pq})\rangle|g_{pq}\rangle$$

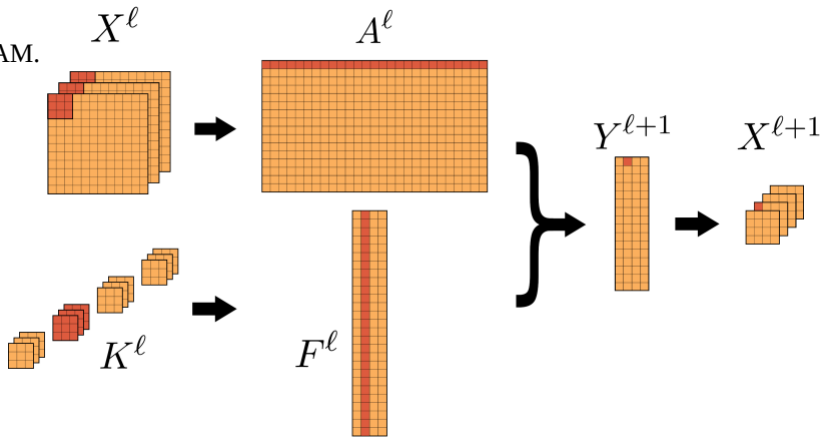



   - **QRAM Update and Pooling**:

     Update QRAM with $A^{\ell+1}$ and apply pooling.

2. **Backward Pass (Quantum Backpropagation)**:
   - **Modify the Gradient:**

     Set to 0 some values of $\frac{\partial \mathcal{L}}{\partial Y^{\ell+1}}$ in QRAM.




   - **Matrix-Matrix Multiplications**:

$$\begin{cases} (A^{\ell})^T \cdot \frac{\partial L}{\partial Y^{\ell+1}} \\ \frac{\partial \mathcal{L}}{\partial Y^{\ell+1}} \cdot (F^{\ell})^T \end{cases}$$

   - **Tomography**:

     Estimate each entry of $\frac{\partial \mathcal{L}}{\partial F^{\ell}}$ and $\frac{\partial \mathcal{L}}{\partial Y^{\ell}}$.

   - **Gradient Descent**:

$$F^{\ell}_{s,q} \leftarrow F^{\ell}_{s,q} - \lambda\left(\frac{\partial \mathcal{L}}{\partial F^{\ell}_{s,q}} \pm 2\delta\left\|\frac{\partial \mathcal{L}}{\partial F^{\ell}}\right\|_2\right)$$

3. **Output**: Updated data matrices $A^{\ell+1}$ and kernel matrices $F^{\ell}$.

## Appendix (A-6): Quantum Language Model with QNLP

Quantum Natural Language Processing (QNLP) is an emerging field that combines the principles of quantum computing with the challenges and opportunities presented by natural language processing (NLP). According to the schematic represented in Figure 9, the difference between the classical NLP and QNLP workflows is illuminated. For instance, the string diagram representation of a sentence along with the quantum circuit representation of the corresponding sentence is provided in Figure 9.

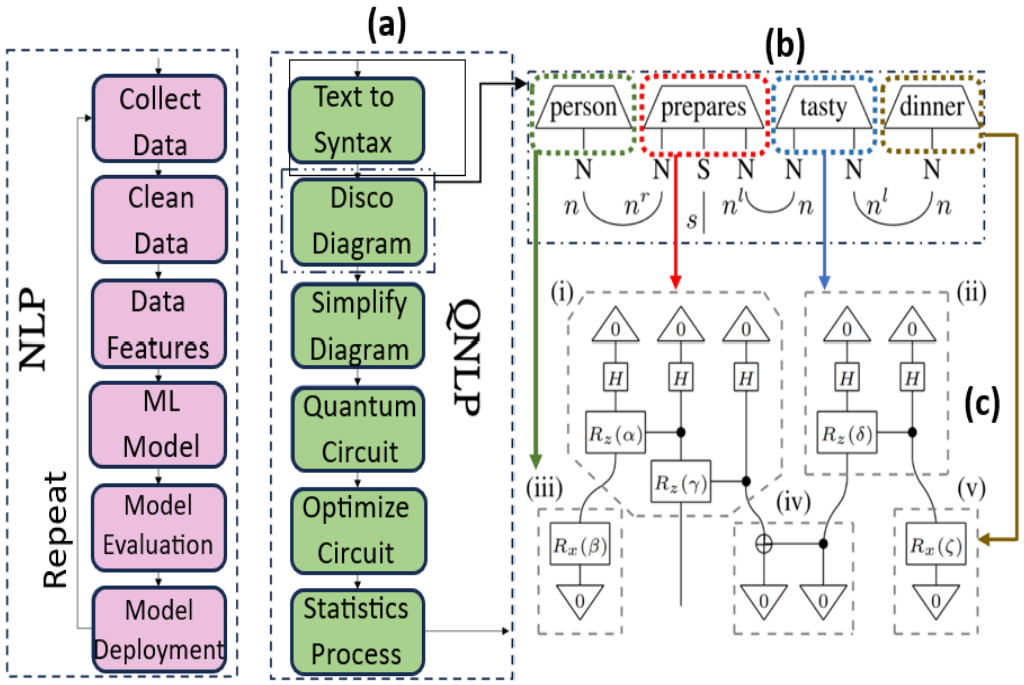



*Figure 9*. (a) QNLP workflow, (b) DisCoCat string diagram, (c) Quantum circuit representation of the sentence in boxes (iii), (i), (ii), (iv), and (v).

## Appendix (B): Dataset Type and Dataset Generation Procedure

The Table 1 provides a comprehensive overview of various datasets used in the study, detailing their size, type, and method of generation. The datasets vary in terms of the number of expressions (Exp. #NUM), objects (Obj. #NUM), scenes (Sce. #NUM), and vocabulary (Voc. #NUM), as well as their application context (indoor or outdoor, and types of objects). The methods of generation include manual and automated techniques, with visual data presented in either 2D or 3D formats. The Quantum3DVG model dataset is applied along this study.

*Table 1.* Utilized dataset details in this study.

| Dataset | Dataset Features | | | | Dataset Type | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Exp. #NUM | Obj. #NUM | Sce. #NUM | Voc. #NUM | Location | | Method of Generation | | Label |
| | | | | | Spa. | Obj. | Txt. | Vis. | 2D/3D |
| SUN-Spot | 7990 | 3245 | 1948 | 2690 | Indoor | Furniture | Manu. | RGB-D | 2D |
| REVERIE | 21702 | 4140 | 90 | 1600 | Indoor | Furniture | Manu. | PC. | 2D |
| ScanRefer | 51583 | 11046 | 704 | 4197 | Indoor | Furniture | Manu. | PC. | 3D |
| Sr3d | 83572 | 8863 | 1273 | 196 | Indoor | Furniture | Temp. | PC. | 3D |
| Nr3d | 41503 | 5879 | 642 | 6951 | Indoor | Furniture | Manu. | PC. | 3D |
| SUNRefer | 38495 | 7699 | 7699 | 5279 | Indoor | Furniture | Manu. | RGB-D | 3D |
| STRefer | 5458 | 3581 | 662 | - | Outdoor | Human | Manu. | PC. & RGB | 3D |
| LifeRefer | 25380 | 11864 | 3172 | - | In/Outdoor | Human | Manu. | PC. & RGB | 3D |
| **Quantum3DVG** | **41140** | **8228** | **2025** | **5271** | **Outdoor** | **Human+ Veh.** | **Manu.+ChatGPT+ Gemini+Perplexity** | **RGB** | **2D/3D** |

The schematic in Figure 10 outlines the process for generating a dataset to train the Quantum 3D Visual Grounding (Q3DVG) model. It starts with categorizing images based on object type (Human + Vehicle) and distance (Far, Medium, Near). Key features such as dimensions, color, direction, occlusion, location, distance, azimuthal angle, and spatial description are extracted from these images. Advanced AI tools like ChatGPT, Gemini, and Perplexity are used to generate detailed text descriptions of the scenes. A verification process ensures the accuracy and quality of the data. This integrated workflow aims to create a robust dataset to train the Q3DVG model for precise 3D visual grounding.

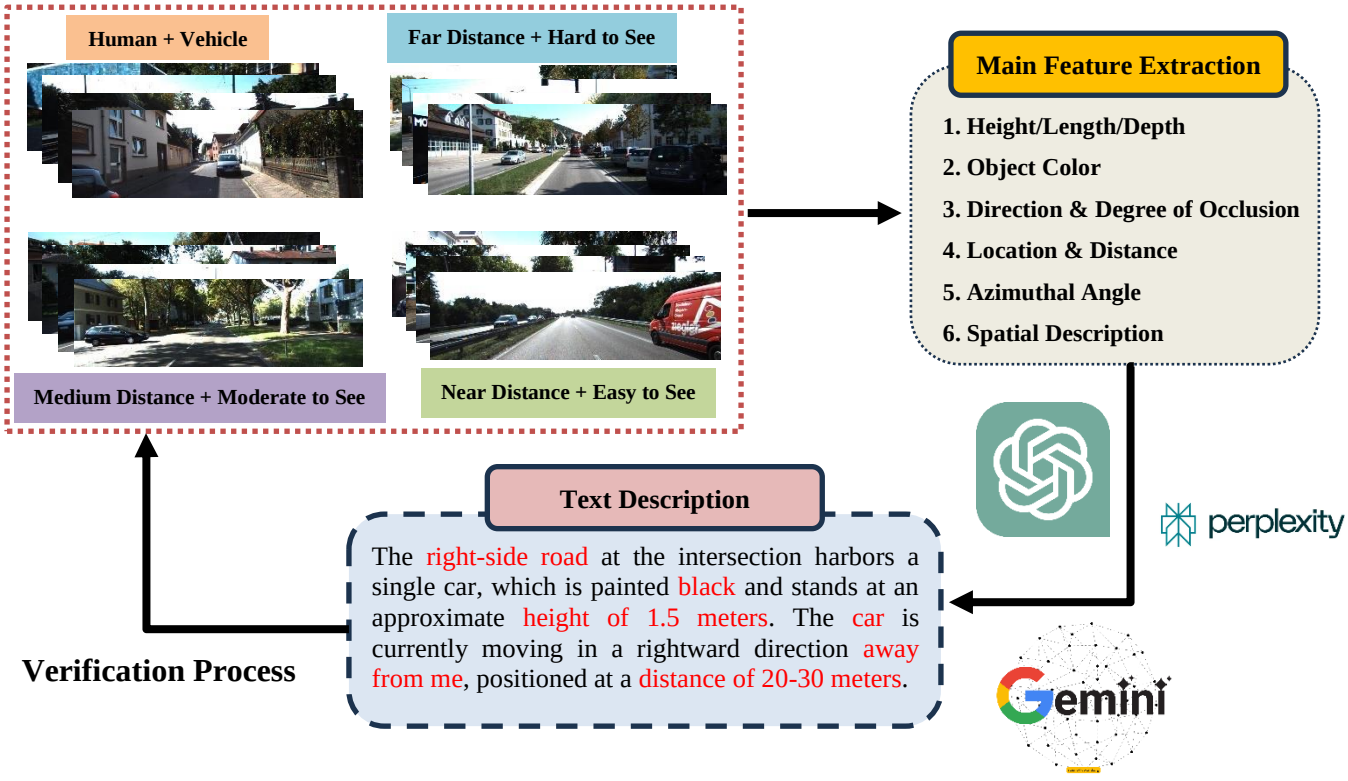



*Figure 10.* The process of dataset generation for training the Quantum 3D Visual Grounding model.

## Appendix (C): Supplementary Quantitative Results

The Table 2 presents a comparison of quantum and classical models on various types of datasets, evaluated based on their accuracy (ACC) at thresholds 0.25 and 0.5. The models, Mono3DVG-TR and Quantum3DVG, are assessed under three categories: “Near Distance + Easy to See”, “Medium Distance + Moderate to See”, and “Far Distance + Hard to See”. In the “Near Distance + Easy to See” category, Quantum3DVG outperforms Mono3DVG-TR at ACC @0.25 (68.23 vs. 64.74) and shows relatively equal performance at ACC @0.5 (54.33). For “Medium Distance + Moderate to See”, Quantum3DVG also outperforms Mono3DVG-TR at both ACC @0.25 (77.12 vs. 75.44) and ACC @0.5 (59.96 vs. 55.48). In the “Far Distance + Hard to See” category, Quantum3DVG slightly outperforms Mono3DVG-TR at ACC @0.25 (46.87 vs. 45.07) and outperforms it at ACC @0.5 (18.41 vs. 15.35). Overall, the Quantum3DVG model demonstrates better or equal performance across all categories and thresholds compared to the Mono3DVG-TR model, a classical 3D visual grounding model, especially in challenging conditions.

*Table 2.* Evaluation of quantum and classical models on various types of datasets.

| **Baseline Model** | **Near Distance + Easy to See** | | **Medium Distance + Moderate to See** | | **Far Distance + Hard to See** | |
|---|---|---|---|---|---|---|
| | ACC @0.25 | ACC @0.5 | ACC @0.25 | ACC @0.5 | ACC @0.25 | ACC @0.5 |
| Mono3DVG-TR | 64.74/72.36 | 53.49/51.80 | 75.44/69.23 | 55.48/48.66 | 45.07/49.01 | 15.35/29.91 |
| Quantum3DVG | **68.23/72.36** | **54.33/51.80** | **77.12/69.23** | **59.96/48.66** | **46.87/49.01** | **18.41/29.91** |

The Table 3 in the ablation study evaluates the performance of a multimodal system that includes both vision and depth tasks, with components consisting of Encoders, Decoders, and Adapters, each of which can be implemented in either a classical or quantum version. The study aims to determine how the inclusion of quantum components affects the system's accuracy, measured by ACC @0.25 and ACC @0.5. For the Classical Version with a Classical Decoder, the ACC @0.25 values start at 47.31 with no quantum components, improve to 60.21 with the addition of a Quantum Vision Encoder, further increase to 61.98 with a Quantum Vision Encoder and Quantum Vision Adapter, and reach the highest value of 64.36 with the addition of a Quantum Depth Adapter. Similarly, the ACC @0.5 values progress from 24.38, to 38.52, 40.12, and finally 44.25 for the same respective configurations.

For the Quantum Version with a Quantum Decoder, the results show improved performance compared to the Classical Decoder. The ACC @0.25 values are 49.55 within quantum components, 62.17 with a Quantum Vision Encoder, 64.18 with a Quantum Vision Encoder and Quantum Vision Adapter, and 67.10 with the inclusion of a Quantum Depth Adapter. The ACC @0.5 values follow the same pattern: 25.88, 39.92, 42.22, and 46.34.

In summary, the Table 3 demonstrates that the performance metrics (ACC @0.25 and ACC @0.5) improve progressively as quantum components are incorporated into the system. The highest performance is achieved when both quantum vision and depth adapters are included. These results are consistent for both classical and quantum decoders, with the Quantum Decoder showing higher performance gains compared to the Classical Decoder. This highlights the potential benefits of integrating quantum technology into multimodal systems for vision and depth tasks.

*Table 3.* The ablation study of quantum and classical Encoder, Decoder, and Adapter components.

| **Decoder** | | **Quantum/Classical Encoder** | | **Quantum/Classical Adapter** | | **Classical Version** | | **Quantum Version** | |
|---|---|---|---|---|---|---|---|---|---|
| **Classical Decoder** | **Quantum Decoder** | Vision | Depth | Vision | Depth | ACC @0.25 | ACC @0.5 | ACC @0.25 | ACC @0.5 |
| ✓ | ✓ | | | | | 47.31 | 24.38 | **49.55** | **25.88** |
| ✓ | ✓ | ✓ | ✓ | | | 60.21 | 38.52 | **62.17** | **39.92** |
| ✓ | ✓ | ✓ | ✓ | ✓ | | 61.98 | 40.12 | **64.18** | **42.22** |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 64.36 | 44.25 | **67.10** | **46.34** |

As can be observed in Table 4, it presents a comprehensive analysis of dataset statistics for Q3DVG and C3DVG, categorized by object location (Near Distance, Medium Distance, Far Distance) and quality of perceptions (Easy to See, Moderate to See, Hard to See). The data is divided into three model evaluation metrics: training, validation, and test sets, along with overall totals. In terms of object location, the distribution is relatively balanced between the datasets across all distances. For instance, Medium Distance has the highest representation with both Q3DVG and C3DVG comprising 43.25% of the total data, followed by Near Distance at 26.49%, and Far Distance at 30.25%. This indicates a focus on medium-range object detection in the datasets. Regarding the quality of perceptions, the "Easy to See" category contains the most data, with each dataset (Q3DVG and C3DVG) comprising approximately 45.92% of the total instances. This is followed by the "Hard to See" category, with each dataset representing around 29.21%, and the "Moderate to See" category with each dataset at 24.88%. This suggests a gradual decrease in data quantity as visibility decreases.

*Table 4.* Dataset statistics for Q3DVG and C3DVG at different object locations and quality of perceptions.

| **Model Evaluation** | **Near Distance** | | **Medium Distance** | | **Far Distance** | | **Overall** |
|---|---|---|---|---|---|---|---|
| | Q3DVG | C3DVG | Q3DVG | C3DVG | Q3DVG | C3DVG | Q3DVG |
| Train | **3,612 (22.45%)** | 7,805 (26.03%) | **7,355 (45.71%)** | 12,815 (42.73%) | **5,121 (31.84%)** | 9,370 (63.44%) | 16,088 |
| Validation | **2,432 (21.56%)** | 1,575 (27.46%) | **4,222 (37.44%)** | 2,525 (44.03%) | **4,621 (41%)** | 1,635 (28.51%) | 11,275 |
| Test | **4,856 (35.24%)** | 1,520 (28.07%) | **6,218 (45.13%)** | 2,455 (45.34%) | **2,703 (19.63%)** | 1,440 (26.59%) | 13,777 |
| Total | **10,900 (26.49%)** | 10,900 (26.49%) | **17,795 (43.25%)** | 17,795 (43.25%) | **12,445 (30.25%)** | 12,445 (30.25%) | 41,140 |
| **Model Evaluation** | **Easy to See** | | **Moderate to See** | | **Hard to See** | | **Overall** |
| | Q3DVG | C3DVG | Q3DVG | C3DVG | Q3DVG | C3DVG | Q3DVG |
| Train | **6,952 (45.28%)** | 13,855 (46.20%) | **3,861 (25.14%)** | 7,425 (24.76%) | **4,540 (29.58%)** | 8,710 (29.04%) | 15,353 |
| Validation | **4,991 (44.65%)** | 2,705 (47.17%) | **3,200 (28.63%)** | 1,390 (24.24%) | **2,986 (26.72%)** | 1,640 (28.60%) | 11,177 |
| Test | **6,947 (47.54%)** | 2,330 (43.03%) | **3,174 (21.72%)** | 1,420 (26.22%) | **4,489 (30.74%)** | 1,665 (30.75%) | 14, 610 |
| Total | **18,890 (45.92%)** | 18,890 (45.92%) | **10,235 (24.88%)** | 10,235 (24.88%) | **12,015 (29.21%)** | 12,015 (29.21%) | 41,140 |

*Table 5.* Ablation study of QVE, QDE, and QD baseline models at different number of layers for Q3DVG and C3DVG.

| **Baseline Model** | **Number of Layer** | | **ACC @0.25** | | **ACC @0.5** | | **Number of Parameters** | |
|---|---|---|---|---|---|---|---|---|
| | Q3DVG | C3DVG | Q3DVG | C3DVG | Q3DVG | C3DVG | Q3DVG | C3DVG |
| QVE | K = 2 | K = 2 | 63.55 | 62.16 | 44.65 | 43.05 | 185.21M | 118.99M |
| | K = 3 | K = 3 | 65.17 | 64.36 | 45.17 | 44.25 | 227.33M | 119.35M |
| | K = 4 | K = 4 | 64.56 | 63.74 | 44.23 | 42.50 | 257.11M | 119.71M |
| QDE | F = 1 | F = 1 | 65.77 | 64.36 | 45.57 | 44.25 | 189.88M | 119.35M |
| | F = 2 | F = 2 | 62.95 | 61.26 | 42.59 | 41.75 | 232.96M | 119.48M |
| | F = 3 | F = 3 | 62.04 | 60.18 | 40.22 | 38.80 | 255.10M | 119.61M |
| QD | S = 1 | S = 1 | 66.16 | 64.36 | 46.36 | 44.25 | 190.21M | 119.35M |
| | S = 2 | S = 2 | 63.91 | 62.84 | 44.96 | 43.52 | 263.87M | 120.25M |
| | S = 3 | S = 3 | 63.27 | 62.34 | 43.11 | 40.80 | 273.44M | 121.16M |
| | S = 4 | S = 4 | 62.33 | 60.38 | 40.13 | 38.56 | 289.12M | 122.06M |
| | S = 5 | S = 5 | 58.01 | 56.14 | 35.71 | 34.28 | 296.76M | 122.97M |

The notable criterion is the smaller number of training datasets for Q3DVG compared to C3DVG as it adjusts the computational time despite of higher number of parameters associated with Q3DVG in comparison with the C3DVG. Despite of a smaller number of training dataset and higher number of datasets for validation and test regarding the Q3DVG model, the quantum-equipped model accuracy still surpasses the threshold (accuracy) for its classical counterpart. It suggests that the quantum model is more trainable data-reluctant rather than the classical counterparts.

Performing further ablation study, the Table 5 presents a comparing investigation on the performance of three baseline models—QVE, QDE, and QD—across varying numbers of layers on two main quantum and classical model versions: Q3DVG and C3DVG. The key performance metrics are accuracy at thresholds of 0.25 and 0.5 (ACC @0.25 and ACC @0.5), along with the number of parameters required for each configuration.

For the QVE model, as the number of layers (K) increases from 2 to 3, the ACC @0.25 improves for both Q3DVG and C3DVG. However, with a further increase to 4 layers, there is a slight decrease in Q3DVG and a slight increase in C3DVG. For ACC @0.5, performance remains relatively stable across different layer configurations, with only minor fluctuations. The number of parameters increases significantly with additional layers. For instance, moving from 2 to 4 layers results in a parameter increase from 185.21M to 257.11M for Q3DVG, indicating a higher computational cost without substantial gains in accuracy.

The QDE model shows a decline in accuracy as the number of layers (F) increases. The highest accuracy at ACC @0.25 is achieved with a single layer, and performance drops as more layers are added. A similar trend is observed for ACC @0.5, where adding layers leads to a consistent decrease in performance. The number of parameters also increases with more layers, similar to the QVE model. However, given the decreasing accuracy, this suggests diminishing returns with additional layers.

The QD model demonstrates the highest accuracy at ACC @0.25 with a single layer and experiences a steady decline as more layers are added. This pattern is consistent across both Q3DVG and C3DVG datasets. For ACC @0.5, the performance also decreases with additional layers, but the decline is more pronounced compared to ACC @0.25. The QD model requires the most parameters among the three models as layers increase. For example, moving from 1 to 5 layers results in a parameter increase from 190.21M to 296.76M for Q3DVG. Despite the increase in parameters, the accuracy gains are not evident, highlighting a potential inefficiency in the model's architecture for deeper configurations.

Key observations include the importance of balancing model complexity with performance. The QD model shows that a simpler configuration (fewer layers) is more effective in terms of accuracy. Adding more layers results in higher computational costs without improving performance, suggesting overfitting or inefficiencies in deeper architectures. The QVE model strikes a balance with 3 layers, offering a stable performance across different accuracy thresholds while managing parameter efficiency better than deeper configurations. The optimal configuration for QVE seems to be with 3 layers, providing a good trade-off between accuracy and the number of parameters. The QDE model performs best with a single layer, indicating that simpler models might be more effective for certain tasks. The QD model, although initially high performing with a single layer, does not scale well with added complexity.

## Appendix (D): Supplementary Qualitative Results

As shown in Figure 12, the schematic displays qualitative results for comparing classical and quantum models for 3D object detection at varying distances: far, medium, and near. The results are showcased using original images, depth predictions, 3D bounding boxes from classical (C3DVG) and quantum (Q3DVG) models, and corresponding textual queries.

For the far distance scenario, the original image shows a street scene with a car highlighted in a red bounding box, located at a considerable distance from the viewer. The depth map displays a grayscale representation where closer objects are lighter, and distant objects are darker. Two 3D bounding boxes are presented (zoom-in mode): the C3DVG in blue, and the Q3DVG in red. The query describes a car about 1.5 meters in height, located approximately 50 meters north of the observer, in the lane ahead. It is the second car in front and is facing away.

In the medium distance scenario, the original image features a different street scene with a car highlighted in an orange bounding box, closer than in the far distance image. The depth map similarly shows the objects' distances with a different grayscale gradient, indicating medium-range objects. Again, two 3D bounding boxes are presented in zoom-in mode: C3DVG in green, and Q3DVG in orange. The query describes a black car, about 1.6 meters high and 3.2 meters long, heading north, 30 meters away. The car is moving away with its back facing the observer.

For the near distance scenario, the original image shows a car much closer to the observer, highlighted in a green bounding box. The depth map shows a clearer distinction of nearby objects, with more pronounced grayscale differentiation. Two 3D bounding boxes are presented: C3DVG in green, and Q3DVG in yellow. The query describes a white car parked on the

right side of the road, about 20 meters away, with its back facing the observer. The car is the second one to the right and positioned around 20 degrees north-northeast.

Comparatively, for the far distance, the classical model's 3D bounding box aligns well with the car's position, while the quantum model provides a more refined and more accurate bounding box. In the medium distance, both models accurately capture the car's dimensions, but the quantum model again appears to offer a more precise bounding box, especially in terms of depth perception. For the near distance, the proximity to the car allows both models to provide accurate bounding boxes, though the quantum model might offer a slight edge in precision. Overall, the qualitative results and accompanying analysis demonstrate the effectiveness of quantum models (Q3DVG) over classical models (C3DVG) in 3D object detection across different distances. Quantum models show superior accuracy in bounding box placement and depth perception, particularly evident in medium and far distances. This qualitative assessment underscores the potential advantages of integrating quantum computing approaches in visual detection tasks.

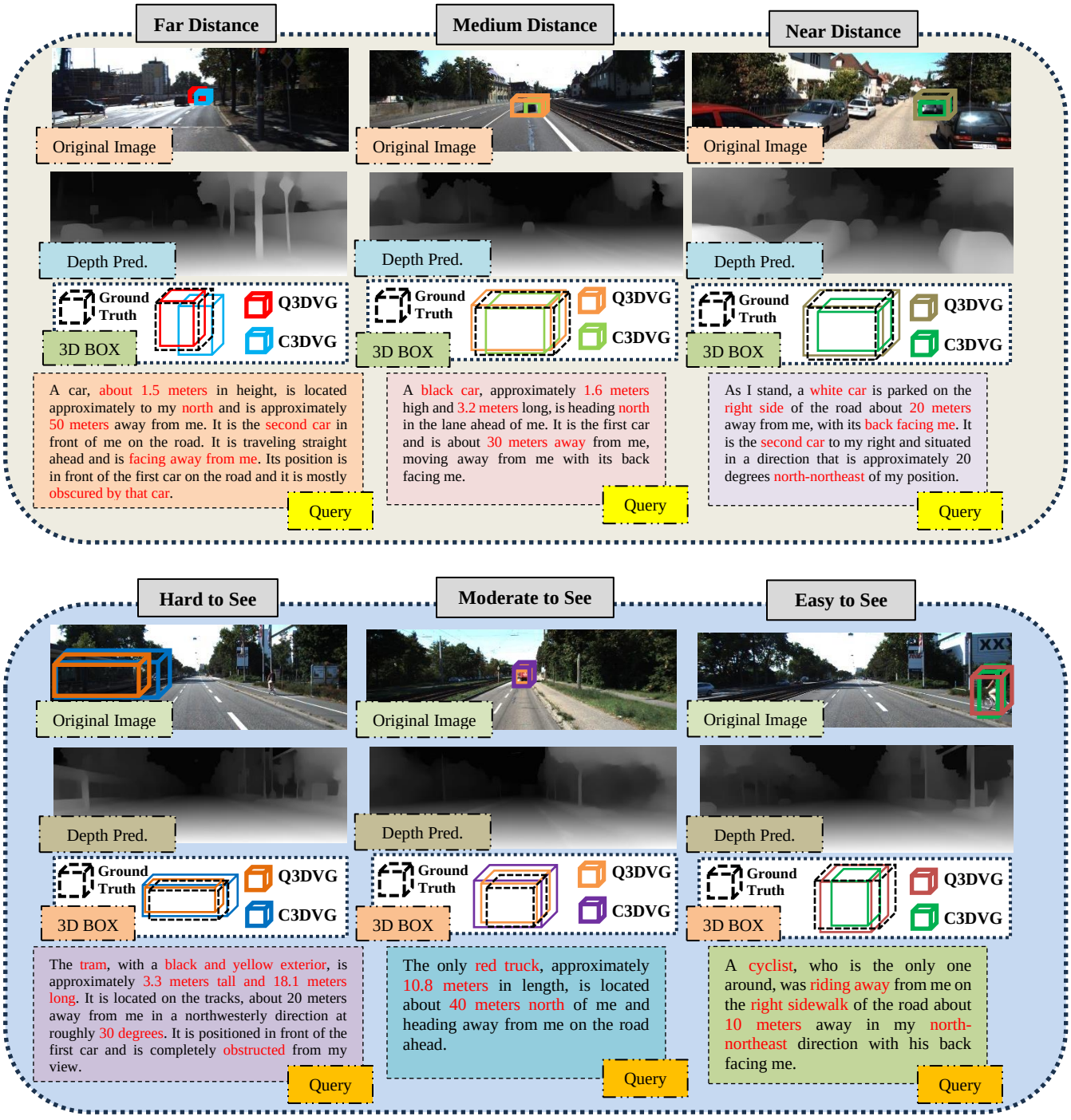



*Figure 12.* Visualization of 3D boxing and depth map prediction for Q3DVG and C3DVG models at different locations and perception views.

Further results illustrate the qualitative outcomes of 3D object detection models, Q3DVG and C3DVG, in scenarios where queries are not directly pointing out clear features of objects. The respected schematic in Figure 13 contains three sections, each with an original image, the corresponding 3D bounding boxes from both models, and textual queries describing the scene. Three scenarios associated with additional qualitative results are mentioned as below:

In the first scenario, the original image depicts a yellow car parked close to the sidewalk. The car is highlighted with both red and blue bounding boxes. The C3DVG model is shown in blue, while the Q3DVG model is shown in red. The query states that the yellow car is parked close to the sidewalk, adjacent to a white fence and green bushes. The car's vibrant yellow color stands out against the more subdued tones of the surrounding vehicles and buildings.

In the second scenario, the original image shows a cyclist on the right side of the road, highlighted in both orange and green bounding boxes. The C3DVG model is shown in orange, and the Q3DVG model is shown in green. The query describes a cyclist wearing a white shirt, riding along the bike path on the right side of the road. The cyclist is moving away from the camera, heading towards the distance where the road and path converge. The cyclist is near a line of pink flowers that border the bike path, adding a splash of color to the scene.

In the third scenario, the original image shows a pedestrian crossing the street, highlighted with both green and yellow bounding boxes. The C3DVG model is shown in yellow, while the Q3DVG model is shown in green. The query mentions a pedestrian crossing the street from right to left. The pedestrian is wearing dark clothing, which contrasts against the bright, sunlit street. The pedestrian is positioned at the center of a zebra crossing, highlighted by white lines on the road. The surrounding environment includes vehicles waiting at traffic lights, a cyclist preparing to cross, and greenery that frames the urban scene.

In each scenario, the models were tested with queries that provided contextual rather than specific object details. This ensures that the models are capable of accurately detecting and describing objects even when the queries are more general and do not highlight clear features. For the first scenario, both models correctly identify and box the yellow car, with the quantum model potentially offering more refined results. In the second scenario, the cyclist is inaccurately detected and boxed by C3DVG model, while the Q3DVG model showing a precise fit. In the third scenario, the pedestrian is not well-detected by C3DVG model, whereas the Q3DVG model is once again demonstrating a considerable edge in accuracy. Overall, this qualitative assessment showcases the robustness of Q3DVG model in handling queries with less specific details considering that the quantum model consistently provides a higher level of precision.

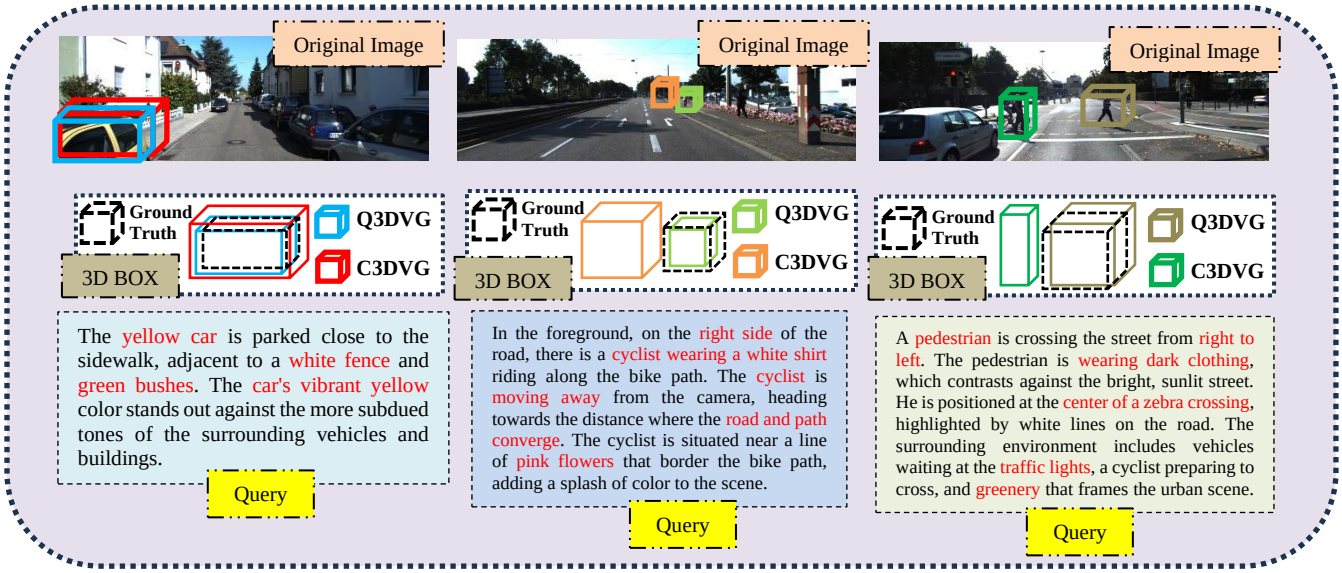



*Figure 13.* Further testaments of 3D boxing for Q3DVG and C3DVG models using types of uncommon query.