

Q-SAGE: Calibrated LLM Reasoning over QCNN-Derived Scene Graphs for Safety-Aware Autonomous Mobility

Emily Jimin Roh[†], Hyojun Ahn[†], Joongheon Kim[†], Soyi Jung[‡] and Soohyun Park[§]

[†]Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea

[‡]Department of Electrical and Computer Engineering, Ajou University, Seoul, Republic of Korea

[§]Division of Computer Science, Sookmyung Women's University, Seoul, Republic of Korea

{emilyjroh, hyojun, joongheon}@korea.ac.kr, sjung@ajou.ac.kr, soohyun.park@sookmyung.ac.kr

Abstract—Quantum convolutional neural networks (QCNNs) provide a promising perception backbone for resource-constrained autonomous mobility systems, but detector outputs alone are insufficient for high-level safety reasoning. In this paper, we propose Q-SAGE, a QCNN-guided safety-aware graph reasoning framework with LLM calibration. Q-SAGE converts QCNN object detection outputs into structured scene graphs and combines language-based reasoning with calibrated safety validation to estimate scene-level risk and high-level maneuver decisions. We evaluate Q-SAGE on a held-out stress-test split using risk reasoning, action reasoning, safety-trigger detection, and composite performance metrics. Compared with Confidence Baseline, Object Count Baseline, LLM-only Reasoning, QCNN+LLM Reasoning, and QCNN Semantic Reasoning, Q-SAGE achieves the best primary reasoning performance. It obtains 0.800 risk accuracy, 0.795 risk macro-F1, 0.753 action accuracy, 0.612 action macro-F1, and the highest composite score of 0.772. In particular, Q-SAGE substantially improves HIGH-risk recognition, achieving 0.77 precision, 0.96 recall, and 0.85 F1-score, while QCNN Semantic Reasoning fails to detect the HIGH class. These results show that QCNN-derived scene graphs combined with calibrated LLM reasoning improve risk severity estimation and safety-aware decision consistency for autonomous mobility.

Index Terms—Quantum Convolutional Neural Network, Large Language Model, Autonomous Mobility

I. INTRODUCTION

Autonomous mobility systems require perception modules that are accurate, efficient, and capable of supporting safety-aware decision making [1]. Object detection is a fundamental component of such systems, since downstream planning and control modules depend on reliable information about vehicles, pedestrians, cyclists, and other surrounding objects. However, deploying perception models on resource-constrained mobility platforms remains challenging due to limited computation, strict latency requirements, and the need for robust operation in dynamic environments [2].

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2026-RS-2024-00436887) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation); and also by IITP grant funded by MSIT (RS-2024-00439803, SW Star Lab). Joongheon Kim, Soyi Jung, Soohyun Park are corresponding authors.

Quantum convolutional neural networks (QCNNs) have recently attracted attention as a promising perception backbone for efficient feature extraction [3]–[5]. In an autonomous mobility scenario, a QCNN-based object detector can provide compact perception outputs such as object classes, bounding boxes, confidence scores, and runtime information [6]. These outputs are useful for real-time perception, but they do not directly provide high-level safety reasoning. For example, the risk of a detected object depends not only on its confidence score, but also on its category, spatial location, approximate distance, ego-vehicle state, and system latency. A pedestrian in the front-center region may require conservative action even when the detector confidence is moderate, whereas a low-confidence object outside the ego trajectory may only require monitoring.

This gap motivates the use of a reasoning layer that can convert low-level detector outputs into structured safety decisions. Large language models (LLMs) provide a natural interface for this purpose because they can reason over symbolic or semi-structured scene descriptions and generate interpretable high-level decisions [7], [8]. Nevertheless, directly relying on an LLM for autonomous decision making is risky, since language models may be poorly calibrated for safety-critical scenarios and may produce inconsistent outputs [9]. Therefore, an effective QCNN–LLM integration should preserve the efficiency of QCNN-based perception while constraining language-based reasoning through structured inputs and safety validation.

In this paper, we propose **Q-SAGE**, a QCNN-guided safety-aware graph reasoning framework with LLM Calibration. Q-SAGE first converts QCNN object detection outputs into a structured scene graph containing object class, spatial region, distance proxy, confidence group, object criticality, ego-state information, and runtime latency. An LLM-based reasoning module then predicts scene-level risk and a high-level maneuver recommendation. Finally, a calibrated safety validator adjusts the LLM output using deterministic safety constraints derived from the QCNN scene graph. This design allows Q-SAGE to combine the efficiency of QCNN perception, the semantic reasoning capability of LLMs, and the reliability of

rule-based safety validation.

We evaluate Q-SAGE on a held-out stress-test split using multiple performance metrics, including risk accuracy, risk macro-F1, action accuracy, action macro-F1, safety-trigger recall, safety-trigger precision, and composite score. Compared with Confidence Baseline, Object Count Baseline, LLM-only Reasoning, QCNN+LLM Reasoning, and QCNN Semantic Reasoning, Q-SAGE achieves the best primary reasoning performance. It obtains 0.800 risk accuracy, 0.795 risk macro-F1, 0.753 action accuracy, 0.612 action macro-F1, 0.960 safety recall, and the highest composite score of 0.772. In particular, Q-SAGE substantially improves HIGH-risk recognition, achieving 0.77 precision, 0.96 recall, and 0.85 F1-score, while QCNN Semantic Reasoning fails to detect the HIGH class. These results suggest that detector-only reasoning can identify some safety-relevant scenes but may underestimate risk severity, whereas Q-SAGE improves calibrated safety-aware decision consistency.

The main contributions are summarized as follows:

- We propose Q-SAGE, a QCNN-guided safety-aware graph reasoning framework that integrates QCNN-based object detection with LLM-based high-level decision reasoning.
- We introduce a scene graph representation that converts QCNN detector outputs into structured reasoning inputs containing object, spatial, confidence, ego-state, and latency information.
- We design a calibrated safety validator that constrains LLM outputs using deterministic safety rules derived from QCNN scene graphs.
- We provide a comparative performance evaluation against Confidence Baseline, Object Count Baseline, LLM-only Reasoning, QCNN+LLM Reasoning, and QCNN Semantic Reasoning.
- We show that Q-SAGE improves risk severity calibration, especially for HIGH-risk recognition, while achieving the best composite score among the evaluated methods.

II. QCNN-GUIDED SAFETY-AWARE GRAPH REASONING WITH LLM CALIBRATION

This section describes the proposed **Q-SAGE** framework, which integrates QCNN-based perception, structured scene graph construction, LLM-based safety reasoning, and calibrated safety validation for autonomous mobility. The main objective of Q-SAGE is to bridge the gap between low-level object detection and high-level safety-aware decision support. While a QCNN detector provides compact perception outputs such as object classes, bounding boxes, confidence scores, and runtime telemetry, these outputs alone do not directly indicate driving risk. Q-SAGE therefore transforms QCNN detections into a structured symbolic representation and uses an LLM reasoning layer to infer scene-level risk, high-level maneuver recommendations, safety-trigger decisions, and explanations.

Algorithm 1 Q-SAGE: QCNN-guided Safety-Aware Graph Reasoning with LLM Calibration

Require: Input frame I_t , ego state $\mathcal{E}_t$, calibrated thresholds θ
Ensure: Final decision $y_t = \{z_t, a_t, u_t, e_t\}$

```

1:  $\mathcal{D}_t, \mathcal{R}_t \leftarrow \text{QCNNDetector}(I_t)$   $\triangleright$  Object detections and
   runtime telemetry
2:  $\mathcal{G}_t \leftarrow \Phi(\mathcal{D}_t, \mathcal{E}_t, \mathcal{R}_t)$   $\triangleright$  Scene graph construction
3: for each object  $(c_i, b_i, s_i) \in \mathcal{D}_t$  do
4:   Compute spatial region  $r_i$  from bounding box center
5:   Estimate distance proxy  $d_i$  from bounding box scale
6:   Assign criticality  $\kappa_i$  based on object class
7:   Add object node  $o_i = \{c_i, r_i, d_i, s_i, \kappa_i\}$  to  $\mathcal{G}_t$ 
8: end for
9:  $\hat{y}_t^{\text{LLM}} \leftarrow \text{LLMReasoner}(\mathcal{G}_t)$   $\triangleright$  Preliminary risk, action,
   trigger, and explanation
10: Initialize  $y_t \leftarrow \hat{y}_t^{\text{LLM}}$ 
11: if  $\ell_t > \tau_\ell$  then
12:    $z_t \leftarrow \text{HIGH}$ 
13:    $a_t \leftarrow \text{SAFE\_MODE}$ 
14: else if  $\exists i : \kappa_i = \text{critical} \wedge d_i = \text{near}$  then
15:    $z_t \leftarrow \text{HIGH}$ 
16:    $a_t \leftarrow \text{SLOW\_DOWN}$ 
17: else if  $\exists i : \kappa_i = \text{critical} \wedge r_i = \text{front-center} \wedge$ 
    $v_t > \tau_v$  then
18:    $z_t \leftarrow \text{HIGH}$ 
19:    $a_t \leftarrow \text{SLOW\_DOWN}$ 
20: else if  $\exists i : \kappa_i = \text{critical} \wedge s_i < \tau_s$  then
21:    $z_t \leftarrow \max(z_t, \text{MEDIUM})$ 
22:    $a_t \leftarrow \text{SENSOR\_RECHECK}$ 
23: else if  $N_t \geq \tau_N$  then
24:    $z_t \leftarrow \max(z_t, \text{MEDIUM})$ 
25:    $a_t \leftarrow \text{KEEP\_LANE\_AND\_MONITOR}$ 
26: end if
27:  $u_t \leftarrow \mathbb{I}[z_t \in \{\text{MEDIUM}, \text{HIGH}\}]$ 
28: Update explanation  $e_t$  with validator rationale
29: return  $y_t = \{z_t, a_t, u_t, e_t\}$ 

```

A. QCNN-based Perception Layer

The perception layer of Q-SAGE is built on a QCNN-based object detector. In this work, the QCNN serves as an efficient perception backbone for extracting object-level information from an input driving frame. Conceptually, the QCNN follows a hybrid perception structure in which visual features are encoded into a quantum-inspired or quantum-assisted feature space, processed through convolution-like quantum operations, and then decoded into classical detection outputs. This design is motivated by the potential of QCNNs to provide compact feature representations for resource-constrained autonomous mobility platforms.

Given an input frame I_t at time t , the QCNN detector produces a set of object detections:

$$\mathcal{D}_t = \{(c_i, b_i, s_i)\}_{i=1}^{N_t}, \quad (1)$$

where c_i is the predicted object class, b_i is the bounding box,

s_i is the confidence score, and N_t is the number of detected objects. The detector also provides runtime telemetry:

$$\mathcal{R}_t = \{\ell_t, f_t\}, \quad (2)$$

where ℓ_t denotes inference latency and f_t denotes the estimated frame rate. These telemetry variables are important because safety reasoning in autonomous mobility depends not only on what the detector observes, but also on whether the perception system is operating within real-time constraints.

The QCNN output remains a perception-level representation. It indicates which objects are detected and where they are located, but it does not explicitly encode whether the scene is safe, uncertain, or safety-critical. For example, a moderate-confidence pedestrian in the front-center region may be more critical than a high-confidence vehicle far from the ego trajectory. Therefore, Q-SAGE does not directly map QCNN confidence to driving risk. Instead, it converts the detector output into a structured scene graph for higher-level reasoning.

B. Scene Graph Construction

The scene graph construction module converts raw QCNN detections into a symbolic representation suitable for LLM-based reasoning. Q-SAGE maps the QCNN detections, ego-state information, and runtime telemetry into a scene graph:

$$\mathcal{G}_t = \Phi(\mathcal{D}_t, \mathcal{E}_t, \mathcal{R}_t), \quad (3)$$

where $\mathcal{E}_t$ denotes ego-state information, such as ego speed or road context, and $\Phi(\cdot)$ is a deterministic transformation function.

Each detected object is represented as an object node:

$$o_i = \{c_i, r_i, d_i, s_i, \kappa_i\}, \quad (4)$$

where r_i is the spatial region, d_i is the distance proxy, and κ_i is the object criticality. The spatial region r_i is computed from the bounding box center and discretized into regions such as front-left, front-center, and front-right. The distance proxy d_i is estimated from the bounding box scale, where larger bounding boxes are treated as closer objects. Although this proxy does not replace metric depth estimation, it provides a lightweight abstraction that is useful when only 2D detection outputs are available.

The criticality attribute κ_i reflects the safety relevance of the object class. Vulnerable road users, such as pedestrians and cyclists, are assigned higher criticality than ordinary vehicles because they typically require more conservative responses in autonomous mobility scenarios. The scene graph also includes scene-level attributes, including the minimum detector confidence, number of detected objects, ego speed, latency, and frame rate. As a result, $\mathcal{G}_t$ provides a structured intermediate representation that combines quantum-assisted perception outputs with classical safety-relevant context.

C. LLM-based Safety Reasoning

The LLM reasoning layer receives the QCNN-derived scene graph rather than the raw image. This design reduces the reasoning burden of the LLM and constrains its input to compact, interpretable, and task-relevant information. The LLM is prompted to reason over object category, spatial position, distance proxy, confidence, object criticality, ego-state information, and runtime telemetry. It then predicts a preliminary decision:

$$\hat{y}_t^{\text{LLM}} = \{\hat{z}_t, \hat{a}_t, \hat{u}_t, \hat{e}_t\}, \quad (5)$$

where $\hat{z}_t \in \{\text{LOW}, \text{MEDIUM}, \text{HIGH}\}$ is the estimated risk level, $\hat{a}_t$ is the recommended high-level maneuver, $\hat{u}_t$ is the safety-trigger decision, and $\hat{e}_t$ is the explanation.

The role of the LLM is not to generate low-level control commands such as steering angle, throttle, or braking force. Instead, it acts as a high-level semantic reasoning module that interprets the structured scene and recommends an abstract maneuver, such as KEEP_LANE, SENSOR_RECHECK, SLOW_DOWN, or SAFE_MODE. This separation is important for safety: Q-SAGE uses the LLM for interpretable supervisory reasoning, while low-level control can remain handled by a conventional planner or safety controller.

The LLM explanation $\hat{e}_t$ also provides interpretability. For example, if the model predicts a HIGH risk level, the explanation may indicate that a critical object is located near the front-center region or that the detector confidence is too low for reliable operation. This makes the reasoning process more transparent than a detector-only heuristic, which typically outputs only a class label or confidence score.

D. Calibrated Safety Validation

Although LLMs provide flexible semantic reasoning, their outputs may be inconsistent or insufficiently conservative in safety-critical scenarios. To address this issue, Q-SAGE applies a calibrated safety validator after the LLM reasoning step. The validator computes the final output:

$$y_t = \Psi(\hat{y}_t^{\text{LLM}}, \mathcal{G}_t; \theta), \quad (6)$$

where $\Psi(\cdot)$ denotes the validator and θ represents calibrated thresholds selected on the validation split. The final decision is defined as

$$y_t = \{z_t, a_t, u_t, e_t\}, \quad (7)$$

where z_t is the calibrated risk level, a_t is the final high-level action, u_t is the safety-trigger decision, and e_t is the final explanation.

The validator is designed to prevent underestimation of safety-critical scenes. For example, if a critical object is detected at a near distance, the risk is escalated to HIGH. If a critical object appears in the front-center region while the ego vehicle is moving at high speed, the action is adjusted to SLOW_DOWN. If a critical object has low detector confidence, the system requests SENSOR_RECHECK. If runtime latency exceeds a calibrated threshold, the system switches to SAFE_MODE. These rules do not replace the LLM; rather, they

TABLE I
PERFORMANCE COMPARISON ON THE HELD-OUT STRESS-TEST SPLIT. Q-SAGE ACHIEVES THE BEST RISK/ACTION REASONING PERFORMANCE.

Method	Risk Acc.	Risk Macro-F1	Action Acc.	Action Macro-F1	Safety Recall	Safety Precision	Composite
Confidence Baseline	0.433	0.302	0.360	0.164	0.248	0.812	0.313
Object Count Baseline	0.520	0.434	0.420	0.233	0.574	0.882	0.427
LLM-only Reasoning	0.507	0.411	0.393	0.214	0.594	0.870	0.410
QCNN+LLM Reasoning	0.667	0.520	0.620	0.431	0.970	0.817	0.604
QCNN Semantic Reasoning	0.700	0.500	0.653	0.427	0.889	0.828	0.616
Q-SAGE (Ours)	0.800	0.795	0.753	0.612	0.960	0.851	0.772

TABLE II
CLASS-WISE RISK CLASSIFICATION RESULTS. Q-SAGE SUBSTANTIALLY IMPROVES HIGH-RISK RECOGNITION COMPARED WITH QCNN SEMANTIC REASONING.

Method	Class	Precision	Recall	F1-score
QCNN Semantic Reasoning	LOW	1.00	0.57	0.73
	MEDIUM	0.63	0.89	0.77
	HIGH	0.00	0.00	0.00
Q-SAGE (Ours)	LOW	1.00	0.58	0.75
	MEDIUM	0.75	0.91	0.82
	HIGH	0.77	0.96	0.85

constrain the LLM output so that the final decision remains conservative under uncertainty.

E. System-level Interpretation

At the system level, Q-SAGE is designed as a supervisory safety reasoning layer placed between QCNN-based perception and downstream motion planning. The QCNN detector provides efficient frame-level perception, the scene graph module organizes the perception output into a structured representation, the LLM performs semantic interpretation, and the calibrated validator enforces safety-aware constraints. The final output can be used by a downstream planner or safety monitor as a high-level decision signal.

This modular design has three practical advantages. First, it preserves the efficiency of QCNN-based perception by keeping the detector as the front-end module. Second, it improves interpretability by converting detections into human-readable scene graphs and natural-language explanations. Third, it improves safety consistency by ensuring that critical conditions, such as low confidence, near vulnerable road users, front-center hazards, and excessive latency, trigger conservative responses. Therefore, Q-SAGE provides a structured and explainable interface between quantum-assisted perception and LLM-based autonomous mobility reasoning.

III. EXPERIMENT RESULTS

Table I summarizes the overall performance comparison on the held-out stress-test split. Q-SAGE achieves the best performance across the primary risk and action reasoning metrics. Specifically, Q-SAGE obtains a risk accuracy of 0.800, risk macro-F1 of 0.795, action accuracy of 0.753, action macro-F1 of 0.612, and the highest composite score of 0.772. Compared with QCNN Semantic Reasoning, Q-SAGE improves risk accuracy from 0.700 to 0.800 and risk

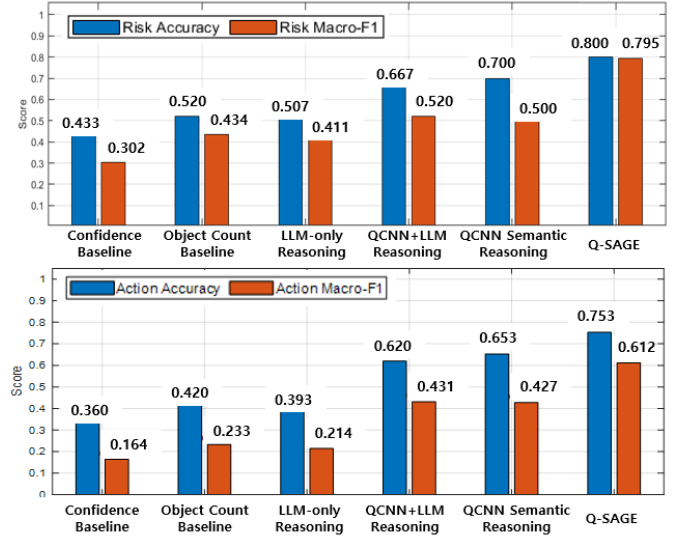


Fig. 1. Risk and action reasoning comparison. Q-SAGE achieves the highest risk and action reasoning performance among the evaluated methods.

macro-F1 from 0.500 to 0.795. This indicates that the proposed combination of QCNN-derived scene graph reasoning, LLM-based semantic reasoning, and calibrated safety validation improves risk severity estimation beyond detector-only semantic reasoning.

The baseline comparison in Table I also shows that simple heuristic methods are insufficient for safety-aware reasoning. The Confidence Baseline achieves the lowest composite score of 0.313 and a low safety recall of 0.248, indicating that confidence scores alone cannot reliably identify safety-critical situations. The Object Count Baseline improves safety precision to 0.882, but its safety recall remains limited at 0.574. This suggests that object count can capture scene complexity to some extent, but it cannot fully represent object criticality, spatial risk, or ego-state-dependent hazards.

The LLM-only and QCNN+LLM baselines further demonstrate the importance of structured QCNN-derived information and calibration. LLM-only Reasoning obtains a composite score of 0.410, showing that language reasoning without structured QCNN scene information is not sufficient for reliable autonomous mobility risk assessment. QCNN+LLM Reasoning improves the composite score to 0.604, but it still underperforms Q-SAGE. This gap indicates that directly

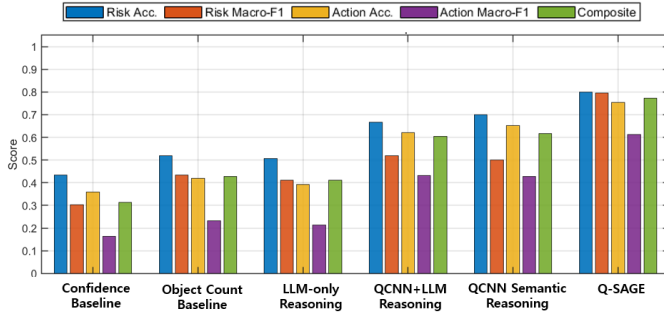


Fig. 2. Metric comparison on the held-out stress-test split. Q-SAGE achieves the highest composite score and strongest primary reasoning performance.

combining QCNN outputs with LLM reasoning is beneficial, but calibrated safety validation is needed to correct uncertain or underestimated decisions.

Table II provides a class-wise analysis of risk classification. The important improvement is observed for the HIGH-risk class. QCNN Semantic Reasoning fails to detect HIGH-risk cases, producing 0.00 precision, 0.00 recall, and 0.00 F1-score for the HIGH class. In contrast, Q-SAGE achieves 0.77 precision, 0.96 recall, and 0.85 F1-score for HIGH-risk cases. This result shows that detector-only semantic reasoning tends to underestimate severe safety-critical scenes, whereas Q-SAGE can more effectively escalate high-risk situations by jointly considering object category, spatial region, distance proxy, confidence, ego-state information, and safety calibration.

Figure 2 visualizes the risk and action reasoning performance across the evaluated methods. The upper plot shows that Q-SAGE achieves the highest risk accuracy and risk macro-F1, while the lower plot shows that Q-SAGE also achieves the highest action accuracy and action macro-F1. The consistent improvement across both plots indicates that Q-SAGE does not merely improve risk classification, but also enhances high-level maneuver recommendation.

Figure 1 compares all primary metrics and the composite score in a single grouped-bar visualization. Q-SAGE achieves the strongest overall profile, with the highest values in risk accuracy, risk macro-F1, action accuracy, action macro-F1, and composite score. Although Object Count Baseline achieves the highest safety precision, its lower safety recall and composite score show that precision alone is insufficient for safety-critical autonomous mobility reasoning. Overall, the results demonstrate that Q-SAGE provides the most balanced and effective safety-aware reasoning performance among the evaluated methods.

IV. CONCLUSION

In this paper, we proposed Q-SAGE, a QCNN-guided safety-aware graph reasoning framework with LLM calibration for autonomous mobility. Q-SAGE converts QCNN object detection outputs into structured scene graphs and combines LLM-based semantic reasoning with calibrated safety validation to estimate scene-level risk and high-level maneuver

decisions. By incorporating object class, spatial region, distance proxy, confidence, ego-state information, and runtime telemetry, Q-SAGE provides a structured interface between efficient QCNN-based perception and safety-aware decision support.

The held-out stress-test evaluation showed that Q-SAGE outperforms Confidence Baseline, Object Count Baseline, LLM-only Reasoning, QCNN+LLM Reasoning, and QCNN Semantic Reasoning on the primary reasoning metrics. Q-SAGE achieved the highest risk accuracy, risk macro-F1, action accuracy, action macro-F1, and composite score. In particular, Q-SAGE substantially improved HIGH-risk recognition, achieving 0.77 precision, 0.96 recall, and 0.85 F1-score, while QCNN Semantic Reasoning failed to detect the HIGH class. These results indicate that detector-only semantic reasoning may underestimate severe safety-critical scenes, whereas Q-SAGE improves risk severity calibration through QCNN-derived scene graphs, LLM-based reasoning, and calibrated safety validation.

Therefore, Q-SAGE provides an interpretable and modular framework for connecting QCNN-based perception with high-level autonomous mobility reasoning. Future work will evaluate Q-SAGE with real QCNN detector outputs on large-scale driving datasets, incorporate multimodal verification through vision-language models, and further study how calibrated safety validation can be optimized for real-time supervisory decision support.

REFERENCES

- [1] E. J. Roh, I. S. Song, S. Kim, and S. Park, "Autonomous mission-oriented unmanned underwater vehicle control using directional policy optimization," *Ocean Engineering*, vol. 320, p. 120242, March 2025.
- [2] H. Baek, W. J. Yun, S. Park, and J. Kim, "Stereoscopic scalable quantum convolutional neural networks," *Neural Networks*, vol. 165, pp. 860–867, August 2023.
- [3] J.-N. Zaech, A. Liniger, M. Danelljan, D. Dai, and L. Van Gool, "Adiabatic quantum computing for multi object tracking," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, June 2022, pp. 8811–8822.
- [4] E. J. Roh, J. Y. Shim, J. Kim, and S. Park, "Hybrid quantum-classical 3D object detection using multi-channel quantum convolutional neural network," *The Journal of Supercomputing*, vol. 81, no. 3, p. 455, February 2025.
- [5] E. J. Roh, H. Baek, D. Kim, and J. Kim, "Fast quantum convolutional neural networks for low-complexity object detection in autonomous driving applications," *IEEE Transactions on Mobile Computing*, vol. 24, no. 2, pp. 1031–1042, February 2025.
- [6] J. H. Bappy and A. K. Roy-Chowdhury, "CNN based region proposals for efficient object detection," in *Proc. of the IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, September 2016, pp. 3658–3662.
- [7] Y. Mao, L. Mathias, R. Hou, A. Almahairi, H. Ma, J. Han, S. Yih, and M. Khabsa, "UniPELT: A unified framework for parameter-efficient language model tuning," in *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, Dublin, Ireland, May 2022.
- [8] E. J. Roh and J. Kim, "Quantum-amplitude embedded adaptation for parameter-efficient fine-tuning in large language models," in *Proc. ACM International Conference on Information and Knowledge Management (CIKM)*, Seoul, Korea, November 2025.
- [9] C. Tao, J. Feng, T. Shen, C. Liu, J. Li, X. Geng, and D. Jiang, "CORE: Cooperative training of retriever-reranker for effective dialogue response selection," in *Proc. Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, Jul. 2023, pp. 3102–3114.