

Visual Element Property Graphs for Bridging the Symbol Description-Recognition Gap

Nicholas Alexander Dehnen

**A Thesis submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of Master of Science**

Graduate Program in Electrical Engineering and Computer Science

**York University
Toronto, Ontario**

April 2025

© Nicholas Alexander Dehnen, 2025

Abstract

This thesis addresses the semantic gap between visual perception and functional significance of symbols used in road vehicles. It presents a novel approach that enables users to identify and understand automotive symbols by describing what they visually perceive, translating visual descriptions into practical implications. A system combining a property graph representation of visual components and semantic relationships with a language model-powered natural language interface is developed. This method explicitly models relationships between visual elements and interpretations, differing from end-to-end vision-language models. Evaluations, using automated metrics and human assessment, demonstrate performance exceeding baseline large language models, with a BERTscore F_1 of 0.765, compared to the best baseline's 0.597. Analysis of visual symbol queries reveals human description tendencies, favoring intuitive analogies and basic shapes. Contributions include a symbol decomposition methodology, an advanced property graph schema, natural language query processing, and evidence supporting structured knowledge representation for symbol description-recognition, applicable beyond automotive interfaces.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Aijun An, and my co-supervisor, Dr. Heidar (Kourosh) Davoudi, for their support, guidance, and valuable feedback throughout my research journey. Their expertise and mentorship have been instrumental in shaping this thesis.

I extend my heartfelt thanks to Dr. Scott MacKenzie, whose excellent book and teachings provided me with a strong foundation in experiments with human participants and statistical analysis that has profoundly influenced my research methodology.

I wish to thank my former German professors, Dr.-Ing. Sabine Radomski, Dr. Paul Plöger, and Dr. Elmar Padilla, whose recommendations and belief in my academic potential opened the door for me to pursue my master's degree in Canada.

My deepest gratitude goes to my parents, Ina and Achim, and my grandparents, Annemarie and Helmut, whose unwavering belief in me has been my foundation throughout this journey. Despite the physical distance between us, their love has transcended borders, providing constant encouragement when I needed it most. Their wisdom, resilience, and unconditional support have not only shaped who I am today but have made this accomplishment possible. I dedicate this thesis to them, especially to my 95-year-old grandfather Helmut, whose lifetime of perseverance continues to inspire my own academic pursuits.

I would also like to acknowledge the Standards Council of Canada for their permission to include symbols from the ISO 2575 standard in this document.

Finally, I am thankful to everyone else who has contributed to my academic and personal growth during this challenging yet rewarding experience.

Contents

Page

Abstract	ii
Acknowledgments	iii
Table of Contents	iv
List of Figures	viii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Contributions	4
1.3 Thesis Overview	5
2 Related Work	8
2.1 Semiotics and Icon Interpretation	8
2.2 The Semantic Gap in Images and Image Descriptions	11
2.3 Large Vision-Language Models	13
2.4 Property Graphs and Knowledge Representation	17
2.5 Natural Language Interfaces to Databases	19
2.6 Visual Question Answering	21
2.7 Icon and Symbol Recognition	24

3	Method	27
3.1	Symbol Decomposition	27
3.1.1	Symbol Source and Extraction	28
3.1.2	Vector Graphics Processing	29
3.2	Hierarchical Labeling System	31
3.3	Graph-Based Representation	35
3.4	Proposed System	38
3.5	Question Collection	40
3.5.1	Human Questions	41
3.5.2	Synthetic Question Generation	42
4	Experimental Setup	44
4.1	Datasets and Symbols	44
4.2	Baseline Systems	46
4.3	Ablation Study	47
4.4	Evaluation Metrics	49
4.4.1	Automatic Evaluation	49
4.4.2	Human Evaluation	50
5	Results	53
5.1	Illustrative Example	53
5.2	Question Answering	55
5.3	Human Evaluation	57
5.4	Performance by Question Set	60
5.5	Analysis of Questions Structure	63

5.6	Use of Clarification Questions	65
5.7	Ablation Study	68
5.8	Failure Analysis	71
5.9	Limitations	77
6	Conclusion	80
	Bibliography	84
	Appendices	103
A	Prompts	103
A.1	Question Generation	103
A.2	Question Answering	106
A.3	LLM-as-a-Judge	108
B	Detailed Results Tables	110

List of Tables

4.1	Summary of baseline models used.	46
5.1	Answers and LLM Committee F_1 -scores for the human question: “What does the symbol of a horizontal line and three wavy arrows pointing up mean?”.	54
5.2	Example case: All models performed poorly.	73
5.3	Example case: Proposed System performed well, others poorly.	74
5.4	Example case: Proposed System performed poorly, others well.	76
B.1	Combined results for all model and metric combinations, on both human and synthetic questions.	110
B.2	Detailed results for all model and metric combinations on the synthetic questions dataset.	110
B.3	Detailed results for all model and metric combinations on the human questions dataset.	110

List of Figures

1.1	“Tire failure” symbol with different user descriptions from Fletcher et al.	2
3.1	Diverse selection of symbols taken from the ISO 2575 standard.	29
3.2	Comparison of SVG structure before and after processing.	30
3.3	Graphical representation of visual elements constituting the “Automatic High Beam” symbol.	33
3.4	Illustration of Gestalt principles in automotive symbols.	34
3.5	Tree representation of the labeled symbol A.36 – “Automatic High Beam”.	35
3.6	Low-level example of property transfer from nodes to edges during the property graph generation.	36
3.7	Architecture of the proposed system.	38
4.1	Property graph of decomposed symbols, visualization restricted to Annex A of ISO 2575 and limited to 750 relationships.	45
4.2	Visualization of the 50×6 incidence matrix used for the approximate BIBD.	51
5.1	Performance comparison of the proposed system versus baseline LLMs across evaluation metrics.	56
5.2	Mean human evaluation ratings on 5-point Likert scale.	58

5.3 Heat map of human evaluation ratings across ISO 2575 symbols. 59

5.4 Comparison of the LLM Committee F_1 -scores between human and synthetic questions. 61

5.5 Proportion of clarification questions generated by each model. 66

5.6 Clarification questions posed by the proposed system and three different models. 67

5.7 Comparison of LLM Committee F_1 -scores between direct answers and clarification questions. 68

5.8 Comparison of LLM Committee F_1 -scores between proposed system and RAG-based ablation. 69

Chapter 1

Introduction

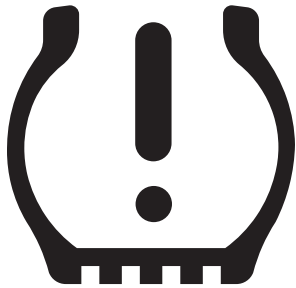
This thesis addresses the challenge of interpreting visual questions about abstract symbols, with a particular focus on icons used in automotive interfaces. The following introduction establishes the context of this research problem, outlines the novel contributions of this work, and provides a roadmap for the subsequent chapters. By examining the semantic gap between visual perception and symbol meaning, this research proposes a structured knowledge representation approach that significantly improves question interpretation and explanation capabilities compared to existing methods.

1.1 Background and Motivation

Icons¹ serve as the visual language of modern interfaces, enabling users to quickly recognize functions across linguistic and cultural boundaries. However, as interfaces become increasingly complex, particularly in specialized domains like automotive systems, the abstract nature of many icons can lead to misinterpretation with potentially serious con-

¹ Throughout this thesis, the terms “icon” and “symbol” are used interchangeably to refer to graphical representations that convey meaning or function within user interfaces.

sequences [1, 2]. A driver unable to correctly interpret a critical warning symbol may make decisions that compromise safety, highlighting the importance of bridging the semantic gap between visual perception and symbol meaning. Studies have shown that many automotive dashboard symbols fail to meet required comprehension levels, with up to half of tested icons falling below the 67% comprehension threshold [1].



Copied by Nicholas Dehnen with the permission of the Standards Council of Canada (SCC) on behalf of ISO. The standard can be purchased from the national ISO member in your country or the ISO Store. Copyright remains with ISO.

User Descriptions

“Two wishbone shaped lines joined at the bottom by a horizontal line with five ridges on the outside, separated by an exclamation mark.”

“A curved container with an exclamation mark inside it. The container does not have a top and has multiple small lines under it.”

“An exclamation mark surrounded by a U-shaped symbol that represents a tire.”

“A string-less, lyre-shaped symbol.”

“A pictogram of an exclamation mark surrounded by a tire.”

Figure 1.1: “Tire failure / malfunction” symbol (ISO 2575, K.10) with different user descriptions from Fletcher et al. [15]. Descriptions have been truncated for brevity.

This semantic gap – the discrepancy between an icon’s visual appearance and its intended meaning – presents a significant challenge in Human-Computer Interaction (HCI) [3–5]. When encountering unfamiliar icons, humans typically resort to analogical reasoning, comparing the symbol to known concepts based on visual similarities [6, 7]. While intuitive for familiar icons like a printer or trash bin, this process often fails with the abstract symbols prevalent in specialized domains like automotive. Figure 1.1 provides examples of the considerable variation in descriptions for the same symbol. Notably, its specific automotive purpose (relating to tires) was identified by only two respondents in the sample.

This problem extends far beyond automotive interfaces; for example, it also affects medical equipment interfaces, where misinterpretation can directly impact patient care [8], industrial control systems where operator errors may cause equipment damage or safety incidents [9], aviation cockpits with their dense display of critical indicators [10], nuclear facility control panels [11], and specialized software applications such as computer-aided design programs [12]. In each of these contexts, users encounter abstract symbols where the relationship between visual representation and function is not immediately apparent, creating potential for dangerous miscommunication across diverse high-stakes environments.

Recent advances in computer vision and natural language processing have produced powerful vision-language models capable of describing images and answering visual queries. However, these models exhibit notable limitations when confronted with abstract iconography [13, 14]. They often struggle to provide accurate explanations of specialized symbols, particularly when the connection between visual appearance and function is not immediately apparent. Prior research on vehicle dashboard icons has demonstrated that large vision-language models produce descriptions that score poorly in user evaluations despite achieving high automated metric scores [15]. This limitation becomes critical in safety-relevant contexts where precise understanding is essential.

The limitations of current approaches highlight the need for systems that can explicitly model the relationships between visual elements and their semantic interpretations, rather than relying solely on end-to-end learning. Structured knowledge representation offers a promising approach to address the symbol description-recognition gap and provide more precise and consistent explanations of abstract symbols, even when they share visual similarities but differ in meaning.

1.2 Contributions

This thesis presents a novel approach to answering visual queries about symbols, focused specifically on symbols for use in road vehicles, as standardized in the ISO 2575 [16, 17] standard. The semantic gap is addressed through a hybrid system that combines a structured knowledge representation using a property graph (cf. [18, 19]) capturing visual components, semantic relationships, and human-interpretable analogs, along with a natural language interface powered by a lightweight language model that translates user queries to database operations (cf. [20, 21]) while supporting clarification dialogues.

The key contributions of this work include:

1. A systematic methodology for decomposing abstract symbols into their constituent visual elements, enabling the representation of visual components at multiple levels of granularity. This hierarchical labeling system aligns with cognitive theories of visual symbol processing and supports both fine-grained visual elements and higher-level semantic groups.
2. A property graph schema that captures visual and semantic relationships between symbol elements, enabling reasoning about similarities and differences between symbols based on their visual characteristics. This structured knowledge representation serves as the foundation for accurate symbol identification and explanation.
3. A natural language query processing system with clarification capabilities that translates user descriptions into database queries, enabling intuitive interaction with the knowledge base. The system includes query refinement mechanisms to handle imperfect initial queries and generates clarification questions when multiple potential matches are identified.

4. Empirical evidence demonstrating the value of structured knowledge representation for bridging the symbol description-recognition gap. The proposed system achieved an LLM-as-a-Judge (cf. [22]) committee F_1 -score of 0.693 compared to the best baseline model’s score of 0.256, with human evaluation confirming its practical utility (average rating of 4.56 on a 5-point Likert scale versus 2.32 – 2.51 for baseline models).

The performance evaluation across different question sets revealed a surprising trend: all models performed better on human-generated questions than on synthetic questions, despite the latter being generated with access to the ground truth visual decomposition. This finding challenges common assumptions about the equivalence of synthetic and human data [23] in evaluating question-answering systems, suggesting that natural human questioning patterns may actually be more aligned with model capabilities than artificially constructed queries.

An ablation study further highlighted the advantages of the property graph-based approach over conventional retrieval-augmented generation (RAG) methods, demonstrating not only better performance but also substantially lower computational costs (2.5× to 150× reduction), reduced latency, and fewer API constraints. These practical advantages, combined with the higher accuracy, emphasize the value of structured knowledge representation in contexts where performance, efficiency, and explainability are essential.

1.3 Thesis Overview

The remainder of this thesis is organized as follows:

Chapter 2 presents a comprehensive review of related work. The literature review ex-

plores semiotics and icon interpretation theories, the semantic gap in images and descriptions, large vision-language models and their limitations with abstract symbols, property graphs for knowledge representation, natural language interfaces to databases, visual question answering approaches, and existing icon recognition and symbol datasets. This section establishes the theoretical foundations and situates the current work within the broader research landscape.

Chapter 3 presents the methodology used in this research. It begins with a detailed explanation of the symbol decomposition process, including the source and extraction of ISO 2575 symbols, vector graphics processing, and the hierarchical labeling system developed to annotate symbols at multiple levels of granularity. The section continues with a description of the graph-based representation using a property graph database, followed by the architecture of the proposed system and the approach to collecting both human-generated and synthetic questions for evaluation.

Chapter 4 describes the experimental setup, including the datasets and symbols used, the baseline systems selected for comparison, the ablation study design, and the evaluation metrics employed. The section details both automatic evaluation methods using metrics such as LLM-as-a-judge, BERTScore, METEOR, and ROUGE, as well as the human evaluation study conducted with six participants.

Chapter 5 presents the research findings, beginning with an illustrative example of system performance compared to baseline models. The section continues with comprehensive results on question answering performance, human evaluation ratings, comparative analysis of performance on human versus synthetic questions, and an analysis of question structure differences. Additional analyses examine the use of clarification questions across models and the results of the ablation study comparing the property graph ap-

proach to conventional RAG methods. The section concludes with a discussion of limitations identified through the experimental evaluation.

Chapter 6 summarizes the key contributions and findings of the research, highlighting the effectiveness of the hybrid approach combining semiotic theory, structured knowledge representation, and natural language processing for interpreting abstract symbols. The section concludes with promising directions for future work, including automated symbol decomposition, enhanced visual synonym coverage, advanced query refinement, cross-cultural adaptation, improved synthetic question generation, domain extension, and real-time optimization.

This thesis addresses an important challenge in human-computer interaction: enabling accurate interpretation and answering of human queries about abstract visual symbols. The findings demonstrate that explicitly modeling the relationships between visual elements and their semantic interpretations can significantly improve the accuracy and reliability of symbol explanations, representing a meaningful step toward creating more intuitive and explainable interfaces for abstract visual languages.

Chapter 2

Related Work

This chapter examines the theoretical foundations and technical approaches relevant to understanding and answering questions about abstract imagery, such as symbols. Beginning with semiotic theory, the review progresses through vision-language models, knowledge representation, and question-answering systems, establishing the interdisciplinary background necessary for this research.

2.1 Semiotics and Icon Interpretation

Understanding icons requires a foundation in semiotics, the study of signs and symbols. American scientist Charles Peirce in 1867 famously classified signs into *icons*, *indices*, and *symbols*, a framework still applied in modern visual communication [**burks_icon_1949**, 25]. In Peirce's terms, an icon bears a physical resemblance to its referent (e.g., a flame pictogram representing fire), an index has a causal or direct association, and a symbol relies on learned convention [26]. This terminology is equally useful for describing how automotive symbols convey meaning, as shown by contemporary analyses of visual in-

interfaces [27]. While icons ideally leverage resemblance for intuitive understanding, many interface symbols (especially standardized ones) function more like abstract symbols, requiring users to learn their meaning [28].

Interpreting an icon is not just about the graphic itself, but also the user's cognitive processes and prior knowledge [117]. Research shows that a person's experience, expertise, and cultural background strongly influence how they recognize and understand icons [pappachan_cultural_2008, 30]. For example, an automotive technician will more readily identify a brake system warning icon than a new driver. Horton (1994) highlighted the roles of perception and memory in icon recognition, noting that users recall familiar shapes faster [32]. Likewise, studies confirm that people from different cultures or domains may interpret the same symbol differently, based on learned conventions and context. Effective icon design thus often aims for familiar metaphors (a floppy disk icon meaning "save", etc.) that tap into shared cultural knowledge [littlemore_effect_2003]. In addition, *Gestalt psychology* explains how humans perceive visual elements as unified patterns rather than distinct parts. Principles such as *proximity*, *similarity*, and *closure* shape how icon components are grouped in perception [34]. For instance, if small triangular shapes are arranged closely in a circle, users may perceive them as a single gear or cog symbol. Leveraging these principles can make icons more quickly recognizable, and well-designed icons, often resembling real-world objects or common metaphors, feel intuitive [35]. For example, a printer icon looks like an actual printer. Recent work by Bühler et al. (2020) provides scientific guidelines to ensure pictograms are as "universal and intuitive" as possible [36]. Human-computer interaction (HCI) studies further show that familiarity and visual concreteness (how much an icon looks like what it represents) significantly improve interpretability [37]. Mcdougall et al. (1999) introduced metrics for

measuring how directly an icon conveys its meaning, and later evaluations reiterate that icons with obvious, concrete imagery are easier to understand across languages [38]. In practice, icons that closely resemble their referent (a flame for “fire”, a telephone handset for “call”) achieve higher recognition rates even by users who speak different languages [39].

Despite these general principles, specialized domains pose unique challenges. In domains like automotive systems, medicine, or industrial control panels, icons are often abstract and standardized, making them less self-evident. For example, the *ISO 2575* standard [16] defines dozens of automotive symbols that drivers are expected to be familiar with. These designs prioritize consistency, simplicity, and distinctiveness over resemblance, resulting in a noticeable semantic gap between the icon’s appearance and its actual meaning. Users without relevant background knowledge or training may not intuitively grasp that an icon of a circle surrounded by parentheses is supposed to represent a drum brake assembly. Studies in automotive human-machine interfaces (HMIs) have underscored that semantic distance – the gap between an icon’s look and its function – can impede quick understanding [40]. In Silvennoinen et al.’s (2017) experiments on in-car infotainment icons, the authors found that an optimal icon set is one where each icon is as *semantically close* as possible to its intended function (and clearly distinct from other icons) [40]. When symbols become too abstract, users must rely on memory or external aids to decipher them [41]. This is precisely the scenario in consumer road vehicles: many warning and status icons (brakes, transmission, ABS, etc.) are not self-explanatory and demand prior knowledge. The need to bridge this gap between visual form and meaning is a driving motivation for the research conducted in this thesis.

2.2 The Semantic Gap in Images and Image Descriptions

The discrepancy between what an image depicts visually and the meaning or description a human would ascribe to is commonly referred to as the semantic gap. This concept has long been recognized in computer vision and multimedia research [3]. In essence, low-level visual data (pixels, shapes, colors) rarely maps directly to the high-level concepts people think of when they see an image. Dorai and Venkatesh (2003) described this gap in the context of image retrieval, highlighting how difficult it is for computational features to capture the “meaning” of an image [4]. Despite decades of progress, the semantic gap remains a core challenge for image understanding systems.

Recent studies reinforce that bridging this gap is not only a technical problem but also a subjective one. Bahmanyar et al. (2015), for instance, explored differences between user interpretations and computer-generated interpretations of images. They found that even when algorithms improve and the gap narrows objectively, individual users may still be dissatisfied with how images are described [42]. One reason is what the authors termed a *linguistic* semantic gap: people use varied language and personal context when describing images, so a single generated description might not align with every user’s expectation [42]. Their work suggests that bridging the gap requires accounting for diverse user perspectives and descriptions, not just optimizing for one “ground truth” label [5]. In the domain of icons and symbols, this issue is acute. An abstract icon can generate widely different descriptions from different users – one might call the symbol “an oil can” while another calls it “a gravy boat shape”. Both refer to the same icon, but the semantics (visual vs. functional description) differ. This divergence complicates the task of automatically answering questions about icons, since a system must correctly understand a variety of different descriptions in order to be able to accurately assist the user.

Another facet of the semantic gap is the semantic distance between a symbol and its meaning, as noted earlier. Silvennoinen et al. (2017) demonstrated that when this distance is large (the icon looks very little like its meaning), users take longer or fail to make the connection [40]. This underscores a broader point: *recognizing* an object (or icon) in an image is not the same as *understanding* its significance. For example, an algorithm might detect that a dashboard icon contains a pictogram of a person wearing a seatbelt, but bridging the semantic gap means conveying *why* that icon is lit – e.g., “seatbelt reminder, please fasten your seatbelt”. The context in which an image appears plays a crucial role in closing this gap [30, 42]. An icon on a car dashboard means something very specific in that setting, which might differ from a similar-looking icon in another context. Therefore, incorporating contextual knowledge (such as the device or domain where the icon is used) is often necessary to truly bridge visual data and meaningful description.

To address these challenges, researchers are developing more sophisticated alignment techniques between vision and language. One emerging idea is to enrich the visual representation with structural and relational information before mapping to language. For example, Zhang et al. (2025) propose using panoptic segmentation (identifying all objects and regions in an image) and a Graph Spiking Hybrid Network to create richer visual semantic representations [43]. By combining spiking neural networks and graph neural networks, their model captures both local details and global context, aiming to preserve the relationships among visual elements [43]. The goal of such work is to generate image embeddings that carry higher-level semantic clues, thereby reducing the gap when these embeddings are translated into language [44]. While this approach targets general images, the principle is relevant to icons: parsing an icon’s components and relations (for example, a circle with an exclamation mark inside means something different than

the same exclamation mark inside a triangle) can help systems move from raw pixels to meaning. In summary, bridging the semantic gap requires combining low-level perception with high-level reasoning. This theme recurs in modern vision-language research and is directly pertinent to explaining icons, where the divide between what is seen and what it *means* is often vast. The design of the system and labeling methodology proposed in this work are explicitly informed by these findings.

2.3 Large Vision-Language Models

In recent years, vision-language models (VLMs) have advanced significantly in connecting visual content with natural language, representing a crucial step toward addressing the semantic gap. Early image captioning systems from the mid-2010s typically employed a convolutional neural network (CNN) as an image encoder and a recurrent neural network (RNN) as a decoder to generate textual descriptions [45]. The *Show-and-Tell* model by Vinyals et al. (2015) and *Show-Attend-and-Tell* by Xu et al. (2015) demonstrated that neural networks could generate coherent captions for commonplace images by learning from extensive labeled datasets [45, 46]. These models encoded images into feature vectors and learned to decode those features into descriptive sentences, effectively establishing a mapping from visual features to linguistic expressions. While functional for photographs depicting common objects and scenes, these models frequently produced generic captions and encountered difficulties with less common or more abstract visual content.

A notable advancement in vision-language alignment emerged with the introduction of *CLIP* (Contrastive Language-Image Pre-training) by Radford et al. (2021). *CLIP* implements an alternative approach: rather than generating captions for images, it learned to

associate images with corresponding text by training on an extensive dataset of 400 million image-text pairs collected from the internet [47]. By establishing a shared embedding space for images and text, CLIP enables zero-shot classification. Specifically, given a set of candidate textual descriptions, it computes similarity scores between the image embedding and the text embeddings to determine which description best matches the image [47]. This approach marked a significant advancement in bridging vision and language representations. For instance, when provided with candidate descriptions for dashboard icons, CLIP can match an image to an appropriate description (e.g., “windshield defrost icon”), even if that particular icon was not explicitly labeled in its training data, as long as similar images and descriptive text were present. However, CLIP does not generate labels or comprehensive explanations; its output is limited to selecting the most relevant textual description among those provided.

Simultaneously, image captioning models continued to evolve. The CNN-RNN paradigm has been superseded by more sophisticated transformer-based architectures and large-scale training methodologies. Models such as *BLIP* (Bootstrapping Language-Image Pre-training) [48] and its successor *BLIP-2* [49] utilize transformers to both comprehend and generate language about images. These models are trained on millions of image-caption pairs, enabling them to produce more nuanced and contextually appropriate descriptions than earlier systems. For example, BLIP can analyze a photograph of a complex scene and generate a detailed sentence, while BLIP-2 further enhances this capability through a two-stage approach in which a vision encoder is coupled with a language model, refining the image-to-text generation process [48, 49]. Such models approach explaining content in a manner resembling human description, but when confronted with abstract or domain-specific images (such as instrument panel icons), they often encounter limitations due to

insufficient specialized background knowledge.

The current trajectory involves integrating Large Language Models (LLMs) with visual processing capabilities – effectively creating multi-modal LLMs, also referred to as Large Vision Language Models (LVLMs). These systems aim to harness the general reasoning and conversational capabilities of LLMs (such as GPT-style models) and incorporate visual information through an attached vision encoder. One example is *LLaVA* (Large Language and Vision Assistant) by Liu et al. (2023), which connects a pretrained vision encoder to an instruction-tuned LLM (Vicuna, based on Meta’s Llama models) [50]. *LLaVA* is fine-tuned to respond to user prompts about images, allowing it to engage in dialogue regarding visual content. Similarly, OpenAI’s *GPT-4* and the more recent *o3-mini* reasoning models support multi-modal input, capable of processing image inputs alongside text, and demonstrating sophisticated capabilities in describing and analyzing images [51, 52]. These vision-augmented LLMs represent the state of the art in image description technology – they not only describe but can also reason about images to a certain extent, respond to inquiries, and provide explanations. For instance, when presented with a dashboard icon, an LVLM could potentially not only identify it but also elaborate on its meaning, contingent upon having the requisite background knowledge.

Despite these advancements, current LVLMs continue to face challenges with non-natural, abstract images, including symbols or icons, which is particularly relevant to the problem domain [53]. Even the sophisticated *GPT-4* model, when tested on unfamiliar icons, may accurately describe the visual features (e.g., “a symbol showing a person with a seatbelt”) without fully comprehending the function or specific context (e.g., that it indicates the seatbelt is not fastened). A recent study by Fletcher et al. (2024) illuminates these limitations. The researchers presented state-of-the-art LVLMs with automotive dashboard

icons and tasked them with generating explanations, introducing a new dataset comprising 408 vehicle dashboard icons with ground-truth descriptions for evaluation [15]. They evaluated models including OpenAI’s vision-enabled GPT-4o, Anthropic’s Claude 3.5, and the open-source model LLaVA. Notably, they had the models produce two categories of descriptions: a *visual* description (the icon’s appearance) and a *functional* description (its meaning within the vehicle context). The findings indicated that the proprietary models (GPT-4o and Claude) performed adequately in generating accurate descriptions when provided with both the icon image *and* contextual information from the vehicle manual, whereas LLaVA demonstrated significantly lower performance. Indeed, without contextual information, all models frequently misinterpreted the correct meaning – reinforcing that visual information alone is insufficient for these abstract symbols. Optimal performance was achieved when models had access to multi-modal inputs (the icon combined with explanatory textual context), which substantially outperformed either image-only or context-only inputs for identifying the icon’s function. This suggests that even sophisticated vision-language models benefit from structured knowledge or contextual cues when interpreting symbols. In conclusion, while LVLMs represent the current state of the art in mapping images to language, they do not constitute a panacea for image understanding. Particularly for non-natural images, they frequently require supplementation with domain knowledge to appropriately comprehend the semantic meaning and to facilitate reasoning about the images. This insight motivates the research conducted in this thesis, while also naturally directing attention toward approaches that combine (vision-) language processing with knowledge representations, as will be explored in subsequent sections.

2.4 Property Graphs and Knowledge Representation

Knowledge representation in AI leverages structured repositories called knowledge bases, with *knowledge graphs* (KGs) representing a modern evolution that organizes information as nodes (entities, or concepts) and labeled edges (relationships). As defined by Hogan et al. (2021), a knowledge graph is “a graph of data intended to accumulate and convey knowledge of the real world”. This graph-based structure facilitates both human and machine interpretation, making KGs valuable for reasoning and query answering [54]. The historical development of KGs traces back to semantic networks of the 1960s, evolving through expert systems in the 1980s–90s, and gaining momentum with the Semantic Web movement [55]. Significant milestones include DBpedia’s extraction of structured data from Wikipedia (2007) [56], Freebase’s collaborative knowledge database (2007) [57], and Google’s introduction of its Knowledge Graph (2012) [58], which popularized the term by framing search as connecting entities rather than isolated keywords.

Two primary frameworks exist for implementing knowledge graphs: the Resource Description Framework (RDF) and property graphs [54]. RDF represents information through subject-predicate-object triples stored in specialized databases called triplestores [59]. This approach offers strong semantic interoperability through standardized query languages (SPARQL) and ontology definitions (OWL) [60, 61], operating under an open-world assumption that encourages data integration. In contrast, property graphs allow both nodes and edges to have properties attached as key-value pairs [18], offering a more flexible representation where additional attributes are embedded directly in the graph structure rather than as separate triples. Property graphs, implemented in systems like Neo4j [62] and accessed via query languages such as Cypher [21], typically operate without enforcing a global schema upfront, making them more approachable for domain-

specific applications. The choice between these models depends on use case requirements: RDF excels in standardization and formal semantics, while property graphs prioritize flexibility and intuitive data representation [63].

Knowledge graphs play a crucial role in reasoning and question answering by making relationships explicit and enabling multi-hop inference [54]. Knowledge Graph Question Answering (KGQA) systems translate natural language questions into graph queries to retrieve answers by traversing the structured relationships [64]. Recent approaches combine neural language models with knowledge graphs to improve reasoning capabilities [65], allowing systems to verify or augment generated answers with factual information. Beyond textual applications, knowledge graphs have become increasingly important in visual reasoning through scene graphs – structured representations that capture visual content by representing detected objects as nodes and their spatial, semantic, or functional relationships as edges, allowing machines to understand and reason about the contents of images or videos [66]. By abstracting images into graph form, complex visual reasoning becomes more tractable, improving performance on tasks like image captioning and Visual Question Answering (VQA) [66].

Multi-modal knowledge graphs extend these capabilities by connecting visual data with traditional knowledge [67]. VisualSem [68] exemplifies this approach with approximately 90,000 visual concept nodes linked to textual definitions and example images, providing an integrated resource for grounding and data augmentation in vision-and-language tasks. Similarly, VisionKG integrates 37 vision datasets into a unified semantic resource comprising over 617 million RDF triples describing about 61 million entities [69]. Such resources enable complex queries combining visual and semantic criteria, providing a reasoning layer above raw visual recognition that supports sophisticated interpretation and

question answering about visual content [67].

This thesis leverages these insights to develop a question-answering system for symbolic images by combining a property graph knowledge base with Large Language Models. While LLMs excel at interpreting natural language and generating fluent responses, they lack grounded knowledge of specific domains such as symbol meanings (cf. [15]). In particular, the property graph constructed for this work serves as a grounding mechanism in two ways, helping to match user queries with the correct symbols, as well as ensuring that responses are based on encoded knowledge rather than the language model’s prior training. This approach combines the LLM’s linguistic flexibility with the knowledge graph’s precise, domain-specific information and reasoning capabilities, enabling robust symbol question-answering.

2.5 Natural Language Interfaces to Databases

To be useful, the information in a database or knowledge graph needs to be accessible to end-users, ideally in an intuitive manner. Natural Language Interfaces to Databases (NLIDB) address this need by enabling users to query structured data using everyday language rather than formal query languages such as SQL or SPARQL [70]. This approach significantly lowers the barrier to data access, especially for domain experts who may lack technical database skills. NLIDB research spans over five decades, evolving from rule-based systems to modern neural approaches that achieve increasingly higher accuracy on complex queries.

Early NLIDB systems emerged in the 1970s with pioneering works such as LUNAR, which answered questions about moon rock samples, and BASEBALL for sports statistics queries

[20]. The 1980s saw systems like Chat-80, built in Prolog, which translated English questions into logical queries and influenced many subsequent systems [20]. Despite their technical innovations, these early systems were limited to narrow domains and struggled with linguistic variations. Users often needed to learn what phrasing the system could understand, as unexpected formulations frequently led to parsing failures [70]. The early 2000s brought more precisely constrained approaches, exemplified by PRECISE, which guaranteed correct interpretations for a subset of English by combining linguistic parsing with graph-matching techniques [71]. This evolution reflected persistent trade-offs between coverage, accuracy, and domain portability.

Modern NLIDB research has experienced a resurgence driven by advances in neural networks and the creation of benchmark datasets. Contemporary systems primarily frame the problem as semantic parsing, using sequence-to-sequence models to translate natural language questions into database queries [72]. Two influential benchmarks have guided this progress: WikiSQL, containing over 80,000 question-SQL pairs for single-table databases [73], and Spider, featuring 10,000+ examples across 200 different databases requiring multi-table joins and complex query structures [72]. Neural architectures typically employ encoder-decoder frameworks initialized with pre-trained language models, with innovations like schema linking to align question terms with database elements, and constrained decoding to ensure syntactically valid queries [74]. These advances have significantly improved cross-domain generalization and complex query handling.

Following a pattern observed in many other fields, the integration of Large Language Models (LLMs) also represents the latest frontier in NLIDB research. Models like GPT-4 demonstrate impressive in-context learning capabilities, generating correct SQL queries from natural language questions with minimal examples [75]. Their pre-training on mas-

sive text corpora enables better handling of linguistic variation compared to models trained solely on NLIDB datasets. Beyond query generation, modern LLMs facilitate knowledge-grounded responses, providing explanations alongside results and clarifying reasoning steps in query construction [76, 77]. This approach enhances transparency and user trust while maintaining the technical accuracy of database interactions. Current research focuses on prompt engineering for complex schemas, implementing safeguards for potentially incorrect outputs, and efficient schema information retrieval for large databases [76].

This thesis draws inspiration from NLIDB approaches to bridge the gap between user questions and structured data about symbols. The developed system translates natural language questions into Cypher queries, which are executed over a property graph knowledge base of visual symbols and their meanings. Similar to how modern NLIDBs combine neural language understanding with structured database queries, this approach leverages an LLM’s linguistic capabilities while ensuring responses remain grounded in verified knowledge.

2.6 Visual Question Answering

Enabling machines to answer questions about visual content bridges a vital gap in human-computer interaction, allowing users to query visual information as naturally as they would textual data. Visual Question Answering (VQA) represents an interdisciplinary research area spanning computer vision, natural language processing, and knowledge representation. The task involves processing an image and a natural language question about that image to produce an accurate textual answer. Unlike image captioning, VQA requires

goal-oriented understanding: the system must identify what information the question seeks and locate relevant visual evidence to formulate an answer [78]. This necessitates scene comprehension and often logical inference capabilities. The VQA benchmark introduced by Antol et al. (2015) exemplifies this range of complexity, including questions from simple object identification to those requiring sophisticated reasoning about image content [78].

Modern VQA models typically integrate CNNs for image processing with transformers for question answering. These architectures may perform adequately on common images by learning statistical patterns from training data [79]. However, many questions inherently require information beyond what is visible in the image itself. This limitation has motivated the development of *knowledge-based VQA*, which extends vision-language pipelines by incorporating external information sources to address questions that visual analysis alone cannot answer [80]. Such systems perform multiple steps: detecting visual elements, identifying them, and retrieving relevant information from a knowledge repository.

Several datasets have advanced knowledge-based VQA research. The Fact-based VQA (FVQA) dataset by Wang et al. (2018) pioneered questions that explicitly required external knowledge to answer correctly [81]. Building on this approach, Knowledge-based VQA (KVQA) by Shah et al. (2019) introduced 183,000 question-answer pairs focused on real-world entities in images [82]. KVQA requires systems to link visual entities to knowledge graph entries (specifically *Wikidata* [83]) and perform multi-hop reasoning to retrieve relevant attributes. These benchmarks demonstrated that structured knowledge bases are essential for answering questions beyond the scope of visual content alone.

Recent advancements in VQA research have explored hybrid approaches that combine

deep neural networks with knowledge-based systems or language models. Tao et al. (2025) propose integrating large language models (LLMs) into the VQA pipeline, where the LLM interprets questions and guides knowledge graph retrieval [84]. In their zero-shot VQA framework, the LLM translates visual scenes and questions into knowledge graph queries through “strategic question search”, while the graph provides factual grounding. This architecture leverages complementary strengths: LLMs excel at language understanding and logical connections, while knowledge graphs provide verifiable facts. The integration yields improved accuracy on tasks neither component could solve independently [84].

Interactive capabilities further enhance VQA system usability by addressing ambiguities through clarification requests. Recent research in interactive VQA emphasizes dialogue capabilities to improve user experience [85]. This conversational approach requires systems to maintain context and generate appropriate follow-up questions – capabilities that LLMs effectively manage. The combination of visual processing, structured knowledge representation, and conversational interfaces creates systems that more closely approximate human-like assistance, integrating perception, memory, and communication functions.

However, abstract, non-natural images such as technical schematics or symbols present a particular challenge for VQA systems [86]. These visual elements often appear infrequently in training data, making them difficult for models to recognize reliably. Furthermore, questions about symbolic meaning inherently require external knowledge, as the semantic interpretation of an abstract symbol is not directly evident from its visual features alone. This domain highlights the importance of knowledge-based approaches that can bridge the gap between visual perception and functional meaning. This thesis builds upon these insights by developing a knowledge-based VQA system specifically for

abstract symbol question answering, with a focus on automotive icons.

2.7 Icon and Symbol Recognition

Icons and symbols represent a class of visual elements that convey information through simplified graphical forms. These visual abstractions appear across numerous domains: graphical user interfaces employ icons to represent commands or statuses [87]; automotive dashboards display standardized warning symbols defined by international standards such as ISO 2575; aviation and industrial control panels utilize symbolic indicators; and various types of documents incorporate symbols for clear and concise communication [88]. These examples illustrate that recognition and interpretation of these symbolic elements constitute a fundamental problem whenever abstract visuals are used to communicate functionality or information.

Early computer vision approaches to icon recognition relied on template matching and geometric detection of fundamental shapes, though these methods struggled with variations in scale and rotation [89, 90]. The 2000s saw adoption of invariant feature extraction techniques like SIFT and SURF for more robust symbol matching [91–93]. The paradigm shifted significantly during the 2010s with data-driven deep learning, as CNNs became the predominant solution, learning visual patterns directly from labeled datasets [94]. CNN-based models have achieved near-human performance in related tasks like traffic sign recognition, with models smaller than 1MB in size, capable of running on embedded devices [95]. Recent years have seen exploration of Vision Transformers (ViTs) and attention-based architectures, which can outperform CNNs on many image recognition tasks through their self-attention mechanisms that provide access to global image infor-

mation [96]. However, domain-specific performance varies significantly – while ViTs demonstrate superior robustness to noise and image augmentation in general computer vision tasks, CNNs may still hold advantages in specialized applications like traffic sign recognition, where performance gaps of up to 12% have been observed in favor of convolutional architectures [97].

Several research domains beyond traffic sign recognition share techniques with icon recognition while addressing symbols in specific contexts. Recent advances in the field of engineering diagram analysis combine object recognition with region proposal to interpret representations such as electrical schematics or flowcharts [98]. Logo detection treats corporate emblems as target classes using object detection methodologies adapted from the icon recognition domain, with state-of-the-art methods employing approaches such as detection transformers and feature pyramid networks [99]. Optical Character Recognition (OCR) focuses on recognizing characters (alphanumeric symbols) in images, a domain in which contemporary LVLN-based approaches such as Mistral OCR [100] have shown vast improvements over traditional techniques [101]. Mathematical expression recognition addresses the identification and spatial parsing of mathematical notation, requiring both symbol classification and analytical understanding of two-dimensional arrangements [102].

Despite significant advancements, substantial research gaps persist in this domain. All existing datasets identified through systematic literature review solely provide class or sentiment labels for icons without deeper semantic annotations or decomposition into constituent elements. This limitation constrains the ability of models to perform higher-level reasoning about symbolic representations. While contemporary techniques can reliably identify which icon appears in an image, understanding why an icon conveys its

particular meaning or analyzing its compositional structure remains challenging. Initial efforts to address this gap include datasets that pair icons with human-written descriptions and definitions [15], though such resources remain limited in scope and availability.

This thesis extends the work of Fletcher et al. by systematically decomposing and semantically labeling ISO 2575 symbols used in road vehicles. Through this structured knowledge representation approach, the system bridges the identified research gap, enabling effective reasoning about both visual components and functional meanings of automotive icons.

Chapter 3

Method

This chapter introduces a novel knowledge-based system designed to interpret queries and explain automotive symbols. The approach combines structured visual representation with natural language processing capabilities, addressing limitations in current symbol recognition technologies. A graph-based architecture is presented that enables transparent reasoning processes and reduces the occurrence of hallucinations, which are common with purely LLM-based approaches. The following subsections detail each component of this integrated system, from initial symbol decomposition to the final user interaction framework.

3.1 Symbol Decomposition

To bridge the semantic gap between visual perception and symbol meaning, a structured decomposition approach for ISO 2575 symbols was developed. This methodical decomposition serves as the foundation for the knowledge representation system, enabling explanations of abstract symbols by modeling relationships between visual elements and their

semantic interpretations. Unlike end-to-end learning approaches, this method captures both the visual components and their semantic relationships in a format that supports reasoning and explanations.

3.1.1 Symbol Source and Extraction

The International Organization for Standardization (ISO) provides a comprehensive set of standardized symbols for use on road vehicles through the ISO 2575 standard [16, 17]. The standard is organized into 19 annexes, each grouping symbols by their specific functional domain within road vehicle design. For example, separate annexes address areas such as lighting, engine and fuel systems, vehicle handling, safety, and more. In essence, these annexes reflect the different operational and safety functions integral to the vehicle, ranging from performance-critical systems to communication and comfort features. Additionally, the standard categorizes the symbols contained within the standard into three functional groups: control, telltale, and indicator. Each category serves specific communication purposes within vehicle interfaces.

Figure 3.1 presents 20 symbols from annexes A–D, F–H, and J–M. The symbols shown in this figure were also used for the following question collection from human participants in Section 3.5.1. Annexes E, I, and N–S were deemed not relevant to this research and were excluded. These annexes primarily contain symbols specific to commercial trucking operations (e.g., specialized trailer configurations) or basic shapes used for creating new symbols. Including these would have introduced domain-specific knowledge requirements beyond what average drivers encounter, potentially skewing the evaluation of the system’s utility for the intended user base. To facilitate analysis, information was first extracted from the ISO 2575 documentation using a pipeline of computational tools in-

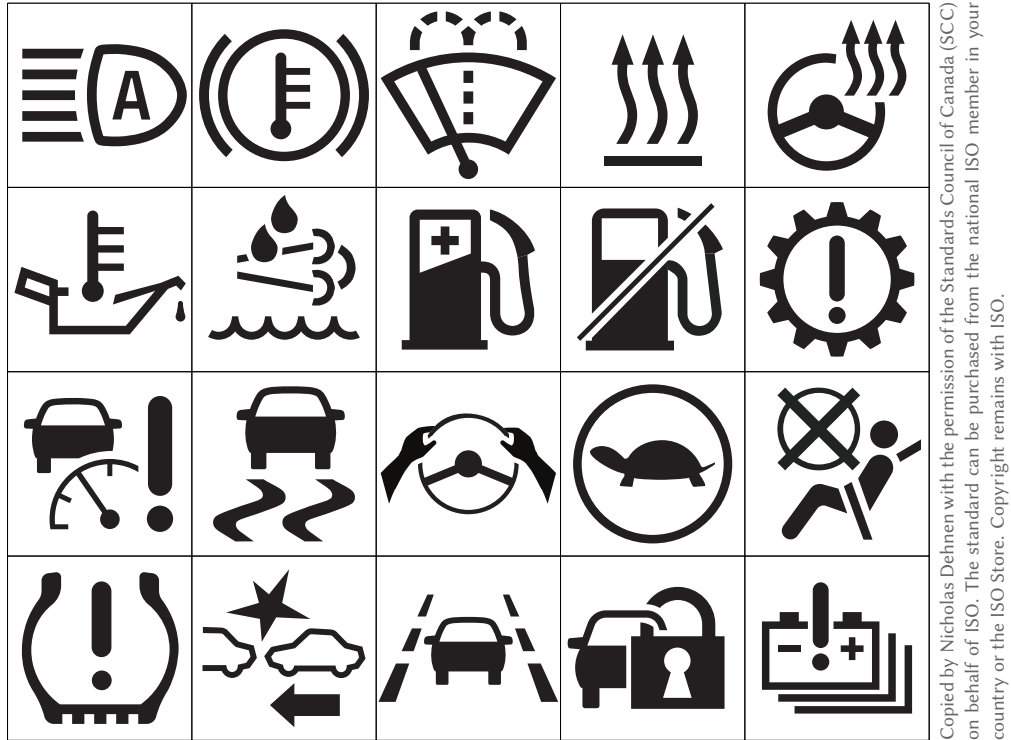


Figure 3.1: Diverse selection of symbols taken from the ISO 2575 standard [16, 17].

cluding Python, Pandas, SpaCy, and PyMuPDF. This automated extraction process converted the information into a machine-readable YAML format, establishing a structured foundation for subsequent processing steps.

3.1.2 Vector Graphics Processing

Symbol vector graphics were either manually extracted from the ISO documentation or carefully recreated using the vector graphics editor Inkscape¹. These graphics were stored in the Scalable Vector Graphics (SVG) format and were simplified and cleaned using both Python processing and Inkscape. This cleaning procedure involved several transformations to ensure consistency and simplicity:

¹ <https://inkscape.org/>

- (i) Conversion of strokes to fills for uniform representation.
- (ii) Transformation of cut paths into regular paths.
- (iii) Simplification or removal of style attributes.
- (iv) Direct application of transforms to path definitions.
- (v) Decomposition of combined paths into individual paths.

The processing pipeline significantly simplified the SVG structure, as illustrated in Figure 3.2. In SVG's XML structure, elements like `<g>` serve as containers to group related

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" />
  <defs id="defs1">
    <clipPath id="clipPath1489">
      <path d="m 85.219,247.983 h 68.671 ..." />
    </clipPath>
    ...
  </defs>
  <g id="g1486" transform="matrix(0.2645 ..." />
    <g id="g2191" />
    <path id="path1490" d="M 0,0 H -22.9 ..." />
    <path id="path1491" d="M 0,0 C 0.15, ..." />
    <path id="path1493" d="m 0,0 c -0.01 ..." />
    ...
  </g>
</svg>
```

Listing 3.1: Before Processing

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" />
  <g id="automatic-high-beam">
    <path id="letter-A" d="m 15.473,8.91 ..." />
    <g id="high-beam">
      <path id="head-light" d="m 14.366,5 ..." />
      <g id="light-beam">
        <path id="line-1" d="m 10.027,6.0 ..." />
        <path id="line-2" d="m 1.941,8.73 ..." />
        <path id="line-3" d="m 1.941,11.4 ..." />
        <path id="line-4" d="m 1.941,14.2 ..." />
        <path id="line-5" d="m 9.608,16.9 ..." />
      </g>
    </g>
  </g>
</svg>
```

Listing 3.2: After Processing

Figure 3.2: Comparison of an SVG symbol's structure before (left) and after (right) processing. The corresponding XML code is shown for each version. Key SVG elements like `<g>` (group) and `<path>` are visible. The transformation involves simplifying the structure, removing transformations (`transform`), styles (`style`), and clip paths (`clip-path`), and converting shapes into explicit paths defined by the `d` attribute. Path drawing commands have been trimmed for brevity.

graphical elements, while `<path>` defines a shape. The geometry of a path is specified within its `d`-attribute using a series of commands and coordinates. For instance, in the processed example shown, the `d`-attribute contains commands such as:

- *m* (*moveto*): Starts a new path component at coordinates relative to the current point (e.g., `m 15.473,8.91`).
- *c* (*curveto*): Draws a cubic Bézier curve using control points and an endpoint, all relative to the current point (e.g., `c -0.565333,2.13 -1.122,4.17`).
- *Z* or *z* (*closepath*): Closes the current shape by drawing a straight line from the current point back to the start point of the current sub-path.

Many other commands exist for drawing lines (*L*, *l*, *H*, *h*, *V*, *v*), quadratic curves (*Q*, *q*, *T*, *t*), and arcs (*A*, *a*). The cleaning process aimed to represent shapes primarily using these path commands, removing complexities like transformations and styles for easier analysis. For a complete definition of SVG elements and path commands, the W3C Scalable Vector Graphics specification provides a comprehensive reference².

The choice of the SVG format served multiple purposes in this research. First, SVG provides a natural hierarchical structure through its XML-based representation, mirroring the conceptual hierarchy of elements within symbols. Second, SVG enables programmatic access to individual graphic elements, facilitating both manual annotation and potential future automation. Third, the vector format ensures resolution independence, providing consistent visual representation across different display environments.

3.2 Hierarchical Labeling System

A hierarchical labeling system was developed to annotate the symbols at multiple levels of granularity, reflecting established cognitive theories of visual symbol processing, as

² <https://www.w3.org/TR/SVG11/>

previously explored in Section 2.1. The system employs SVG attributes to encode both visual and semantic information:

- (a) The `id`-attribute identifies each path with a descriptive label (e.g., “wheel”).
- (b) A `synonym`-attribute captures potential visual synonyms or analogies for the path (e.g., “u-shape” for “tire-cross-section”).
- (c) Additional properties such as `orientation` (horizontal, diagonal, vertical) and `appearance` (dotted, dashed, solid) are introduced as needed and encoded as attributes.

The system distinguishes between two key levels of visual representation:

- (i) **Basic visual elements:** These represent the smallest, non-divisible components of a symbol. They include geometric primitives (lines, circles, triangles, rectangles, etc.) and recognizable shapes that function as atomic units within the symbol system (e.g., a stylized turtle shape or letter forms). While the identification of these elements maintains some inherent subjectivity depending on cultural background and domain knowledge, they represent the most fundamental units that maintain recognizable identity within the symbol context.
- (ii) **Semantic units:** These are meaningful groupings of basic visual elements that collectively convey a higher-level concept through the application of Gestalt principles. For example, five individual lines may be grouped into a “light beam” semantic unit, or two crossed lines might form an “X” that signifies cancellation or warning. These units represent an intermediate level of meaning between basic elements and the complete symbol.

In the SVG implementation, paths represent the basic visual elements. To capture higher-level semantic relationships, related visual elements are grouped into semantic units using SVG groups. For instance, two wheels and a trailer-bed would be contained within a semantic group labeled “trailer”. Similar to paths, these groups can possess property attributes of their own. At the highest level, the top-level group encompasses all semantic groups and corresponds to the symbol’s official designation, such as “automatic-high-beam”. Figure 3.3 illustrates the labeling approach as applied to symbol A.36 (“Automatic

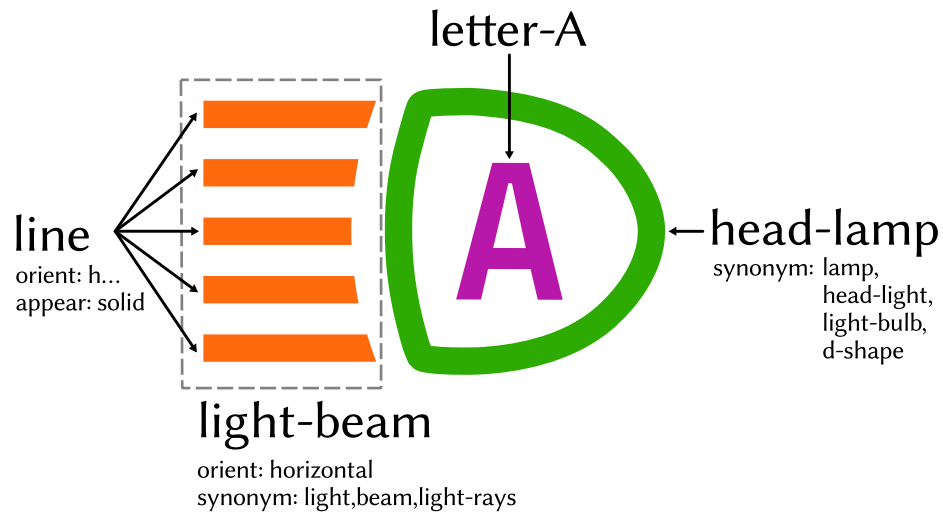


Figure 3.3: Graphical representation of visual elements constituting the “Automatic High Beam” symbol.

High Beam”). The figure identifies basic visual elements including “line”, “letter-A”, and “head-lamp”. The semantic group “light-beam” consists of five individual “line” elements that collectively represent the light beam. For clarity, the figure presents a simplified view; not shown are the semantic group “high-beam” (which would encompass everything except the “letter-A”) and the top-level group “automatic-high-beam”.

This hierarchical structure aligns with established cognitive theories of visual symbol processing: As Horton discusses, humans perceive and interpret symbols by first recognizing primitive visual elements before assembling them into meaningful wholes [32]. The ap-

proach mirrors this bottom-up processing in the tree structure of the SVGs, where paths represent basic visual elements that, when combined into groups, form semantic units. *Gestalt* psychology provides the organizational rules guiding this assembly process: The Gestalt principles of proximity, similarity, and continuity determine how these elements coalesce into a unified and meaningful symbol, thus explaining how humans perceive visual elements as unified patterns rather than isolated parts [34].

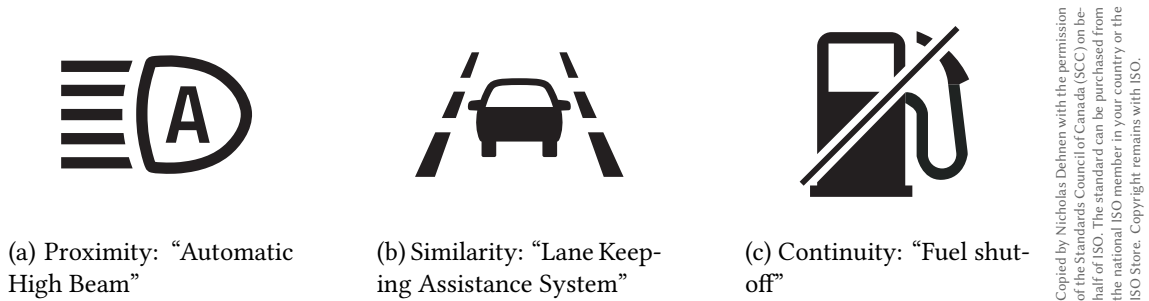


Figure 3.4: Illustration of Gestalt principles in automotive symbols. (a) Proximity: the five horizontal lines are perceived as a unified light beam due to their close spatial arrangement. (b) Similarity: the two diagonal lines are interpreted as road markings through their similar appearance. (c) Continuity: despite being interrupted by a diagonal line, the fuel pump shape is perceived as a complete object due to the principle of good continuation.

The application of Gestalt principles in symbol design, as illustrated in Figure 3.4, demonstrates how these perceptual laws govern the formation of semantic units. The principle of proximity allows multiple distinct lines to be perceived as a unified “light beam” when positioned close together (Figure 3.4a). Similarity enables viewers to interpret parallel diagonal lines as representing the same concept – road markings – in the stability control symbol (Figure 3.4b). Continuity explains how observers mentally complete interrupted forms, such as the fuel pump in the “Fuel shut-off” symbol (Figure 3.4c), despite the diagonal line breaking its visual continuity. These principles directly inform the semantic grouping strategy by providing the cognitive basis for determining which basic visual elements should be combined into meaningful semantic units.

The labeling system reflects this cognitive process by modeling both individual elements (e.g., “line”, “circle”) and their meaningful groupings (e.g., “light-beam”, “high-beam”). Its design was inspired by an inspection of the data and the findings presented by Fletcher et al. in [15], while its structure was informed by cognitive theories of visual perception to ensure alignment with human interpretive processes.

3.3 Graph-Based Representation

To facilitate computational analysis and knowledge representation, the labeled vector graphics were transformed into directed multi-graphs using Python and NetworkX. During this transformation, multiple instances of identical visual elements were consolidated into single nodes, with instance-specific properties transferred to the connecting edges of type CONTAINS. This approach created a hierarchical structure that mirrors how humans perceive and interpret symbols – recognizing basic shapes that combine into meaningful semantic units.

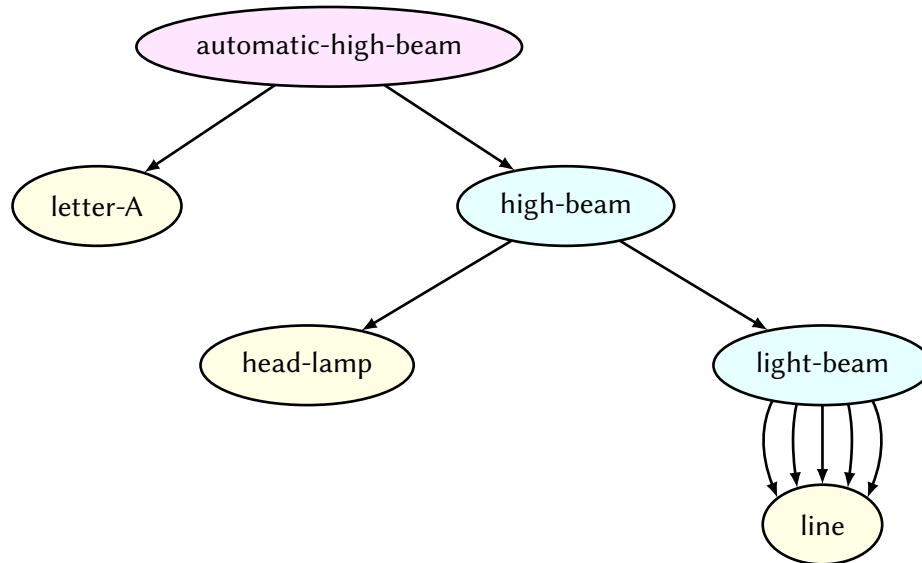


Figure 3.5: Tree representation of the labeled symbol A.36 – “Automatic High Beam”.

Figure 3.5 presents the same symbol as Figure 3.3 but as a tree representation that clarifies the underlying graph structure. Notably, despite the symbol containing five separate line elements in the light beam, the graph contains only a single “line” node with five incoming edges from the “light-beam” node. This representation effectively captures the hierarchical nature of the labeling system: the top-level node represents the complete symbol, internal nodes correspond to semantic groups, and leaf nodes represent basic visual elements that cannot be easily decomposed further. The following Figure 3.6 visualizes the node consolidation procedure during the property graph generation on a low-level, using the “line” elements and the “light-beam” semantic unit shown in Figure 3.3. Since many symbols share common elements, properties are transferred to the edge to avoid property conflicts within the nodes themselves. The directionality of the edges is used to indicate that its properties apply to the target node.

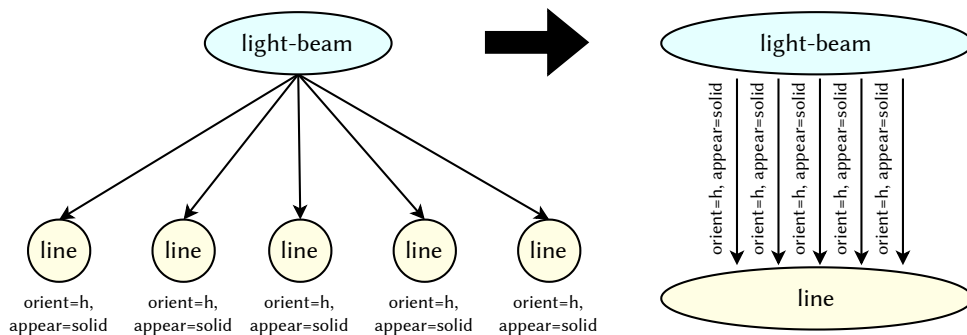


Figure 3.6: Low-level example of property transfer from nodes to edges during the property graph generation.

In the final step of the processing pipeline, all individual directed multi-graphs were merged into a comprehensive graph that encompasses the entire symbol set. This unified graph was exported in GraphML format for subsequent import into a Neo4j property graph database.

The choice of a property graph over alternative knowledge representations is supported by research by Yakin and Totu, showing that humans mentally connect symbols to referents through associative networks rather than strict taxonomies [26]. By enabling properties on edges and nodes, the model can capture the nuanced relationships that Silvenoinen et al. identified as crucial for reducing semantic distance in icon interpretation [40]. And unlike conventional RDF graphs, the property graph model offers flexibility in representing complex relationships while supporting efficient querying through the Cypher query language. This capability is essential for capturing the rich semantic attributes described in Section 3.2.

Furthermore, this structured approach enables transparent reasoning chains that can be presented to users, addressing limitations of black-box Large Language Models (LLMs) in safety-critical contexts. Unlike end-to-end neural approaches that might struggle with the semantic gap (cf. [4]), this explicit modeling of visual components and their semantic relationships facilitates more explainable reasoning. The multi-level graph structure – connecting basic visual elements to semantic groups to complete symbols – enables querying at different levels of abstraction, from low-level visual features (“symbols with circles”) to high-level semantic concepts (“warning indicators for the brake system”). This capability is crucial for bridging what Bahmanyar et al. termed the “linguistic semantic gap” between user descriptions and formal representations [42]. The following section details how this graph representation integrates with language model capabilities to create a complete symbol interpretation system.

3.4 Proposed System

The proposed system integrates property graph database capabilities with language model intelligence to effectively interpret and respond to user queries about dashboard symbols. This integration leverages the complementary strengths identified by Tao et al.: the factual grounding of knowledge graphs and the linguistic flexibility of language models [84]. As illustrated in Figure 3.7, the system architecture employs a multi-stage approach that

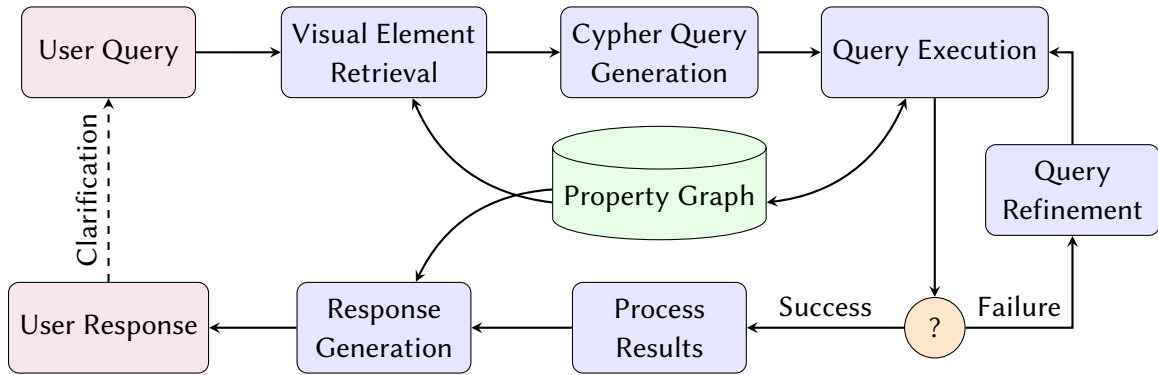


Figure 3.7: Architecture of the proposed system.

utilizes the property graph database described in Section 3.3 to retrieve relevant symbols based on visual characteristics, coupled with a language model for query interpretation and response generation.

When a user submits a query (e.g., “What is the symbol which has a circle and an exclamation mark in it?”), the system processes it through the following steps:

1. **Knowledge Retrieval:** All relevant visual element names, properties, and visual synonyms are retrieved from the property graph (e.g., “circle”, “exclamation-mark”).
2. **Query Translation:** The user query, alongside retrieved element names and visual synonyms, is passed to an LLM. The LLM interprets the natural language query and translates it into a structured Cypher query as seen in the following example:

```

1 MATCH (symbol {type: "symbol"})-[:RELATED*]->(element1)
2 WHERE element1.id CONTAINS "exclamation-mark" OR element1.id CONTAINS "
   exclamation-point"
3 MATCH (symbol)-[:RELATED*]->(element2)
4 WHERE element2.id CONTAINS "circle"
5 RETURN DISTINCT symbol.id

```

3. **Query Execution:** The generated Cypher query is executed against the property graph database to retrieve visual element sub-trees and further information such as functional knowledge for all matching symbols.
4. **Query Refinement:** If the query execution fails – either due to syntax errors or no matching results – the system implements an adaptive retry mechanism:
 - (a) The query and error message are fed back to the LLM.
 - (b) The LLM either corrects syntax errors or relaxes query constraints to broaden the search.
 - (c) A single retry attempt is made with the refined query.
5. **Response Generation:** Finally, the LLM processes the query results to generate an appropriate response:
 - (a) If an unambiguous match is found, the system provides a definitive answer.
 - (b) If multiple potential matches exist, the system formulates a clarification question to disambiguate.
 - (c) If no results are returned or none match the original query criteria, the system informs the user about this and attempts to make an educated guess based on available knowledge.

This example demonstrates how the system leverages both structured knowledge and natural language understanding to interpret ambiguous queries and provide contextually relevant information.

For the language model component, Google’s Gemini 2.0 Flash was selected due to its balanced performance characteristics: competitive pricing (\$0.1/M input tokens, \$0.4/M output tokens), low latency (time to first token³ of 0.52s), and high throughput (112.6 tokens/s)⁴. While no official information about the model size is available, analysts estimate it to be significantly smaller than 70 billion parameters⁵.

The combined architecture provides several key benefits: accurate symbol identification through structured knowledge representation, natural explanations through language model capabilities, and reduced hallucinations by grounding responses in the knowledge graph. Additionally, the query refinement mechanism enables the system to handle imperfect initial queries through an iterative improvement process.

To properly evaluate this system’s effectiveness, a diverse collection of realistic user queries is essential. The following section details the approach to gathering both human and synthetic, machine-generated questions, that capture the variety of ways users might inquire about symbols, ranging from purely visual descriptions to semantically rich inquiries.

3.5 Question Collection

To establish a comprehensive evaluation framework for the proposed system, a multifaceted corpus of questions was collected through two complementary approaches: direct

³ Refer to <https://docs.nvidia.com/nim/benchmarking/llm/latest/metrics.html> for a detailed description of common LLM inference metrics (Retrieved 03/01/2025, *archived*).

⁴ <https://openrouter.ai/google/gemini-2.0-flash-001> (Retrieved 03/03/2025, *archived*)

⁵ <https://x.com/ArtificialAnlys/status/1867292015181942970> (Retrieved 03/03/2025, *archived*)

sampling from human participants and algorithmic generation using LLMs.

3.5.1 Human Questions

A total of six human participants were recruited from the local university campus without compensation. All participants were North American, aged between 21-29, and in possession of a driver’s license. 20 symbols were hand-picked from the dataset, with a deliberate bias towards safety-critical symbols and symbols from the tell-tale category (i.e., symbols which would appear as lights on the dashboard). This bias was implemented by selecting more symbols from ISO 2575 Annex K and L (“Active and passive safety systems” and “Security”). The symbols are shown in Figure 3.1 in the previous Section 3.1.1. Each participant was presented with all 20 symbols and asked to formulate a question about the meaning of each symbol. Participants were instructed to phrase their questions as they would when speaking to a voice assistant (e.g., Google Assistant or Siri) on their phone, keeping their queries concise. In total, 120 human questions were collected. Prior to use in the experiments, all human questions were proof-read to correct minor typing mistakes. Example questions from human participants demonstrate the diversity of potential queries to be answered in an in-the-wild environment:

- **Visual-focused:** “Why is there a symbol with two cars, a star-like symbol, and an arrow pointing from the rear car towards the other car?”
- **Function-oriented:** “I see a car and lock symbol. What does it mean?”
- **Context-driven:** “Should I be concerned when a person wearing a seatbelt with an X symbol appears on my dashboard?”

- **Action-seeking:** “Does the turtle in a circle symbol mean anything significant, and should I do anything about it?”

The rationale behind this question collection approach was twofold: First, focusing on safety-critical symbols and telltales aligns with research by Friedrich et al., who demonstrated that warning and status icons typically have the highest semantic distance, making them the most challenging for users to interpret without assistance [41]. Second, regarding participant sampling, six participants were deemed sufficient for question collection, as this sample size is in line with Nielsen’s finding that approximately five users can uncover the majority of usability patterns in formative research [103].

3.5.2 Synthetic Question Generation

Additionally, ground truth of visual elements from the dataset was used to generate synthetic questions. Five different LLMs (Claude 3.5 Haiku, Gemini 2.0 Flash, GPT-4o mini, Llama 3.3 70B Instruct, and Qwen2.5 72B Instruct) were provided with the ground truth visual element decomposition, including the visual synonyms for the individual elements. The LLMs were instructed, in the same way as human participants, to ask questions about the visuals of the symbol. They were additionally encouraged to come up with their own synonyms for some of the elements, increasing the variability in the resulting questions. The full text of the prompt utilized for the synthetic question generation is provided in Appendix A.1. LVLMs were not used for this purpose, as Fletcher et al. has shown that these models struggle with properly interpreting abstract icon images [15]. Where models had multi-modal capabilities (Gemini 2.0 Flash, GPT-4o mini), only text input was utilized. Out of a targeted 600 synthetic questions (five times the size of the human question set) about randomly sampled symbols, a total of 533 were generated. The reason for this dis-

crepancy lies in models not honoring the instructions at all times. While the structured outputs feature was utilized, which allows to enforce a JSON output adhering to a certain schema, the implementation and support for schema validation differs. OpenAI models do not support the `minItems` and `maxItems` keywords, used to ensure a certain number of array elements⁶, and Anthropic's models (i.e., the Claude family of models) do not support structured outputs at all, resulting in a mismatch between the desired and obtained amount of questions.

⁶ "Structured Outputs – OpenAI API" (Retrieved on 03/01/2025, *archived*)

Chapter 4

Experimental Setup

This chapter outlines the comprehensive experimental framework developed to evaluate the proposed symbol decomposition and reasoning system. The following subsections describe the curated dataset of consumer road vehicle symbols, the selection of state-of-the-art baseline models, and an ablation study designed to isolate the contribution of the graph-based knowledge representation. Additionally, both automatic and human evaluation metrics are detailed, employed to thoroughly assess the system’s performance and robustness.

4.1 Datasets and Symbols

As discussed in Section 3.1.1, only annexes with relevance to consumer road vehicles were included in this research. These annexes reduced the space of symbols by 90, decreasing the total number to 296. On average, each symbol in the dataset contained 2.85 semantic groups, and 5.78 basic visual elements.

Figure 4.1 shows a plot of the property graph contained in the Neo4j database after de-

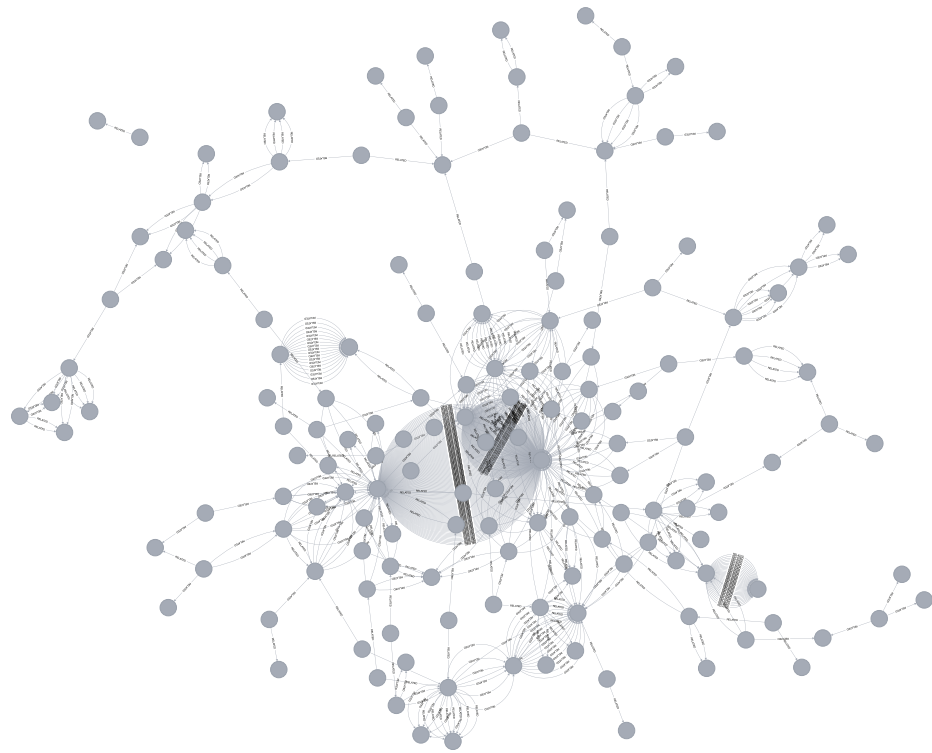


Figure 4.1: Property graph of decomposed symbols, visualization restricted to Annex A of ISO 2575 and limited to 750 relationships.

composition, labeling, and translation into a graph-based representation, as laid out in Section 3.1, 3.2, and 3.3; limited to Annex A for brevity. Notable are the clusters in the center of the graph, which represent edges between line, light, and lamp nodes.

4.2 Baseline Systems

At present, to the best of available knowledge and research efforts, no knowledge bases or other systems exist that can decompose symbols or other abstract imagery into their individual elements for reasoning over those elements. Additionally, no systems are specialized in answering questions about the visual features of abstract imagery, such as symbols. Therefore, recent LLMs were selected as the most suitable baseline for comparison, relying on the inherent knowledge and the extensive training data these models are trained on to provide a comparison to the proposed system. VLMs were not used, as previously discussed in Section 3.5.2.

Author	Name	Year	Size
Anthropic	Claude 3.5 Haiku	2024	?
Google	Gemini 2.0 Flash	2025	~70B ¹
OpenAI	GPT-4o mini	2024	~40B ²
Meta	Llama 3.3	2024	70B
Qwen	Qwen2.5	2024	72B

Table 4.1: Summary of baseline models used.

Table 4.1 shows all baseline models used, as well as their model size in number of parameters, if known. Sizes for Gemini 2.0 Flash and GPT-4o mini are numbers estimated by AI analysts and are only provided for reference. To access the baseline models in a unified

¹ See footnote in Section 3.5.2.

² https://x.com/oscarle_x/status/1874516567381487724 (Retrieved 03/03/2025, *archived*)

way, the OpenRouter³ platform was used. OpenRouter acts as an API aggregator, simplifying integration compared to managing individual APIs and offering a cost-effective alternative to running larger open-source models like Llama or Qwen locally.

All baseline models were instructed to function as in-vehicle assistants that help users with visual questions about symbols. Provided with a user question, models were prompted to respond with either a confident answer (providing a concise explanation) or a request for clarification (when multiple interpretations were possible). Models were additionally provided with five example question-answer pairs. The prompt emphasized clear, concise language and strict adherence to the output format, ensuring consistent interaction across all models. The full text of the prompt utilized for question answering is provided in Appendix A.2.

4.3 Ablation Study

To evaluate the contribution of the property graph-based knowledge representation to overall system performance, a standard retrieval-augmented generation (RAG) approach was implemented as an ablation condition. This experiment was conducted to determine whether conventional document retrieval with large language models could achieve comparable results to the proposed graph-based system.

The RAG pipeline was implemented using the Unstructured⁴ Python library for document processing and the Sentence Transformers⁵ library for semantic retrieval. The ISO 2575 standard document was processed with Unstructured’s “by_title” chunking strategy, creating chunks of approximately 2000 characters that preserved document structure. Em-

³ <https://openrouter.ai/>

⁴ <https://github.com/Unstructured-IO/unstructured>

⁵ <https://github.com/UKPLab/sentence-transformers>

beddings were generated using the Snowflake Arctic Embed L v2.0 model, which created vector representations of both document chunks and user queries. At the time of writing, this model was among the top 10 models in the Huggingface Massive Text Embedding Benchmark⁶ (MTEB) benchmark.

For each question, the top three most relevant chunks were retrieved based on cosine similarity between the query embedding and chunk embeddings. To provide comprehensive multi-modal information, corresponding page images were extracted from the PDF document using the pdf2image library at 150 DPI. Both the textual context and page images were then provided to the language models along with the user’s question. This process aimed to mimic the way OpenAI’s models search and utilize PDF documents whenever they are uploaded through the ChatGPT web-interface⁷.

The experiment was conducted using the human questions dataset to facilitate direct comparison with the main study results. Due to the multi-modal nature of the task (requiring image processing capabilities), certain baseline models were replaced with their multi-modal variants:

- Claude 3.5 Sonnet replaced Claude 3.5 Haiku
- Llama 3.2 90B Vision Instruct replaced Llama 3.3 70B Instruct
- Qwen 2.5 VL 72B Instruct replaced Qwen 2.5 72B Instruct

It should be noted that Claude 3.5 Sonnet and Llama 3.2 90B Vision Instruct are considered more capable than their counterparts in the main experiments, potentially favoring the ablation condition for these specific models. All models were accessed through the OpenRouter API to maintain consistency with the baseline evaluation methodology.

⁶ <https://huggingface.co/spaces/mteb/leaderboard>

⁷ <https://platform.openai.com/docs/assistants/tools/file-search> (Retrieved 03/10/2025, archived)

4.4 Evaluation Metrics

The evaluation strategy employed automatic metrics as well as human assessments to examine both quantitative and qualitative dimensions of system performance. Automatic evaluation utilized established NLP metrics and recent LLM-as-a-Judge based approaches, while human evaluation ensured insights into practical user-perceived accuracy and clarity. These complementary methods collectively ensured a comprehensive analysis of the proposed system. Given the inherently open-ended and subjective nature of question-answering tasks, purely automatic metrics can be inadequate in capturing nuanced aspects of response quality. Thus, human evaluation was incorporated to complement automatic measures, providing insights into how clearly and accurately the systems’ explanations align with human judgment. This combined evaluation approach ensured the assessment remained robust and aligned with real-world usability considerations.

4.4.1 Automatic Evaluation

Multiple complementary metrics were employed in order to assess the quality of the generated answers. *LLM-as-a-Judge* [104, 105] leverages LLMs to evaluate responses based on criteria such as accuracy, helpfulness, and safety without requiring reference answers. For the experiments in this thesis, three LLMs (Llama 3.3, Gemini 2.0 Flash, and GPT-4o mini) were used to directly assess recall and precision for each answer, resulting in an averaged “LLM Committee” score. The full text of the prompt utilized for LLM-based scoring is provided in Appendix A.3. *BERTScore* [106] computes token-level similarity between candidate and reference explanations using contextual embeddings, capturing semantic similarity better than lexical approaches. *METEOR* [107] balances precision and recall while accounting for synonyms, stemming, and word order. *ROUGE-1* [108] measures

unigram overlap between generated explanations and references, while *ROUGE-L* identifies the longest common subsequence, capturing word order sensitivity and allowing for subsequence matches. This multi-metric approach provides a comprehensive assessment of both semantic accuracy and linguistic quality of the symbol explanations.

Initially, the use of *BLEU* [109] was explored, but it was found unsuitable for the task, as it yielded near-zero scores across all models due to limited n -gram overlap at the sentence level. This limitation is consistent with findings that *BLEU* performs poorly for tasks with high linguistic variability in valid responses [110–112].

4.4.2 Human Evaluation

To complement the automatic metrics, a human evaluation study of 50 question and their respective answers was conducted with six participants. Each participant assessed 25 question-answer pairs within a 20–30 minute session. An approximate Balanced Incomplete Block Design (BIBD, cf. [113]) was employed to determine which samples each participant would rate. Specifically, the goal was to ensure that:

1. Each of the $v = 50$ items (question–answer pairs) was rated by $r = 3$ different participants;
2. Each of the $b = 6$ participants rated exactly $k = 25$ items;
3. The total number of “item–slot” assignments satisfied $v \times r = b \times k$.

Thus, from a purely combinatorial perspective:

$$v \times r = 50 \times 3 = 150 \quad \text{and} \quad b \times k = 6 \times 25 = 150.$$

These conditions guaranteed that each item appeared the required number of times (three) and that each participant rated the required number of items (25). A perfect BIBD further requires that each *pair* of distinct items co-occur in exactly λ blocks, where λ is an integer satisfying the standard BIBD identity of $r(k - 1) = \lambda(v - 1)$. In this setup, plugging in $r = 3$, $k = 25$, and $v = 50$ yields

$$3 \times (25 - 1) = 3 \times 24 = 72, \quad v - 1 = 49, \quad \lambda = \frac{72}{49} \approx 1.469,$$

which is not an integer. Therefore, this design is *not* a strict BIBD but an *approximate* incidence design that meets the practical requirements: every question–answer pair receives exactly three ratings, and each participant sees exactly 25 pairs. There was no way to perfectly balance pairwise co-occurrences under these parameters.

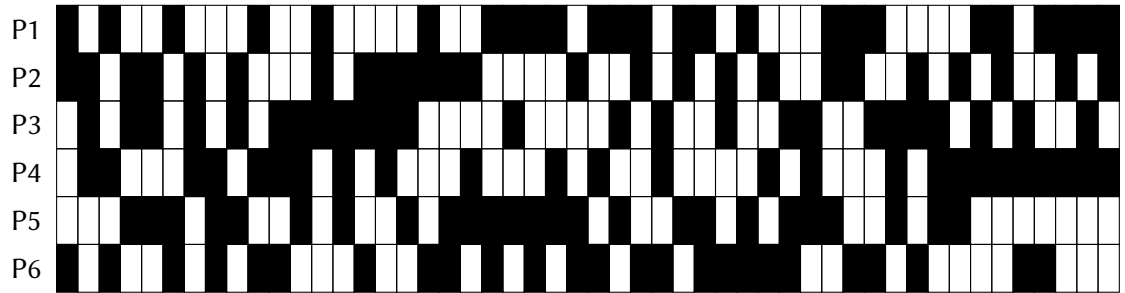


Figure 4.2: Visualization of the 50×6 incidence matrix used for the approximate BIBD. Rows represent participants, and columns represent samples. Filled squares indicate that a particular participant rated that item.

Implementation of the assignment was performed by a backtracking algorithm over a 50×6 incidence matrix, which is shown in Figure 4.2. Additionally, the order of the samples, as well as the order of the responses for each sample was randomized, to offset any potential order effects. During the study, participants were shown the dashboard symbol image, the corresponding question, the ground truth description, and anonymized answers from

different systems. For each explanation, participants rated the overall answer quality on a five-point Likert scale (1 = poor, 5 = excellent), focusing on how accurately and clearly it conveyed the symbol's function.

Chapter 5

Results

In this chapter, the results of the proposed system’s performance are presented and compared with several baseline LLMs. The analysis begins with an illustrative example demonstrating the system’s approach to symbol identification, followed by a detailed evaluation of overall question-answering capabilities using automated metrics. Furthermore, human evaluation results are provided, highlighting the practical utility of the proposed method from a user perspective. The chapter concludes with examinations of performance differences between human and synthetic questions, the effectiveness of clarification questions, insights gained from an ablation study assessing the impact of the structured knowledge representation, and an analysis of the failure modes of the proposed system and the baseline models.

5.1 Illustrative Example

Table 5.1 presents a representative example of the answers given by the proposed system compared to the baseline LLMs for the user question “What does the symbol of a

horizontal line and three wavy arrows pointing up mean?”, accompanied by their respective LLM-as-a-Judge committee scores¹. The question asked about the “interior-heating”

Model	Answer (abbreviated)	F1
Proposed System	The symbol indicates the interior heating system.	0.91
GPT 4o-mini	The symbol represents the heated steering wheel function.	0.42
Claude 3.5 Haiku	This symbol typically indicates a heated steering wheel function.	0.42
Gemini 2.0 Flash	This symbol indicates the rear window defroster is active.	0.36
Qwen 2.5 72B	The symbol indicates the air conditioning or climate control system is active.	0.08
Llama 3.3 70B	The symbol [...] typically indicates the Rear Fog Lamp.	0

Table 5.1: Answers and LLM Committee F₁-scores for the human question: “What does the symbol of a horizontal line and three wavy arrows pointing up mean?”.

symbol (ISO 2575: Symbol D.03), previously shown in the first row of Figure 3.1. The proposed system translated this question into the following Cypher query:

```

1 MATCH (symbol {type: "symbol"})-[:RELATED*]->(element1)
2 WHERE element1.id CONTAINS "line"
3 MATCH (symbol)-[:RELATED*]->(element2)
4 WHERE element2.id CONTAINS "arrow-up" OR element2.id CONTAINS "arrows"
5 RETURN DISTINCT symbol.id

```

This query retrieved a total of five potentially relevant symbols, including “heated-steering-wheel”, “interior-heating”, “head-lamp-leveling-control”, “turn-signals”, and “turn-signals-trailer”. However, by accurately interpreting the user’s description of a “horizontal line” and “wavy arrows pointing up”, the system correctly identified “interior-heating” as the intended symbol, achieving a significantly higher Precision (0.96), Recall (0.86), and F₁-score (0.91) than all baseline models.

¹ From here on referred to as “LLM Committee” scores.

In contrast, the baseline LLMs, relying primarily on their pre-trained knowledge without explicit symbol decomposition, exhibited significantly lower performance. GPT-4o mini and Claude 3.5 Haiku incorrectly identified the symbol as representing the “heated steering wheel”, highlighting their susceptibility to visually similar but semantically distinct symbols. This error shows a fundamental limitation in processing subtle visual variations within the realm of abstract symbol representations: While numerous symbols share common visual characteristics, the absence or presence of specific elements can significantly impact their interpretation. In this case, the difference between the “interior-heating” and the “heated-steering-wheel” symbol lies in the omission of the horizontal line, and the addition of a steering wheel. The other baseline models showed even weaker performance, demonstrating that this symbol, and by extension the domain of understanding implicit visual descriptions in questions about symbols, poses significant challenges. Gemini 2.0 Flash answered with rear window defroster, Qwen 2.5 with the air conditioning, both of which are at least somewhat related to heating, while Llama completely failed. These results, consistent with broader trends observed throughout the evaluation, emphasize the importance of structured knowledge representation and explicit reasoning capabilities for accurate and reliable symbol interpretation, especially in safety-critical contexts.

5.2 Question Answering

Figure 5.1 presents the evaluation results for question answering, comparing the proposed system against several baseline models across multiple metrics. The proposed system demonstrated better performance across all metrics compared to the baseline models. Specifically, the proposed system achieved an LLM Committee F_1 -score of 0.693, considerably higher than the best-performing baseline, Qwen 2.5 72B, which scored 0.256. This

substantial performance difference was consistent across most evaluation metrics, including ROUGE-1, ROUGE-L, and METEOR. BERTScore showed a less pronounced, though still significant, difference between the proposed system and the baselines. This observation aligns with prior research by Fletcher et al., which found that the BERTScore metric tends to exhibit smaller performance gaps in the domain of abstract symbol understanding [15]. Detailed results for all models and metrics are provided in Table B.1 and broken down by synthetic/human questions in Tables B.2 and B.3 in Appendix B.

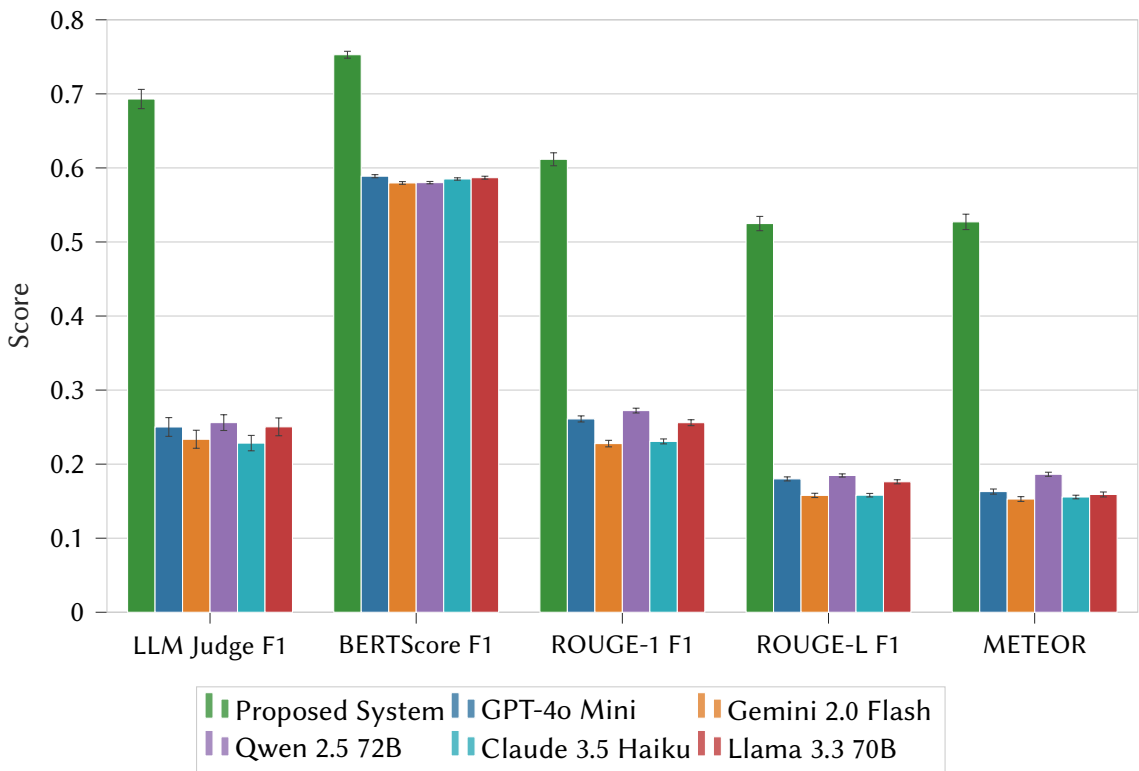


Figure 5.1: Performance comparison of the proposed system against the five baseline LLMs across different evaluation metrics. Error bars show $\pm 1 SE$.

An omnibus test [114] revealed that the data were not normally distributed. Thus, a non-parametric Friedman test was performed to assess the effect of model type on the LLM Committee F_1 -scores. The test revealed a statistically significant effect ($\chi^2(5) =$

107.549, $p < .001$), with Kendall's $W = 0.398$, indicating a moderate to strong effect size [115]. Subsequent post-hoc analysis, employing Wilcoxon Signed-Rank tests with Holm-Bonferroni correction, confirmed that the proposed system outperformed all baseline models across all pairwise comparisons (all $p < .001$).

The higher performance of the proposed system can be attributed to its knowledge-grounded architecture. This allowed for more precise query interpretation and retrieval of relevant information, leading to accurate and informative answers, as shown in the previous Section 5.1. The baseline LLMs, lacking this explicit symbol knowledge, often struggled to correctly interpret the user's query or provide relevant information, resulting in lower scores across all metrics. The use of Cypher query refinement, described in section 3.4, additionally contributed to the robustness of the proposed approach, enabling it to recover from initial query failures and improve retrieval accuracy.

5.3 Human Evaluation

To assess the real-world utility of the proposed system from a user-centered perspective, a human evaluation was conducted using the same twenty ISO 2575 symbols shown in Figure 3.1. Human raters were asked to evaluate the quality of answers provided by each model on a 5-point Likert scale.

Figure 5.2 displays the average ratings across all models. The proposed system achieved a mean rating of 4.56 on the 5-point Likert scale, while baseline models received considerably lower scores ranging from 2.32 to 2.51 (GPT-4o Mini: 2.35, Gemini 2.0 Flash: 2.51, Qwen 2.5 72B: 2.32, Claude 3.5 Haiku: 2.33, Llama 3.3 70B: 2.33).

To determine whether these differences were statistically significant, a Kruskal-Wallis H

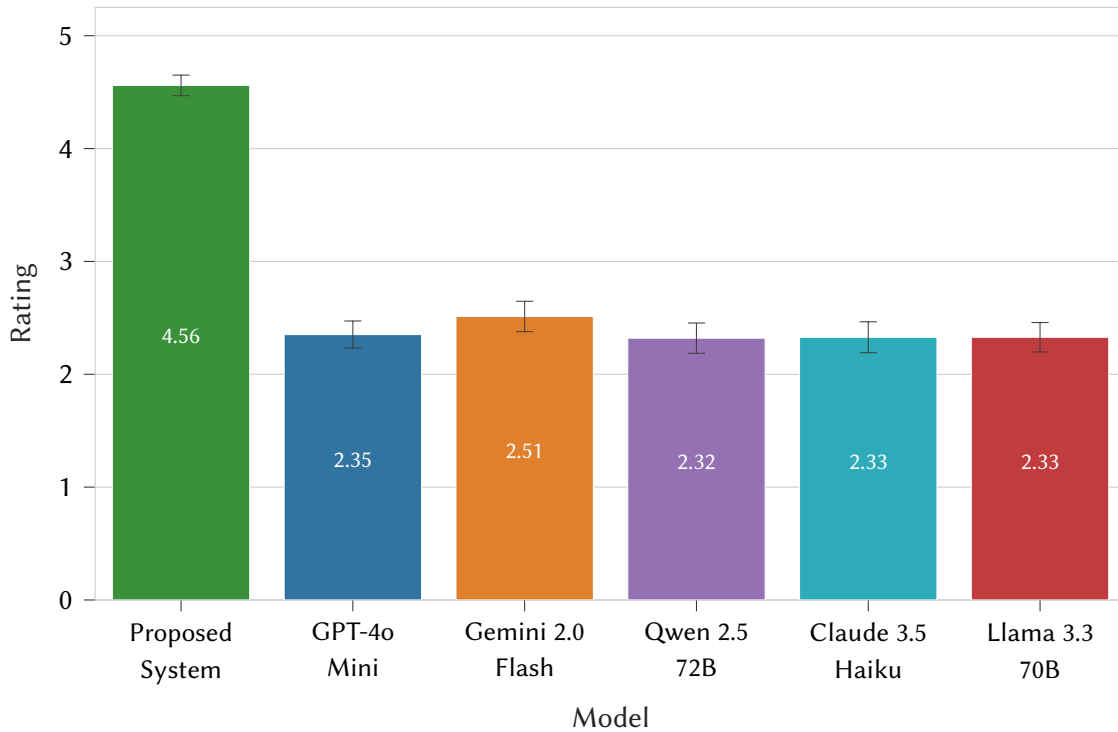


Figure 5.2: Mean human evaluation ratings on a 5-point Likert scale. Error bars show ± 1 SE.

test was conducted, appropriate for ordinal Likert scale data. The test revealed a significant effect of model type on human ratings ($H(5) = 183.53, p < 0.001$). Subsequent pairwise comparisons using Dunn’s post hoc test confirmed that the proposed system was rated significantly higher than all baseline models (all $p < 0.001$).

Figure 5.3 provides a more granular view of performance across the 20 tested ISO 2575 symbols. The heat-map reveals that the proposed system was rated higher than baseline models for 19 of the 20 symbols, with ratings typically between 4 and 5. The most notable exception was symbol K.10 (“Low Tire Pressure”), where baseline models achieved comparable or slightly higher ratings. Analysis of the questions for this particular symbol revealed that the collected human questions frequently used visual analogies not present in the symbol decomposition property graph (e.g., describing the symbol as a “wide circle with treads” or a “partially open vase”). Since the evaluation sample contained multi-

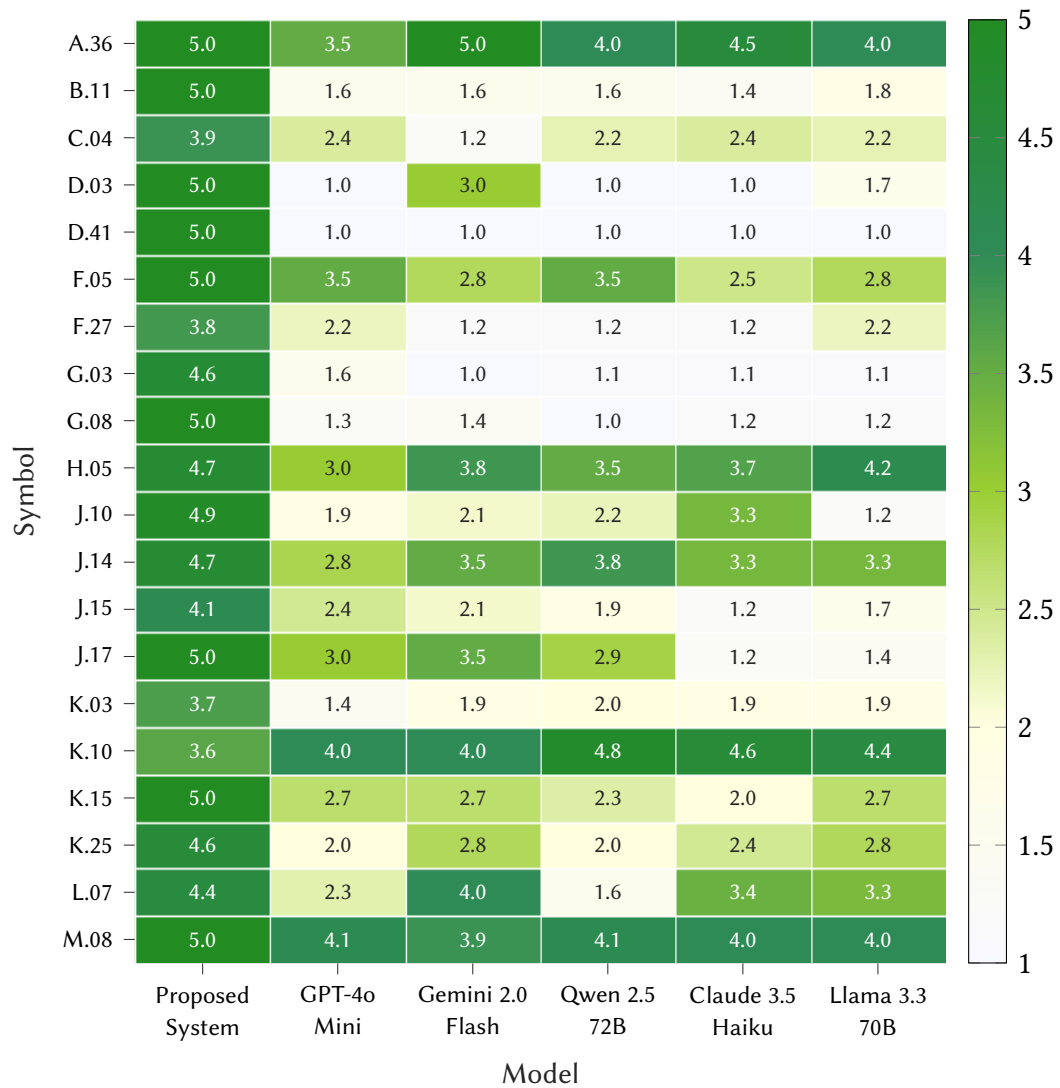


Figure 5.3: Heat map of human evaluation ratings across ISO 2575 symbols.

ple instances of these questions with unexpected analogies, the knowledge graph queries failed to identify the intended symbol. This resulted in the system generating clarification questions rather than direct answers, leading to lower ratings for this specific symbol.

The performance advantage across diverse symbols suggested robustness in the hierarchical decomposition approach described in Sections 3.1 and 3.2. Even for symbols where baseline models performed reasonably well (such as symbols A.36, H.05, J.14, and M.08), the proposed system maintained higher ratings, indicating better perceived accuracy and helpfulness. The exception case (K.10) highlighted a limitation of the current implementation – the system’s sensitivity to the completeness of the symbol decomposition in the knowledge graph when handling novel visual analogies not anticipated during the manual annotation process described in Section 3.2.

These human evaluation results complement the automated metrics presented earlier, providing a user-centered perspective that confirmed the effectiveness of the proposed system’s approach to interpreting and responding to symbol-related queries. The consistent pattern of higher ratings suggest that the property graph-based representation combined with LLM-based query translation offers advantages over approaches that rely solely on pre-trained knowledge.

5.4 Performance by Question Set

To investigate the effect of the question source on the model scores, performance on questions collected from humans was compared against synthetically generated questions. Figure 5.4 illustrates the mean LLM Committee F_1 scores for each model across both question sets.

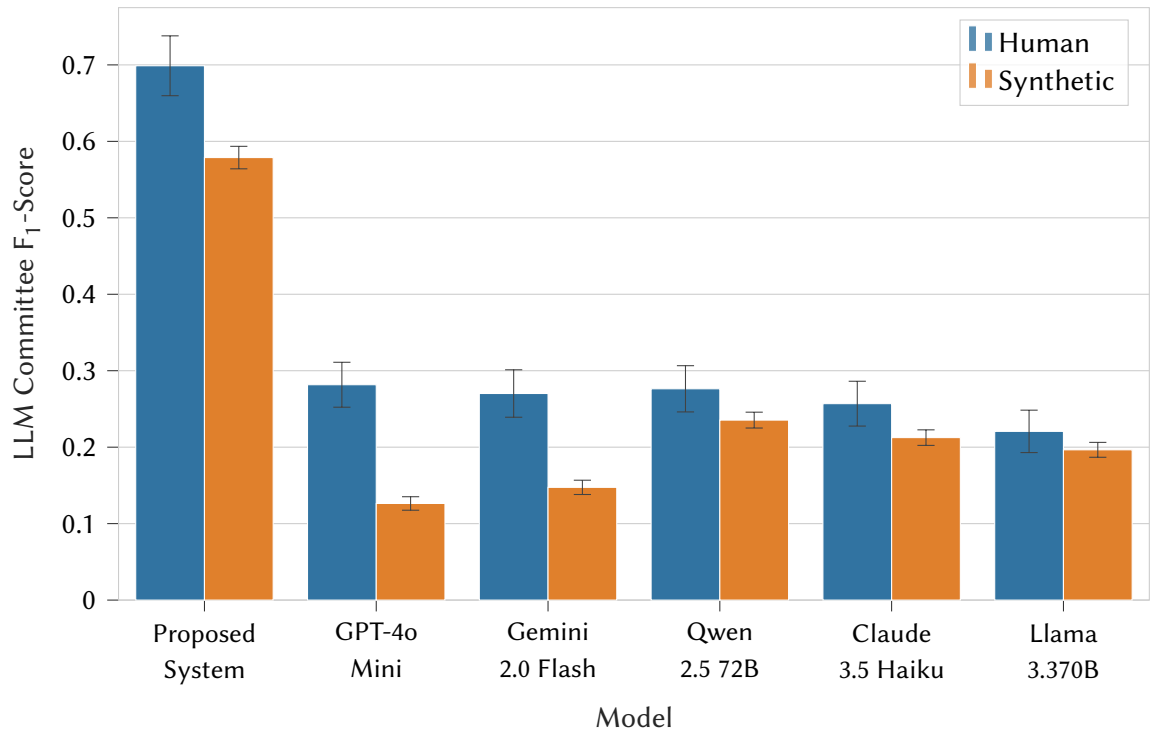


Figure 5.4: Comparison of the LLM Committee F₁-scores between human and synthetic questions. Error bars show $\pm 1 SE$.

The results revealed a surprising, yet consistent trend across all models: performance on human questions exceeded that on synthetic questions. The proposed system consistently achieved higher F_1 scores on human questions ($F_1 = 0.736$) than on synthetic questions ($F_1 = 0.684$). This pattern was also observed for all baseline models. For example, GPT-4o Mini's F_1 -score was 0.339 on human questions, decreasing to 0.225 on synthetic questions. Gemini 2.0 Flash exhibited a similar reduction from 0.325 to 0.212. Descriptive statistics confirmed this overall trend. The mean F_1 score across all models and metrics was higher for human questions ($M = 0.381$) than for synthetic questions ($M = 0.313$). The standard deviation was also slightly higher for human questions ($SD = 0.355$) compared to synthetic questions ($SD = 0.300$), indicating somewhat greater variability in performance on the human-generated data. A Mann-Whitney U test comparing overall scores on both datasets, chosen due to non-normal data distribution (confirmed by an omnibus test) and unequal sample sizes, revealed a statistically significant difference ($U = 657803.0, p < 0.005$).

Further analysis, using pairwise Mann-Whitney U tests to examine the effect of question source for each model individually, revealed a significant effect for the proposed system ($U = 25148.5, p < 0.001$), GPT-4o Mini ($U = 14841.0, p < 0.005$), and Gemini 2.0 Flash ($U = 17109.5, p < 0.05$), indicating that the decrease in F_1 scores between human and synthetic question sets was statistically significant for these models. However, for Qwen 2.5 72B ($U = 21996.0, n.s.$), Llama 3.3 70B ($U = 12949.0, n.s.$), and Claude 3.5 Haiku ($U = 22859.0, n.s.$), the effect was not statistically significant.

This performance difference suggests that the synthetic questions, while intended to replicate human queries, presented a different, and potentially more challenging, distribution of linguistic expressions and question structures. This finding highlights the importance

of evaluating question-answering systems on a variety of question sets, including both natural, human-generated queries and systematically constructed synthetic ones. Furthermore, the lack of significant differences for some baseline models may indicate that these models were less sensitive to the specific distinctions between these two input types, possibly due to their overall lower performance levels. The differences between synthetic and human questions will be qualitatively investigated in the following Section 5.5.

5.5 Analysis of Questions Structure

This section examines the performance difference between human-created and synthetically generated questions. Twenty questions were randomly selected from each set for comparison and analyzed manually.

Human-collected questions consistently employed intuitive, object-based comparisons when interpreting unfamiliar symbols, aligning with Gentner and Smith’s findings on analogical reasoning [7]. For instance, humans combined basic shapes with familiar analogies (e.g., “What does the symbol that looks like a sideways ice cream cone over a wavy line mean?”). Synthetic questions, however, provided detailed descriptions based primarily on visual elements (e.g., “What does this dashboard light with two diagonal lines and one horizontal line along with a letter ‘P’ mean?”). This distinction reflects the semantic gap described by Bahmanyar et al. between visual elements and their interpretations [4, 42]. The symbol decomposition methodology proposed in Section 3.1 attempted to systematically capture this hierarchy of visual elements and semantic units.

Synthetic questions also exhibited greater length (averaging 22.2 words versus 16.8 for human questions) and syntactic complexity, often containing embedded clauses and mul-

multiple descriptive elements. For example, synthetic questions frequently enumerated multiple components (e.g., “Why does my dashboard show a light beam symbol with five horizontal lines, a wiper icon with a diagonal line and a dot...”), while human questions were more concise (e.g., “What does the oil pan with a thermometer inside it mean?”). This increased verbosity likely resulted in higher cognitive processing demands for the models, potentially causing confusion in symbol identification. This is consistent with findings by Johnson et al., who showed that when questions become more syntactically and semantically complex – requiring the model to track multiple objects, relationships, and attributes – model performance degraded noticeably [116].

Additionally, there were differences in how each question type prioritized information. Synthetic questions often exhaustively listed visual elements, sometimes including non-essential features. Human questions, however, focused on distinctive features that make symbols recognizable (e.g., “What does the symbol with a star above and an arrow below pointing left with two cars mean?”). This approach aligns with findings from Isherwood et al. and Mcdougall et al. on icon interpretability [38, 117], suggesting that identifying distinctive features is more valuable than exhaustive descriptions.

The proposed graph-based representation (Section 3.3) captured both fine-grained visual elements and higher-level semantic groups. The performance advantage of human questions suggests that models generally utilize semantic groupings more effectively than detailed visual decompositions. For example, the proposed system returned an uncertain response on 21.2% of all questions in the synthetic question set, compared to only 16.7% on the human question set. This also supports Chi and Dewi’s research on how familiar reference points influence symbolic interpretation [30]. While the hierarchical labeling system introduced in this work already accommodated both levels, future refinements

could strengthen connections between them to better support intuitive descriptive patterns.

The performance discrepancy furthermore has important implications for VQA systems in general. Though synthetic question generation offers advantages in terms of scalability and cost, current approaches have limitations. The findings highlight the challenges identified by Fletcher et al. regarding VLMs' difficulties with abstract iconography [15]. Improving synthetic generation to better mimic human question patterns – using familiar analogies, focusing on distinctive features, and employing simpler structures – could potentially reduce this performance gap.

More broadly, this discrepancy highlights a key challenge in AI evaluation: synthetic benchmarks may not accurately reflect real-world usage patterns. For safety-critical applications, as often present in the automotive industry, this distinction becomes particularly relevant given the serious consequences of symbol misinterpretation [1]. As such, future work may focus on developing more natural synthetic question generation methods and include more diverse human participants to capture cultural variations in how people describe symbols.

5.6 Use of Clarification Questions

The analysis of when models required clarification provides additional insights into system behavior. Figure 5.5 illustrates the distribution of clarification questions across models as a percentage of total clarifications observed. As shown in the pie chart, the baseline models exhibited varying tendencies to ask clarification questions, with GPT-4o Mini (28.3%), Llama 3.3 70B (23.3%), and Gemini 2.0 Flash (21.9%) requesting clarifications most

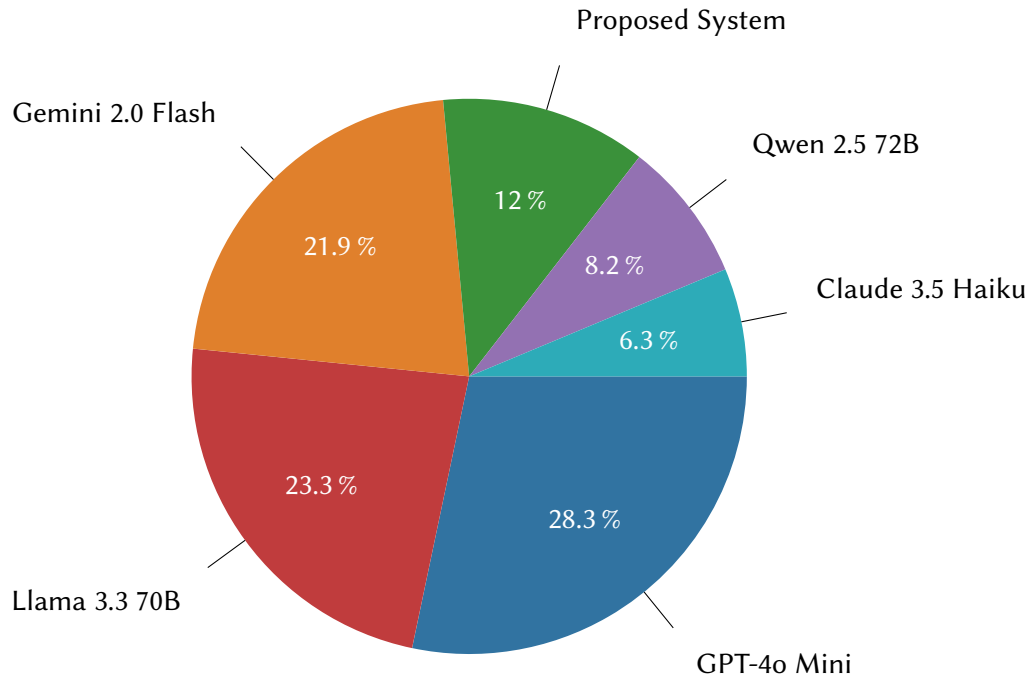


Figure 5.5: Proportion of clarification questions generated by each model.

frequently. In contrast, the proposed system accounted for only 12.0% of all clarifications, with Qwen 2.5 72B (8.2%) and Claude 3.5 Haiku (6.3%) requiring the fewest clarifications.

It is important to note that this distribution should not be interpreted as a direct performance indicator. In the proposed system, clarification questions were systematically generated when multiple potential symbol matches were identified in the knowledge graph, as described in Section 3.4. This represented a deliberate design choice rather than model uncertainty. Baseline models, lacking this structured knowledge representation, may have asked clarifications for different reasons, including general uncertainty or lack of knowledge about abstract symbols. An example of the difference in precision of clarification questions between the models is given in the following Figure 5.6.

Figure 5.7 presents the relationship between model performance and whether clarification



Copied by Nicholas Dehnen with the permission of the Standards Council of Canada (SCC) on behalf of ISO. The standard can be purchased from the national ISO member in your country or the ISO Store. Copyright remains with ISO.

Clarification Questions

Proposed System: “Several symbols relate to driving ahead. Is the symbol showing a car with skid marks, a car between dashed lines, a car possibly crashing, or a night scene?”

Gemini 2.0 Flash: “To best assist you, I need more information about the symbol. Could you describe the symbol in more detail, or tell me what the symbol looks like?”

GPT 4o-mini: “Can you describe any specific features of the symbol, such as colors or additional elements?”

Claude 3.5 Haiku: “I cannot confidently identify the specific symbol you are describing. Could you provide more details about what the symbol looks like, such as its color, shape, or other visual elements?”

Figure 5.6: Clarification questions asked by the proposed system and three baseline models for the user question: “**This symbol looks like I should be concerned about driving ahead. Is that true?**”. The question asks about the “Stability Control System” symbol (J.14) shown on the left. Only models which responded with clarification questions are included.

was required. Across all models, performance was higher when no clarification was required. The proposed system maintained a higher F_1 scores in both scenarios, though the difference was much more pronounced when clarification was not required. This aligned with the system design, where clarification questions are asked specifically when symbol identification was ambiguous.

It should be noted, however, that these metrics aren’t directly interpretable in terms of performance. Analyzing the effectiveness of clarification questions requires a thorough and methodological approach, and was not conducted as part of this thesis. The comparative performance on questions requiring clarification was influenced by several factors. First, the sample sizes differed substantially between models, with baseline models asking clarification questions in different scenarios than the proposed system. Second, the proposed system’s clarification questions were triggered by specific conditions in the knowl-

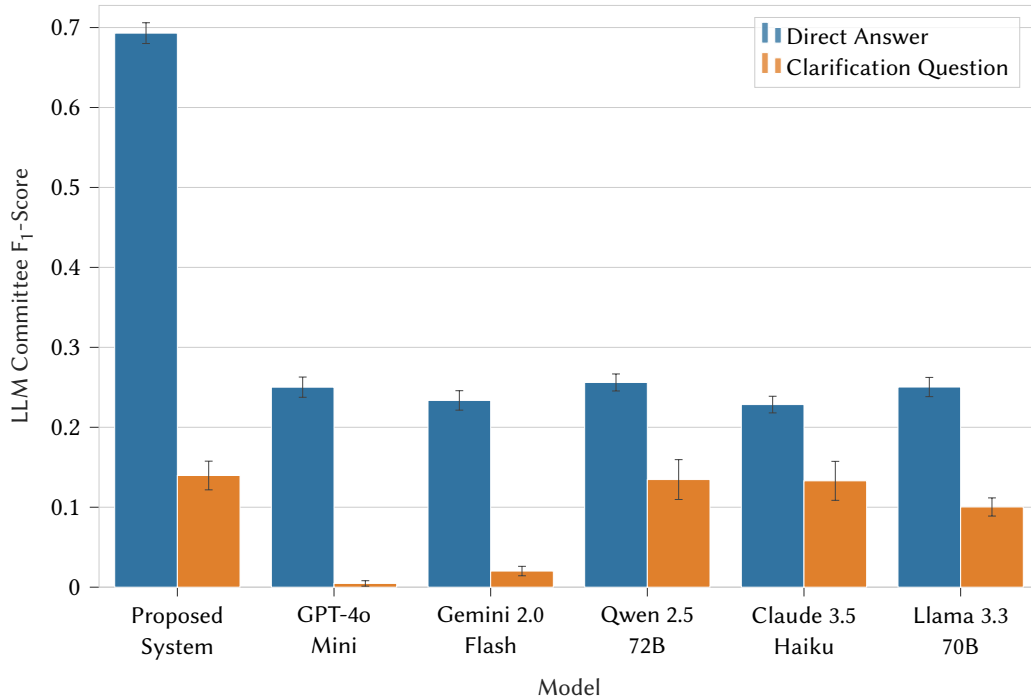


Figure 5.7: Comparison of LLM Committee F₁-scores between direct answers and clarification questions Error bars show $\pm 1 SE$.

edge retrieval process, while baseline models may have asked clarifications due to general uncertainty about abstract symbols. This fundamental difference in when clarifications were triggered limits the validity of direct performance comparisons in these cases.

5.7 Ablation Study

An ablation study was conducted to assess the relative contribution of the property graph-based knowledge representation to overall system performance. This study implemented a conventional retrieval-augmented generation (RAG) approach that replaced the structured knowledge graph with direct document retrieval, providing relevant text chunks and images from the ISO 2575 standard to the baseline models. Figure 5.8 presents a comparison of model performance between the regular baseline approach and the RAG-based

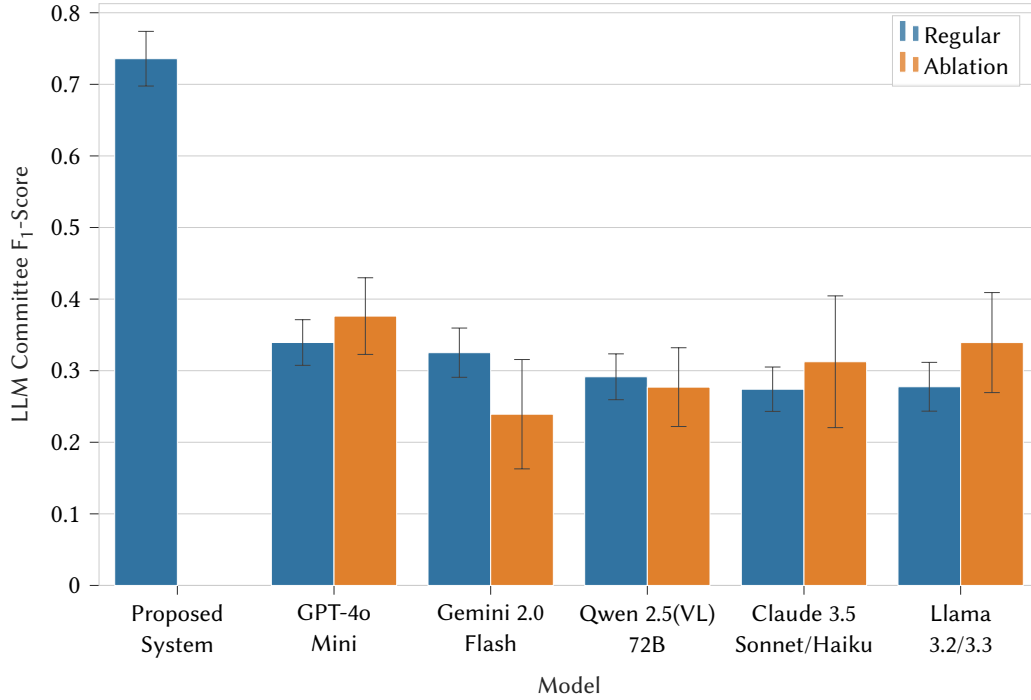


Figure 5.8: Comparison of LLM Committee F_1 -scores between proposed system and RAG-based ablation. Error bars show $\pm 1 SE$.

ablation condition. The results revealed mixed effects across different models. GPT-4o Mini showed slight improvement in the ablation condition ($F_1 = 0.38$ vs. 0.34), while Gemini 2.0 Flash exhibited decreased performance ($F_1 = 0.24$ vs. 0.32). Qwen and Claude models showed minimal differences between conditions. Statistical analysis using Mann-Whitney U tests revealed no significant differences between regular and ablation conditions for any combination of baseline models (all $p > 0.25$). Similarly, the overall comparison across all models yielded non-significant results ($U = 28144.5$, $n.s$), with mean F_1 scores of 0.38 for regular conditions and 0.31 for ablation conditions. These findings suggest that the addition of retrievable context and images did not substantially alter model performance.

Notably, the proposed system maintained substantially higher performance ($F_1 = 0.74$)

than all ablation variants. This performance gap indicates that the structured knowledge representation and reasoning capabilities provided by the property graph approach offer advantages beyond what can be achieved through simple retrieval and context augmentation. A striking difference observed in the ablation study was the marked increase in uncertainty exhibited by the baseline models when provided with additional context. Models in the ablation condition produced uncertain responses or clarification questions at dramatically higher rates: Claude 3.5 Sonnet (85%), Llama 3.2 (84%), Gemini 2.0 Flash (80%), GPT-4o Mini (66%), and Qwen 2.5 VL (64%). This contrasts sharply with the proposed system, which requested clarification in only 6% of cases. This increased uncertainty suggests that while the RAG approach provided relevant information, models struggled to integrate this information effectively to form confident conclusions.

Beyond performance metrics, several practical limitations of the RAG-based approach were identified that would impact real-world applicability:

Computational Cost The RAG-based approach incurred substantially higher computational costs compared to the proposed system. The per-question cost ranged from US\$0.0005 (Gemini 2.0 Flash) to US\$0.03 (Claude 3.5 Sonnet), compared to approximately US\$0.0002 for the proposed system. This represents a cost increase of 2.5× to 150× depending on the model employed.

Latency Response generation latency was significantly increased in the ablation condition, with time-to-first-token ranging from 0.75 seconds (Llama 3.2) to 8 seconds (Claude 3.5 Sonnet), not including the additional time required for complete response generation. The proposed system’s query execution and response generation typically completes within 1 – 2 seconds total.

API Limitations Rate limiting presented a substantial challenge, particularly for Claude 3.5 Sonnet, which exhausted its quota of 40,000 tokens per minute after processing only five questions. This limitation would severely restrict the system’s ability to handle multiple concurrent users in real-world deployment scenarios.

The results of this ablation study highlight a few important implications. First, the absence of significant performance improvements in the RAG-based approach, despite providing models with direct access to relevant document content and images, underscores the importance of structured knowledge representation. Second, the pronounced increase in response uncertainty demonstrates that simply providing more information does not necessarily lead to more confident or accurate responses. Finally, the practical limitations identified – including higher costs, increased latency, and API constraints – demonstrate that the proposed system offers significant advantages in terms of deployability and scalability. These considerations are particularly relevant for in-vehicle applications where responsiveness and reliability are critical requirements.

5.8 Failure Analysis

This section aims to give a brief, qualitative overview of the failure modes experienced in question answering, for both the proposed system, as well as the baseline systems. Failure was defined as receiving a human evaluation score below 3.0 (“Average”) on the 5-point Likert scale used. Three main categories of failures were identified based on the collected data.

There were three instances where all evaluated models, including the proposed system, failed to provide an adequate answer. These cases involved questions that were inher-

ently ambiguous, and misinterpreted or misdescribed basic visual elements, leading to diverse but incorrect interpretations across all systems. An example is detailed in Table 5.2. In this case (Symbol C.04), none of the models correctly identified the symbol as the “Instrument Panel Brightness Adjustment” symbol; instead, they guessed functions like tachometer, engine temperature, or general warnings, all receiving ratings of 2.0 or lower. This suggests that certain human questions about visual configurations may pose a universal challenge to all models, even when a symbol decomposition-based approach is utilized.

In 16 cases, the proposed system provided a satisfactory answer while all baseline models failed. This often occurred questions involving the visual composition of lesser known symbols, captured in the system’s property graph but not readily available or inferable by the generalist baseline models. Table 5.3 illustrates such a case (Symbol J.10), where the proposed system correctly identified the adaptive cruise control failure symbol (receiving a perfect rating of 5.00), whereas baseline models offered generic or incorrect interpretations like TPMS or charging system warnings, all with ratings of 2.00 or lower. This highlights the benefit of using structured knowledge approach for question answering, especially for the visual questions about domain-specific symbols addressed in this work.

Conversely, there was just one instance where the proposed system received a marginally failing score while all baseline models performed above average. This occurred with Symbol K.10 (“Low Tire Pressure”), as documented in Table 5.4. Upon closer examination, the likely cause was the system’s use of a database identifier “tire-failure-malfunction” rather than the actual symbol name that human users would recognize. Although the system’s explanation was technically accurate in content, human evaluators appeared to down-

Question: A gauge with a line and dashed lines arcing appeared. Should I take any action?



Copied by Nicholas Dehnen with the permission of the Standards Council of Canada (SCC) on behalf of ISO. The standard can be purchased from the national ISO member in your country or the ISO Store. Copyright remains with ISO.

Model	Rating	Answer
Proposed System	1.00	This symbol identifies the control that adjusts the brightness of the displays on the instrument panel.
GPT-4o Mini	2.00	Could you please describe the gauge more specifically? Is there anything inside or around it that indicates temperature, pressure, or another measurement?
Claude 3.5 Haiku	1.00	The symbol appears to indicate a measurement that has entered a warning zone. This could be related to engine temperature, oil pressure, or another critical system. You should check the vehicle's manual and monitor the gauge closely. If the warning persists or is accompanied by other warning lights, have your vehicle checked by a professional.
Gemini 2.0 Flash	1.00	This symbol typically indicates information about your vehicle's engine temperature. If the indicator is in the red zone, your engine may be overheating, and you should pull over and allow it to cool down.
Llama 3.3 70B	1.00	The symbol you're describing sounds like a tachometer, which measures engine speed. If the line is in the dashed area, it may indicate that the engine is nearing its maximum recommended speed. You should avoid exceeding the red-line to prevent engine damage.
Qwen 2.5 72B	1.00	This symbol typically indicates the status of your vehicle's speed or tachometer gauge. It may be showing that the gauge is in a limited or diagnostic mode. Please check your vehicle's manual for specific instructions, but generally, no immediate action is required unless accompanied by a warning message.

Table 5.2: Example case: All models performed poorly.

Question: What does the symbol of an exclamation point, a car, and a gauge mean?



Copied by Nicholas Dehnen with the permission of the Standards Council of Canada (SCC) on behalf of ISO. The standard can be purchased from the national ISO member in your country or the ISO Store. Copyright remains with ISO.

Model	Rating	Answer
Proposed System	5.00	This symbol indicates that the adaptive cruise control system has failed or is operating outside of its normal parameters.
GPT-4o Mini	1.67	The symbol represents a warning about a potential issue with the car's engine or vehicle systems, typically indicating that you should check your vehicle's status.
Gemini 2.0 Flash	1.00	This symbol usually indicates a problem with the tire pressure monitoring system (TPMS). It could mean low tire pressure in one or more tires, or a malfunction in the TPMS itself.
Qwen 2.5 72B	2.00	The symbol with an exclamation point, a car, and a gauge typically indicates a vehicle system warning or a need for immediate attention to a vehicle's performance or maintenance issue.
Llama 3.3 70B	1.00	The symbol indicates a vehicle malfunction or issue, often related to the engine or emissions system, and is typically displayed as part of the car's onboard diagnostics system.
Claude 3.5 Haiku	1.00	This symbol typically indicates a serious warning about your vehicle's charging system or battery, suggesting an electrical problem that requires immediate attention.

Table 5.3: Example case: Proposed System performed well, others poorly.

grade the response based on this terminology choice, perceiving it as less accessible and user-friendly than the more natural language descriptions provided by the baseline models. Since the proposed system uses a large language model to generate responses based on knowledge graph results, it introduces inherent variations in output quality and phrasing. Such inconsistencies could potentially be mitigated through future improvements to the response generation, ensuring that technically accurate information is also presented in user-friendly language that meets human expectations for quality responses.

Question: What does the wide circle with treads and an exclamation mark mean, and should I be concerned?



Copied by Nicholas Dehnen with the permission of the Standards Council of Canada (SCC) on behalf of ISO. The standard can be purchased from the national ISO member in your country or the ISO Store. Copyright remains with ISO.

Model	Rating	Answer
Proposed System	2.50	The tire-failure-malfunction symbol indicates that your tire pressure is abnormal or that there is a malfunction in the tire pressure monitoring system. You should be concerned and check your tire pressures.
Llama 3.3 70B	3.50	The symbol you are describing is likely related to tire pressure, indicating that one or more of your tires is underinflated. You should check your tire pressure as soon as possible.
Claude 3.5 Haiku	5.00	This symbol indicates a tire pressure warning. It means one or more of your tires may have low air pressure. You should check your tire pressures as soon as possible and inflate them to the recommended levels to ensure safe driving and prevent potential tire damage.
GPT-4o Mini	4.00	The symbol typically represents a tire pressure warning. It suggests that one or more tires may be under-inflated, which can affect your handling and safety. You should check your tire pressures as soon as possible.
Qwen 2.5 72B	4.50	This symbol typically indicates an issue with the tire pressure. It suggests that one or more of your tires may be underinflated or overinflated, which can affect your vehicle's handling and fuel efficiency. You should check your tire pressure as soon as possible.
Gemini 2.0 Flash	4.00	This usually indicates a tire pressure monitoring system (TPMS) warning. It means one or more of your tires has significantly low pressure, and you should check and inflate your tires as soon as possible.

Table 5.4: Example case: Proposed System performed poorly, others well.

5.9 Limitations

Several limitations of the current approach have been identified through the experimental evaluation and are discussed in the following paragraphs.

Knowledge Engineering Effort The symbol decomposition methodology required significant manual effort to annotate symbols with visual elements and semantic groups. While providing a robust foundation for the knowledge graph, this labor-intensive process would need to be repeated when applying the approach to new symbol sets or standards. This requirement for manual annotation presents a scalability challenge for broader application of the methodology.

Query Generation Challenges Translating natural language to Cypher queries presents technical challenges. Syntax errors occasionally occur, and certain complex queries can be computationally intensive. The system implemented timeouts to address performance concerns, but this can lead to incomplete results for particularly complex queries. The single retry mechanism, while effective in many cases, may be insufficient for especially ambiguous or complex user descriptions.

Demographic Representation The human participants recruited were all North American, aged 21-29, and licensed drivers. This demographic homogeneity limits understanding of how the system would perform with questions from users of different cultural backgrounds or age groups. Cultural variations in symbol description, vocabulary preferences, and questioning styles may significantly impact system performance in more diverse real-world applications.

Symbol Coverage and Visual Analogies As demonstrated by the poor performance on symbol K.10 (“Low Tire Pressure”), the system’s effectiveness was sensitive to the completeness of visual synonyms in the knowledge graph. When users employed unexpected visual analogies not anticipated during manual annotation (e.g., describing the symbol as a “partially open vase”), the system failed to identify the intended symbol. This limitation highlights a potential imbalance between the structure provided by the knowledge graph and the flexible, creative ways humans describe unfamiliar symbols.

Synthetic Question Representation As demonstrated in Section 5.5, synthetic questions differed substantially from human-generated questions in structure, focus, and complexity. Synthetic questions exhibited greater length (averaging 22.2 words versus 16.8 for human questions), higher syntactic complexity, and more exhaustive descriptions compared to the intuitive, object-based comparisons favored by humans. These differences limited the effectiveness of synthetic questions as proxies for natural human inquiries about abstract symbols.

External API Dependencies The implementation relied on external API services to access the Gemini 2.0 Flash model. This dependency introduces potential issues related to service availability, rate limiting, and long-term sustainability if API terms change. As shown in the ablation study, these concerns become even more pronounced when attempting to implement RAG-based approaches with multi-modal models, which face substantial challenges related to computational cost, latency, and API limitations.

Performance on Complex Queries The analysis of clarification questions revealed that while the proposed system required clarification less frequently than most baseline

models (12.0% vs. up to 28.3% for baselines), evaluation scores were still notably lower when clarification was required. These metrics however aren't directly interpretable in terms of performance. Analyzing the effectiveness of clarification questions requires a thorough and methodological approach, and was not conducted as part of this thesis.

Chapter 6

Conclusion

This thesis addressed the semantic gap between visual perception and symbol meaning in automotive symbols by introducing a hybrid system that combines semiotic theory and structured knowledge representation with natural language processing. The proposed approach demonstrated significant improvements over state-of-the-art (vision-) language models in interpreting and explaining abstract symbols, with the system achieving an LLM Committee F_1 -score of 0.693 compared to the best baseline model's score of 0.256. Human evaluation further confirmed the practical utility of the approach, with the proposed system receiving an average rating of 4.56 on a 5-point Likert scale, substantially higher than baseline models (2.32 to 2.51).

The key contributions of this work include: (1) a systematic methodology for decomposing abstract symbols into their constituent visual elements, (2) a property graph schema capturing visual and semantic relationships, (3) a natural language query processing system with clarification capabilities, and (4) empirical evidence demonstrating the value of structured knowledge representation for bridging the symbol description-recognition gap. The analysis of human versus synthetic questions revealed important insights about

how humans naturally describe unfamiliar symbols, preferring intuitive analogies and focusing on distinctive features rather than exhaustive descriptions.

Notably, the performance evaluation across different question sets demonstrated a surprising trend: All models performed better on human-generated questions than on synthetic questions, despite the latter being generated with access to the ground truth visual decomposition. This finding challenges common assumptions about the equivalence of synthetic and human data in evaluating question-answering systems, suggesting that natural human questioning patterns may actually be more aligned with model capabilities than artificially constructed queries.

The ablation study further highlighted the advantages of the property graph-based approach over conventional RAG methods, demonstrating not only better performance but also substantially lower computational costs, reduced latency, and fewer API constraints. These practical advantages, combined with the higher accuracy, emphasize the value of structured knowledge representation in safety-critical contexts where performance, efficiency, and explainability are essential.

This research lays the foundations for several promising directions for future work:

1. **Automated Symbol Decomposition:** Developing computer vision techniques to automate or semi-automate the symbol decomposition process could significantly reduce the manual effort required for knowledge engineering and facilitate extension to larger, and more general symbol sets. A Human-in-the-Loop approach with Active Learning could streamline annotation while maintaining the quality required for safety-critical environments.
2. **Enhanced Visual Synonym Coverage:** Expanding the range of visual synonyms

and analogies in the knowledge graph would improve robustness to diverse description styles. This could be achieved through systematic collection and analysis of user descriptions or through generative approaches that anticipate potential visual comparisons.

3. **Advanced Query Refinement:** Implementing more sophisticated query retry mechanisms with multiple fallback strategies could enhance the system's robustness when handling complex or ambiguous queries, potentially reducing the need for clarification questions.
4. **Cross-Cultural Adaptation:** Conducting evaluations with demographically diverse participants could reveal cultural variations in symbol description patterns, allowing any adaptations to better serve global user populations.
5. **Improved Synthetic Question Generation:** Developing methods to generate synthetic questions that better mimic the structure, focus, and complexity of human questions could improve the utility of synthetic data for evaluation and training purposes.
6. **Domain Extension:** Applying the approach to other domains with abstract iconography such as medical equipment interfaces, industrial control systems, or public signage would further underscore the generalizability of the methodology and potentially reveal domain-specific considerations.
7. **Real-Time Optimization:** Optimizing the system for real-time deployment in vehicles, potentially through edge computing approaches or more efficient query processing, would enhance practical utility in automotive contexts.

The findings in this thesis demonstrate that explicitly modeling the relationships between visual elements and their semantic interpretations can significantly improve the accuracy and reliability of symbol explanations, particularly in safety-critical contexts where misinterpretation could have serious consequences. This approach represents a meaningful step toward creating more intuitive and explainable human-computer interfaces for abstract visual languages, with potential applications extending well beyond automotive symbols to other domains where accurate interpretation of abstract visual information is essential.

Bibliography

- [1] U. Manojlovic et al., “Usability of Certain Symbols Indicating Automobile Safety Status Based on Youth Assessment,” *Applied Sciences*, vol. 13, no. 17, p. 9749, 17 Jan. 2023. DOI: 10.3390/app13179749.
- [2] A. Rodak, M. Kruszewski, and B. Sztandera, “Does the Driver Understand the Warning? Comprehension of the Request to Intervene,” *Applied Sciences*, vol. 12, no. 19, p. 9451, 19 Jan. 2022. DOI: 10.3390/app12199451.
- [3] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000. DOI: 10.1109/34.895972.
- [4] C. Dorai and S. Venkatesh, “Bridging the semantic gap with computational media aesthetics,” *IEEE MultiMedia*, vol. 10, no. 2, pp. 15–17, Apr. 2003. DOI: 10.1109/MMUL.2003.1195157.
- [5] A. M. Hein, *Identification and Bridging of Semantic Gaps in the Context of Multi-Domain Engineering*. Jan. 1, 2010.

- [6] D. Gentner, "Structure-mapping: A theoretical framework for analogy," *Cognitive Science*, vol. 7, no. 2, pp. 155–170, Apr. 1, 1983. DOI: 10.1016/S0364-0213(83)80009-3.
- [7] D. Gentner and L. A. Smith, *Analogical Learning and Reasoning*. Oxford University Press, Mar. 11, 2013. DOI: 10.1093/oxfordhb/9780195376746.013.0042.
- [8] FDA CDRH, *Applying Human Factors and Usability Engineering to Medical Devices*, United States Food and Drug Administration, Feb. 9, 2019.
- [9] C. D. Wickens, J. G. Hollands, S. Banbury, and R. Parasuraman, *Engineering Psychology and Human Performance*, 4th ed. New York: Psychology Press, Aug. 19, 2015, 544 pp., ISBN: 978-1-315-66517-7. DOI: 10.4324/9781315665177.
- [10] A. Scarborough, L. Bailey, J. Pounds, and United States. Department of Transportation. Federal Aviation Administration. Office of Aviation. Civil Aerospace Medical Institute, "Examining ATC Operational Errors Using the Human Factors Analysis and Classification System," DOT/FAA/AM-05/25, Dec. 1, 2005.
- [11] W. Stubler and J. O'Hara, *Soft Controls: Technical Basis and Human Factors Review Guidance (NUREG/CR-6635)*, United States Nuclear Regulatory Commission, Mar. 2000.
- [12] C. M. Allwood and T. Kalén, "Usability in CAD—a psychological perspective," *International Journal of Human Factors in Manufacturing*, vol. 4, no. 2, pp. 145–165, Jan. 1994. DOI: 10.1002/hfm.4530040204.
- [13] K. Hayashi, Y. Sakai, H. Kamigaito, K. Hayashi, and T. Watanabe, "Towards Artwork Explanation in Large-scale Vision Language Models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2:*

- Short Papers*), L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 705–729. DOI: 10 . 18653 /v1/2024 . acl-short . 65.
- [14] D. Campbell, “Understanding the Limits of Vision Language Models Through the Lens of the Binding Problem,” Poster (Vancouver, BC, Canada), Oct. 11, 2024.
- [15] J. Fletcher et al., “Generating Vehicular Icon Descriptions and Indications Using Large Vision-Language Models,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, F. Dernoncourt, D. Preoțiuc-Pietro, and A. Shimorina, Eds., Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 1107–1120. DOI: 10 . 18653 /v1 / 2024 . emnlp-industry . 83.
- [16] ISO, *Road vehicles — Symbols for controls, indicators and tell-tales*, 2575, Sep. 2021, Published.
- [17] ISO, *Road vehicles — Symbols for controls, indicators and tell-tales, Amendment 1*, 2575/Amd 1, Jun. 2024, Published.
- [18] R. Angles, “The Property Graph Database Model,” *Alberto Mendelzon Workshop on Foundations of Data Management*, 2018.
- [19] H. Moon, Z. Zhao, J. Choi, and S. Han, “A novel property graph model for knowledge representation on the Web,” *International Journal of Engineering & Technology*, vol. 7, no. 3.33, p. 187, Aug. 29, 2018. DOI: 10 . 14419/ijet . v7i3 . 33 . 210 10.

- [20] K. Majhadi and M. Machkour, “The history and recent advances of Natural Language Interfaces for Databases Querying,” *E3S Web of Conferences*, vol. 229, S. Krit, Ed., p. 01 039, 2021. DOI: 10 . 1051/e3sconf/202122901039.
- [21] N. Francis et al., “Cypher: An Evolving Query Language for Property Graphs,” in *Proceedings of the 2018 International Conference on Management of Data*, Houston TX USA: ACM, May 27, 2018, pp. 1433–1445, ISBN: 978-1-4503-4703-7. DOI: 10 . 1145/3183713 . 3190657.
- [22] J. Gu et al., “A Survey on LLM-as-a-Judge,” version 1, no. arXiv:2411.15594, Nov. 23, 2024, Comment: 33 pages, 9 figures. arXiv admin note: text overlap with arXiv:2310.05470 by other authors. DOI: 10 . 48550/arXiv . 2411 . 15594, pre-published.
- [23] R. Puri, R. Spring, M. Shoeybi, M. Patwary, and B. Catanzaro, “Training Question Answering Models From Synthetic Data,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 5811–5826. DOI: 10 . 18653/v1/2020 . emnlp-main . 468.
- [24] A. W. Burks, “Icon, Index, and Symbol,” *Philosophy and Phenomenological Research*, vol. 9, no. 4, pp. 673–689, 1949. DOI: 10 . 2307/2103298.
- [25] A. Atkin, “Peirce’s Theory of Signs,” in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta and U. Nodelman, Eds., Spring 2023, Metaphysics Research Lab, Stanford University, 2023.
- [26] H. S. M. Yakin and A. Totu, “The Semiotic Perspectives of Peirce and Saussure: A Brief Comparative Study,” *Procedia - Social and Behavioral Sciences*, The International Conference on Communication and Media 2014 (i-COME’14) - Communica-

- tion, Empowerment and Governance: The 21st Century Enigma, vol. 155, pp. 4–8, Nov. 6, 2014. DOI: 10.1016/j.sbspro.2014.10.247.
- [27] W. Nöth and I. Jungk, “Peircean visual semiotics: Potentials to be explored,” *Semiotica*, vol. 2015, no. 207, pp. 657–673, Oct. 1, 2015. DOI: 10.1515/sem-2015-0058.
- [28] R. Lin, “An Application of the Semantic Differential to Icon Design,” *Proceedings of the Human Factors Society Annual Meeting*, vol. 36, no. 4, pp. 336–340, Oct. 1, 1992. DOI: 10.1177/154193129203600416.
- [29] S. J. Isherwood, S. J. P. McDougall, and M. B. Curry, “Icon identification in context: The changing role of icon characteristics with user experience,” *Human Factors*, vol. 49, no. 3, pp. 465–476, Jun. 2007. DOI: 10.1518/001872007X200102.
- [30] C.-F. Chi and R. S. Dewi, “Matching performance of vehicle icons in graphical and textual formats,” *Applied Ergonomics*, vol. 45, no. 4, pp. 904–916, Jul. 2014. DOI: 10.1016/j.apergo.2013.11.009.
- [31] P. Pappachan and M. Ziefle, “Cultural influences on the comprehensibility of icons in mobile-computer interaction,” *Behav. Inf. Technol.*, vol. 27, no. 4, pp. 331–337, Jul. 1, 2008. DOI: 10.1080/01449290802228399.
- [32] W. K. Horton, *The ICON Book: Visual Symbols for Computer Systems and Documentation*. USA: John Wiley & Sons, Inc., Feb. 1994, 417 pp., ISBN: 978-0-471-59901-2.
- [33] J. Littlemore, “The Effect of Cultural Background on Metaphor Interpretation,” *Metaphor and Symbol*, vol. 18, no. 4, pp. 273–288, Oct. 1, 2003. DOI: 10.1207/S15327868MS1804_4.

- [34] R. R. Behrens, "Art, Design and Gestalt Theory," *Leonardo*, vol. 31, no. 4, pp. 299–303, 1998. DOI: 10.2307/1576669.
- [35] Y. Li and K. Fu, "Research on Minimalism in Interface Design Based on Gestalt Psychology," presented at the 2022 International Conference on Science Education and Art Appreciation (SEAA 2022), Atlantis Press, Nov. 19, 2022, pp. 825–837, ISBN: 978-2-494069-05-3. DOI: 10.2991/978-2-494069-05-3_101.
- [36] D. Bühler, F. Hemmert, and J. Hurtienne, "Universal and intuitive? Scientific guidelines for icon design," in *Proceedings of Mensch Und Computer 2020*, ser. MuC '20, New York, NY, USA: Association for Computing Machinery, Sep. 6, 2020, pp. 91–103, ISBN: 978-1-4503-7540-5. DOI: 10.1145/3404983.3405518.
- [37] D. B. Stotts, "The Usefulness of Icons on the Computer Interface: Effect of Graphical Abstraction and Functional Representation on Experienced and Novice Users," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 42, no. 5, pp. 453–457, Oct. 1, 1998. DOI: 10.1177/154193129804200502.
- [38] S. J. P. Mcdougall, M. B. Curry, and O. De Bruijn, "Measuring symbol and icon characteristics: Norms for concreteness, complexity, meaningfulness, familiarity, and semantic distance for 239 symbols," *Behavior Research Methods, Instruments, & Computers*, vol. 31, no. 3, pp. 487–519, Sep. 1999. DOI: 10.3758/BF03200730.
- [39] R. S. EASTERBY, "The Perception of Symbols for Machine Displays," *Ergonomics*, vol. 13, no. 1, pp. 149–158, Jan. 1, 1970. DOI: 10.1080/00140137008931127.
- [40] J. M. Silvennoinen, T. Kujala, and J. P. P. Jokinen, "Semantic distance as a critical factor in icon design for in-car infotainment systems," *Applied Ergonomics*, vol. 65, pp. 369–381, Nov. 1, 2017. DOI: 10.1016/j.apergo.2017.07.014.

- [41] M. Friedrich, D. Richards, and M. Vollrath, “Icon Design for Representing Safety-Critical Aircraft Functions to Support Supervisory Control of Remotely Piloted Aircraft Systems,” *Aerospace*, vol. 9, no. 4, p. 181, 4 Apr. 2022. DOI: 10.3390/aerospace9040181.
- [42] R. Bahmanyar, A. Murillo Montes de Oca, and M. Datcu, “The Semantic Gap: An Exploration of User and Computer Perspectives in Earth Observation Images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 10, pp. 2046–2050, Oct. 2015. DOI: 10.1109/LGRS.2015.2444666.
- [43] S. Zhang, H. Zheng, Y. Wu, and Y. Chen, “Improving vision-language alignment with graph spiking hybrid Networks,” no. arXiv:2501.19069, Jan. 31, 2025. DOI: 10.48550/arXiv.2501.19069, pre-published.
- [44] Y. Xing, J. Kang, A. Xiao, J. Nie, L. Shao, and S. Lu, “Rewrite Caption Semantics: Bridging Semantic Gaps for Language-Supervised Semantic Segmentation,” presented at the NeurIPS 2023, Comment: NeurIPS 2023. Code is available at <https://github.com/xing05>. New Orleans, LA, USA: arXiv, Dec. 10, 2023. DOI: 10.48550/arXiv.2309.13505.
- [45] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and Tell: A Neural Image Caption Generator,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3156–3164.
- [46] K. Xu et al., “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention,” in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR, Jun. 1, 2015, pp. 2048–2057.

- [47] A. Radford et al., “Learning Transferable Visual Models From Natural Language Supervision,” in *Proceedings of the 38th International Conference on Machine Learning*, PMLR, Jul. 1, 2021, pp. 8748–8763.
- [48] J. Li, D. Li, C. Xiong, and S. Hoi, “BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation,” in *Proceedings of the 39th International Conference on Machine Learning*, PMLR, Jun. 28, 2022, pp. 12 888–12 900.
- [49] J. Li, D. Li, S. Savarese, and S. Hoi, “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models,” in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, Jul. 3, 2023, pp. 19 730–19 742.
- [50] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual Instruction Tuning,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 34 892–34 916, Dec. 15, 2023.
- [51] OpenAI et al., “GPT-4 Technical Report,” no. arXiv:2303.08774, Mar. 4, 2024, Comment: 100 pages; updated authors list; fixed author names and added citation. DOI: 10.48550/arXiv.2303.08774, pre-published.
- [52] OpenAI. “O3-mini: Pushing the frontier of cost-effective reasoning.” o3-mini: Pushing the frontier of cost-effective reasoning., Accessed: Mar. 12, 2025. [Online]. Available: <http://archive.ph/20250214/https://openai.com/index/openai-o3-mini/>.
- [53] W. Dai et al., “InstructBLIP: Towards general-purpose vision-language models with instruction tuning,” in *Proceedings of the 37th International Conference on*

- Neural Information Processing Systems*, ser. NIPS '23, Red Hook, NY, USA: Curran Associates Inc., May 30, 2024, pp. 49 250–49 267.
- [54] A. Hogan et al., “Knowledge Graphs,” *ACM Comput. Surv.*, vol. 54, no. 4, 71:1–71:37, Jul. 2, 2021. DOI: 10.1145/3447772.
- [55] V. K. Chaudhri et al., “Knowledge graphs: Introduction, history, and perspectives,” *AI Magazine*, vol. 43, no. 1, pp. 17–29, 2022. DOI: 10.1002/aaai.12033.
- [56] C. Bizer et al., “DBpedia - A crystallization point for the Web of Data,” *Journal of Web Semantics*, vol. 7, no. 3, pp. 154–165, Sep. 2009. DOI: 10.1016/j.websem.2009.07.002.
- [57] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: A collaboratively created graph database for structuring human knowledge,” in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '08, New York, NY, USA: Association for Computing Machinery, Jun. 9, 2008, pp. 1247–1250, ISBN: 978-1-60558-102-6. DOI: 10.1145/1376616.1376746.
- [58] A. Singhal. “Introducing the Knowledge Graph: Things, not strings,” Google, Accessed: Mar. 11, 2025. [Online]. Available: <https://blog.google/products/search/introducing-knowledge-graph-things-not/>.
- [59] J. Risher. “Rhetorical Device: Triple Store,” Accessed: Mar. 11, 2025. [Online]. Available: <https://www.w3.org/2001/sw/Europe/events/20031113-storage/positions/risher.html>.
- [60] F. Manola and E. Miller, “RDF primer,” W3C, W3C Recommendation, Feb. 2004, <https://www.w3.org/TR/2004/REC-rdf-primer-20040210/>.

- [61] D. Brickley and R. Guha, “RDF schema 1.1,” W3C, W3C Recommendation, Feb. 2014, <https://www.w3.org/TR/2014/REC-rdf-schema-20140225/>.
- [62] Neo4j inc. “Neo4j Graph Database,” Graph Database & Analytics, Accessed: Feb. 28, 2025. [Online]. Available: <https://neo4j.com/product/neo4j-graph-database/>.
- [63] D. Alocci, J. Mariethoz, O. Horlacher, J. T. Bolleman, M. P. Campbell, and F. Lisacek, “Property Graph vs RDF Triple Store: A Comparison on Glycan Substructure Search,” *PLOS ONE*, vol. 10, no. 12, e0144578, Dec. 14, 2015. DOI: 10.1371/journal.pone.0144578.
- [64] C. Mavromatis and G. Karypis, “ReaRev: Adaptive Reasoning for Question Answering over Knowledge Graphs,” in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 2447–2458. DOI: 10.18653/v1/2022.findings-emnlp.181.
- [65] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, “QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova et al., Eds., Online: Association for Computational Linguistics, Jun. 2021, pp. 535–546. DOI: 10.18653/v1/2021.naacl-main.45.
- [66] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, “A Comprehensive Survey of Scene Graphs: Generation and Application,” *IEEE Transactions on Pattern*

- Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1–26, Jan. 2023. DOI: 10.1109/TPAMI.2021.3137605.
- [67] Y. Chen, X. Ge, S. Yang, L. Hu, J. Li, and J. Zhang, “A Survey on Multimodal Knowledge Graphs: Construction, Completion and Applications,” *Mathematics*, vol. 11, no. 8, p. 1815, 8 Jan. 2023. DOI: 10.3390/math11081815.
- [68] H. Alberts et al., “VisualSem: A high-quality knowledge graph for vision and language,” in *Proceedings of the 1st Workshop on Multilingual Representation Learning*, D. Ataman, A. Birch, A. Conneau, O. Firat, S. Ruder, and G. G. Sahin, Eds., Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 138–152. DOI: 10.18653/v1/2021.mr1-1.13.
- [69] J. Yuan, A. Le-Tuan, M. Nguyen-Duc, T.-K. Tran, M. Hauswirth, and D. Le-Phuoc, “VisionKG: Unleashing the Power of Visual Datasets via Knowledge Graph,” in *The Semantic Web*, A. Meroño Peñuela et al., Eds., Cham: Springer Nature Switzerland, 2024, pp. 75–93, ISBN: 978-3-031-60635-9. DOI: 10.1007/978-3-031-60635-9_5.
- [70] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch, “Natural language interfaces to databases – an introduction,” *Natural Language Engineering*, vol. 1, no. 1, pp. 29–81, Mar. 1995. DOI: 10.1017/S135132490000005X.
- [71] A.-M. Popescu, O. Etzioni, and H. Kautz, “Towards a theory of natural language interfaces to databases,” in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, ser. IUI '03, New York, NY, USA: Association for Computing Machinery, Jan. 12, 2003, pp. 149–157, ISBN: 978-1-58113-586-2. DOI: 10.1145/604045.604070.

- [72] T. Yu et al., “Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3911–3921. DOI: 10.18653/v1/D18-1425.
- [73] V. Zhong, C. Xiong, and R. Socher, “Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning,” no. arXiv:1709.00103, Nov. 9, 2017, Comment: 12 pages, 5 figures. DOI: 10.48550/arXiv.1709.00103, pre-published.
- [74] B. Wang, R. Shin, X. Liu, O. Polozov, and M. Richardson, “RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schlueter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 7567–7578. DOI: 10.18653/v1/2020.acl-main.677.
- [75] N. Rajkumar, R. Li, and D. Bahdanau, “Evaluating the Text-to-SQL Capabilities of Large Language Models,” no. arXiv:2204.00498, Mar. 15, 2022. DOI: 10.48550/arXiv.2204.00498, pre-published.
- [76] S. Chang and E. Fosler-Lussier, “How to Prompt LLMs for Text-to-SQL: A Study in Zero-shot, Single-domain, and Cross-domain Settings,” presented at the NeurIPS 2023 Second Table Representation Learning Workshop, Oct. 28, 2023.

- [77] M. Liu and J. Xu, “NLI4DB: A Systematic Review of Natural Language Interfaces for Databases,” no. arXiv:2503.02435, Mar. 4, 2025. DOI: 10.48550/arXiv.2503.02435, pre-published.
- [78] S. Antol et al., “VQA: Visual Question Answering,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 7, 2015, pp. 2425–2433. DOI: 10.1109/ICCV.2015.279.
- [79] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel, “Visual question answering: A survey of methods and datasets,” *Computer Vision and Image Understanding, Language in Vision*, vol. 163, pp. 21–40, Oct. 1, 2017. DOI: 10.1016/j.cviu.2017.05.001.
- [80] P. Wang, Q. Wu, C. Shen, A. Dick, and A. Van Den Hengel, “Explicit Knowledge-based Reasoning for Visual Question Answering,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia: International Joint Conferences on Artificial Intelligence Organization, Aug. 2017, pp. 1290–1296, ISBN: 978-0-9992411-0-3. DOI: 10.24963/ijcai.2017/179.
- [81] P. Wang, Q. Wu, C. Shen, A. Dick, and A. Van Den Hengel, “FVQA: Fact-Based Visual Question Answering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2413–2427, Oct. 1, 2018. DOI: 10.1109/TPAMI.2017.2754246.
- [82] S. Shah, A. Mishra, N. Yadati, and P. P. Talukdar, “KVQA: Knowledge-Aware Visual Question Answering,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 8876–8884, 01 Jul. 17, 2019. DOI: 10.1609/aaai.v33i01.33018876.

- [83] D. Vrandečić and M. Krötzsch, “Wikidata: A free collaborative knowledgebase,” *Commun. ACM*, vol. 57, no. 10, pp. 78–85, Sep. 23, 2014. DOI: 10.1145/2629489.
- [84] Q. Tao, X. Fan, Y. Xu, X. Zhu, and Y. Tang, “Combining Knowledge Graph and LLMs for Enhanced Zero-shot Visual Question Answering,” no. arXiv:2501.12697, Jan. 22, 2025. DOI: 10.48550/arXiv.2501.12697, pre-published.
- [85] G. M. Biancofiore, Y. Deldjoo, T. D. Noia, E. Di Sciascio, and F. Narducci, “Interactive Question Answering Systems: Literature Review,” *ACM Comput. Surv.*, vol. 56, no. 9, 239:1–239:38, May 8, 2024. DOI: 10.1145/3657631.
- [86] S. S. Sakhinana, G. Sannidhi, and V. Runkana, “Towards Human-Level Understanding of Complex Process Engineering Schematics: A Pedagogical, Introspective Multi-Agent Framework for Open-Domain Question Answering,” no. arXiv:2409.00082, Aug. 24, 2024, Comment: Our paper is accepted for publication at ML4CCE workshop at ECML PKDD 2024. DOI: 10.48550/arXiv.2409.00082, pre-published.
- [87] S. S. Daneshvar and S. Wang, “GUI Element Detection Using SOTA YOLO Deep Learning Models,” version 1, no. arXiv:2408.03507, Aug. 7, 2024. DOI: 10.48550/arXiv.2408.03507, pre-published.
- [88] J. Lladós, E. Valveny, G. Sánchez, and E. Martí, “Symbol Recognition: Current Advances and Perspectives,” in *Graphics Recognition Algorithms and Applications*, D. Blostein and Y.-B. Kwon, Eds., red. by G. Goos, J. Hartmanis, and J. Van Leeuwen, vol. 2390, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 104–128, ISBN: 978-3-540-44066-6 978-3-540-45868-5. DOI: 10.1007/3-540-45868-9_9.
- [89] E. Valveny and E. Martí, “Deformable Template Matching within a Bayesian Framework for Hand-Written Graphic Symbol Recognition,” in *Graphics Recognition Re-*

- cent Advances*, A. K. Chhabra and D. Dori, Eds., red. by G. Goos, J. Hartmanis, and J. Van Leeuwen, vol. 1941, Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 193–208, ISBN: 978-3-540-41222-9 978-3-540-40953-3. DOI: 10.1007/3-540-40953-X_16.
- [90] E. Valveny and E. Marti, “Learning of structural descriptions of graphic symbols using deformable template matching,” in *Proceedings of Sixth International Conference on Document Analysis and Recognition*, Sep. 2001, pp. 455–459. DOI: 10.1109/ICDAR.2001.953831.
- [91] P. P. Roy, U. Pal, and J. Lladós, “Touching Text Character Localization in Graphical Documents Using SIFT,” in *Graphics Recognition. Achievements, Challenges, and Evolution*, J.-M. Ogier, W. Liu, and J. Lladós, Eds., red. by D. Hutchison et al., vol. 6020, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 199–211, ISBN: 978-3-642-13727-3 978-3-642-13728-0. DOI: 10.1007/978-3-642-13728-0_18.
- [92] A. González, L. Bergasa, J. J. Yebes, and M. Sotelo, “Automatic information recognition of traffic panels using SIFT descriptors and HMMs,” in *13th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2010, pp. 1289–1294. DOI: 10.1109/ITSC.2010.5625029.
- [93] M. Z. Abedin, P. Dhar, and K. Deb, “Traffic Sign Recognition using SURF: Speeded up robust feature descriptor and artificial neural network classifier,” in *2016 9th International Conference on Electrical and Computer Engineering (ICECE)*, Dec. 2016, pp. 198–201. DOI: 10.1109/ICECE.2016.7853890.

- [94] Y. Li, "Research and Application of Deep Learning in Image Recognition," in *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*, Jan. 2022, pp. 994–999. DOI: 10.1109/ICPECA53709.2022.9718847.
- [95] A. Wong, M. J. Shafiee, and M. St. Jules, "MicronNet: A Highly Compact Deep Convolutional Neural Network Architecture for Real-Time Embedded Traffic Sign Classification," *IEEE Access*, vol. 6, pp. 59 803–59 810, 2018. DOI: 10.1109/ACCESS.2018.2873948.
- [96] J. Maurício, I. Domingues, and J. Bernardino, "Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review," *Applied Sciences*, vol. 13, no. 9, p. 5521, 9 Jan. 2023. DOI: 10.3390/app13095521.
- [97] Y. Zheng and W. Jiang, "Evaluation of Vision Transformers for Traffic Sign Classification," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 3 041 117, 2022. DOI: 10.1155/2022/3041117.
- [98] D.-Y. Yun, S.-K. Seo, U. Zahid, and C.-J. Lee, "Deep Neural Network for Automatic Image Recognition of Engineering Diagrams," *Applied Sciences*, vol. 10, no. 11, p. 4005, 11 Jan. 2020. DOI: 10.3390/app10114005.
- [99] D. A. Velazquez, J. M. Gonfaus, P. Rodríguez, F. X. Roca, S. Ozawa, and J. González, "Logo Detection With No Priors," *IEEE Access*, vol. 9, pp. 106 998–107 011, 2021. DOI: 10.1109/ACCESS.2021.3101297.
- [100] Mistral AI Team. "Mistral OCR: Introducing the world's best document understanding API," Mistral OCR, Accessed: Mar. 6, 2025. [Online]. Available: <http://>

archive.today/20250308/https://mistral.ai/en/news/mistral-ocr.

- [101] L. Ueno. “Best OCR Models for Text Recognition in Images,” Roboflow Blog, Accessed: Mar. 7, 2025. [Online]. Available: <http://archive.today/20250313/https://blog.roboflow.com/best-ocr-models-text-recognition/>.
- [102] J. Zhang et al., “Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition,” *Pattern Recognition*, vol. 71, pp. 196–206, Nov. 1, 2017. DOI: 10.1016/j.patcog.2017.06.017.
- [103] J. Nielsen. “Why You Only Need to Test with 5 Users,” Nielsen Norman Group, Accessed: Mar. 10, 2025. [Online]. Available: <http://archive.today/2024.01.08/https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- [104] L. Zheng et al., “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 595–46 623, Dec. 15, 2023.
- [105] A. S. Thakur, K. Choudhary, V. S. Ramayapally, S. Vaidyanathan, and D. Hupkes, “Judging the Judges: Evaluating Alignment and Vulnerabilities in LLMs-as-Judges,” no. arXiv:2406.12624, Jan. 21, 2025. DOI: 10.48550/arXiv.2406.12624, pre-published.
- [106] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” presented at the Eighth International Conference on Learning Representations, Apr. 2020.

- [107] S. Banerjee and A. Lavie, “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments,” in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, J. Goldstein, A. Lavie, C.-Y. Lin, and C. Voss, Eds., Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 65–72.
- [108] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81.
- [109] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: A Method for Automatic Evaluation of Machine Translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, P. Isabelle, E. Charniak, and D. Lin, Eds., Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. DOI: 10.3115/1073083.1073135.
- [110] C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder, “(Meta-) Evaluation of Machine Translation,” in *Proceedings of the Second Workshop on Statistical Machine Translation*, C. Callison-Burch, P. Koehn, C. S. Fordyce, and C. Monz, Eds., Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 136–158.
- [111] E. Reiter, “A Structured Review of the Validity of BLEU,” *Computational Linguistics*, vol. 44, no. 3, pp. 393–401, Sep. 2018. DOI: 10.1162/coli_a_00322.
- [112] A. Yang, K. Liu, J. Liu, Y. Lyu, and S. Li, “Adaptations of ROUGE and BLEU to Better Evaluate Machine Reading Comprehension Task,” in *Proceedings of the Workshop on Machine Reading for Question Answering*, E. Choi, M. Seo, D. Chen, R. Jia, and

- J. Berant, Eds., Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 98–104. DOI: 10.18653/v1/W18-2611.
- [113] H. Hanani, “Balanced incomplete block designs and related designs,” *Discrete Mathematics*, vol. 11, no. 3, pp. 255–369, Jan. 1, 1975. DOI: 10.1016/0012-365X(75)90040-0.
- [114] R. D’Agostino and E. S. Pearson, “Tests for Departure from Normality,” *Biometrika*, vol. 60, no. 3, pp. 613–622, 1973. DOI: 10.2307/2335012.
- [115] E. Tomczak and M. Tomczak, “The need to report effect size estimates revisited. An overview of some recommended measures of effect size,” *TRENDS in Sport Sciences*, vol. 21, no. 1, pp. 19–25, 2014.
- [116] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick, “CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1988–1997. DOI: 10.1109/CVPR.2017.215.
- [117] S. J. Isherwood, S. J. P. McDougall, and M. B. Curry, “Icon Identification in Context: The Changing Role of Icon Characteristics With User Experience,” *Human Factors*, vol. 49, no. 3, pp. 465–476, Jun. 1, 2007. DOI: 10.1518/001872007X200102.

Appendices

A Prompts

This section provides the full-text prompts used for the individual sub-tasks of question generation, answering, and evaluation using LLM-as-a-Judge.

A.1 Question Generation

The following prompt was used for synthetic question generation, as detailed in Section 3.5.2.

```
1 You are a creative question generator. You are provided with structured information
   in JSON format about a symbol shown on a car's dashboard, or otherwise used in
   an automotive context. This JSON contains:
2 - The visual elements constituting the symbol.
3 - Any semantic groups along with their sub-components.
4 Each component may also have properties (e.g., orientation, style).
5
6 Your Task:
7 Imagine you are a driver who sees a lit symbol on the dashboard and wants to
   identify or understand its meaning. You don't fully know what the symbol
   represents – you might only notice specific shapes, letters, or lines.
```

- 8 1. Questions About Visual Features:
- 9 - Refer to the shapes, letters, lines, or orientations you observe.
 - 10 - It's okay to repeat certain phrasings like "lamp icon", "letter A", or "five lines", as real users often do.
- 11 2. Question Style:
- 12 - Some questions should focus on low-level visual details (e.g., shapes, lines, letters).
 - 13 - Others should show recognition of semantic groups (e.g., calling multiple lines plus a lamp a "light").
- 14 3. Important Restrictions:
- 15 - Make sure your question includes components of the symbol, either through semantic groups, or low-level details. You may also include counts and properties.
 - 16 - Do NOT include expressions of confusion.
 - 17 - Do NOT include guesses at the symbol's function in the question.
- 18 3. Question Count: Generate exactly `{{target_question_count}}` question(s) per symbol.
- 19 4. Output Format: Output the questions as a JSON array. Do not output any other text. Do not output code fences.

20

21 Example Symbol Data:

```
22 [  
23   {  
24     "name": "light",  
25     "components": [  
26       { "name": "line", properties: {"orientation": "horizontal"} },  
27       { "name": "line", properties: {"orientation": "horizontal"} },  
28       { "name": "line", properties: {"orientation": "horizontal"} },  
29       { "name": "line", properties: {"orientation": "horizontal"} },
```

```
30 { "name": "line", properties: {"orientation": "horizontal"} },
31 { "name": "lamp" }
32 ]
33 },
34 {
35   "name": "letter-A"
36 }
37 ]
```

38
39 Example output:

```
40 [
41   "I see a small lamp shape with five horizontal lines and a letter 'A'. What does
42     it mean?",
43   "There's a light-looking icon with lines radiating out and an 'A' next to it, can
44     you tell me what it does?",
45   "Why does this dashboard light with a lamp symbol and an 'A' always show up?",
46   "There is this lamp icon and five lines, plus the letter 'A' showing up, can you
47     explain what it means?",
48   "What is the function of this icon that appears to have a lamp and an 'A'?"
49 ]
```

47
48 Guideline summary:

- 49 - Keep the context as if the driver just noticed the symbol and is unsure of its
50 meaning.
- 51 - Allow small variations or minor repetition in how you describe the symbol's
52 features (e.g., "lamp", "light", "lines", "A").
- 53 - Focus on the 'drivers uncertainty and need for clarification or explanation.
- 54 - Return the questions as a JSON array. Do not output anything else.

```
54 ---
55
56 Now, please generate diverse user questions for the following symbol:
57 {{symbol_json}}
```

A.2 Question Answering

The following prompt was used for the baseline models to answer human and synthetic questions about automotive symbols (see Section 4.2).

```
1 You are an assistant designed to help users understand symbols used in road
   vehicles. These symbols may appear on a vehicle's dashboard, or anywhere else
   within, or on a vehicle.
2
3 Here are some examples of questions and correct, helpful answers:
4 {% for qa_pair in samples %}
5   - Question: "{{ qa_pair['question'] }}"
6     Answer:  "{{ qa_pair['answer'] }}"
7 {% endfor %}
8
9 When answering:
10 1. Always respond in a valid JSON format without any additional text or code fences
   .
11 2. Return one of the following two structures:
12   A. Confident Answer
13   If you are confident in the specific symbol being described, provide a concise
   explanation:
14   {
15     "response": "<Concise, plain-language explanation of what the symbol means and
   its function.>",
```

```
16     "symbol_name": "<Name of the symbol>",
17     "clarification_required": false
18 }
19 B. Clarification Required
20 If you suspect there are multiple possible meanings or symbols that match the
    user's description, and you need more details:
21 {
22     "response": "<A short, direct question asking for clarification about the
    symbol's appearance.>",
23     "symbol_name": ["<Possible symbol name 1>", "<Possible symbol name 2>", ...],
24     "clarification_required": true
25 }
26 3. Do not provide any additional keys or textual explanations outside of the JSON
    structure.
27
28 Make sure to strictly follow these rules and formats whenever you respond to a user
    's question.
29
30 ---
31
32 Now, provide your answer to the following question:
33 "{{ question }}"
34
35 Provide your answer in JSON only. Do not reply with any other text and do not use
    code fences.
```

For the ablation study in Section 4.3, the following system prompt was used in addition to the prompt above: “You are a helpful assistant designed to help users understand car dashboard symbols. The following excerpts have been retrieved from the ISO 2575 stan-

dard (Road vehicles - Symbols for controls, indicators and tell-tales) for you to answer the user's question: [..]".

A.3 LLM-as-a-Judge

The following prompt was used for the LLM-as-a-Judge committee metric to evaluate question-answer pairs produced by the baseline models and the proposed system:

```
1 You are an expert judge evaluating the quality of a candidate answer for a visual
  question-answering task. You will be provided with a question, a ground-truth
  answers, and a candidate answer. Your task is to assess how well the candidate
  answer meets the ground truth by determining two key metrics: precision and
  recall.
2
3 Precision should reflect the accuracy of the candidate answer - i.e. the proportion
  of the candidate's content that is correct.
4 Recall should reflect the completeness - i.e. the proportion of the ground-truth
  information that is captured by the candidate answer.
5
6 Using your internal knowledge and simple commonsense reasoning, along with the
  provided ground-truth answer, carefully compare the candidate answer to the
  ground truth. Provide a detailed explanation of your reasoning process. Then,
  assign a precision score and a recall score, each as a floating-point number
  between 0 and 1.
7
8 Output your result as a JSON object with the following keys: "explanation", "
  precision", and "recall".
9
10 For example:
11 {
```

```
12 "explanation": "The candidate answer accurately mentions key points such as X
and Y, but misses Z. Therefore, while most of the answer is correct (high
precision), it does not cover all aspects of the ground truth (moderate recall)
.",
13 "precision": 0.9,
14 "recall": 0.7
15 }
16
17 ---
18
19 Now, please evaluate the following data point:
20
21 Question: {{ question }}
22 Ground-truth Answer: {{ reference }}
23 Candidate Answer: {{ candidate }}
24
25 Provide your evaluation according to the instructions above and output your result
as a JSON object.
```

B Detailed Results Tables

The following tables, show the scores across all metrics on both (Table B.1), the human and synthetic questions, as well as on the individual synthetic (Table B.2) and human (Table B.3) question sets. For brevity, metrics have been abbreviated as follows: **P** = Precision, **R** = Recall, **F1** = F_1 -Score (harmonic mean of precision and recall).

Model	LLM Committee			BERTScore			ROUGE-1			ROUGE-L			METEOR
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	Score
Proposed System	0.759	0.655	0.693	0.765	0.744	0.753	0.605	0.650	0.612	0.521	0.554	0.525	0.527
GPT-4o Mini	0.324	0.213	0.250	0.597	0.584	0.589	0.269	0.290	0.261	0.187	0.199	0.180	0.163
Gemini 2.0 Flash	0.305	0.199	0.234	0.584	0.579	0.580	0.242	0.253	0.228	0.170	0.176	0.158	0.153
Qwen 2.5 72B	0.334	0.219	0.256	0.576	0.586	0.580	0.311	0.270	0.272	0.213	0.182	0.185	0.186
Claude 3.5 Haiku	0.299	0.195	0.228	0.584	0.588	0.585	0.256	0.233	0.231	0.176	0.159	0.158	0.156
Llama 3.3 70B	0.332	0.211	0.250	0.594	0.582	0.587	0.261	0.285	0.256	0.180	0.196	0.176	0.159

Table B.1: Combined results for all model and metric combinations, on both human and synthetic questions.

Model	LLM Committee			BERTScore			ROUGE-1			ROUGE-L			METEOR
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	Score
Proposed System	0.659	0.542	0.579	0.718	0.704	0.709	0.541	0.567	0.536	0.453	0.471	0.448	0.448
GPT-4o Mini	0.174	0.106	0.126	0.580	0.560	0.569	0.203	0.248	0.210	0.149	0.182	0.154	0.120
Gemini 2.0 Flash	0.204	0.123	0.147	0.571	0.563	0.566	0.215	0.241	0.211	0.154	0.172	0.150	0.132
Qwen 2.5 72B	0.318	0.198	0.235	0.570	0.578	0.573	0.298	0.259	0.261	0.204	0.176	0.177	0.178
Claude 3.5 Haiku	0.290	0.179	0.213	0.580	0.581	0.579	0.243	0.230	0.225	0.166	0.155	0.153	0.149
Llama 3.3 70B	0.279	0.161	0.197	0.581	0.568	0.573	0.232	0.255	0.228	0.164	0.180	0.160	0.138

Table B.2: Detailed results for all model and metric combinations on the synthetic questions dataset.

Model	LLM Committee			BERTScore			ROUGE-1			ROUGE-L			METEOR
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	Score
Proposed System	0.743	0.679	0.699	0.761	0.759	0.759	0.628	0.618	0.613	0.570	0.562	0.558	0.578
GPT-4o Mini	0.341	0.249	0.282	0.598	0.599	0.597	0.306	0.268	0.271	0.213	0.188	0.188	0.187
Gemini 2.0 Flash	0.322	0.240	0.270	0.584	0.595	0.589	0.297	0.244	0.251	0.216	0.177	0.181	0.176
Qwen 2.5 72B	0.334	0.248	0.276	0.590	0.607	0.597	0.332	0.261	0.278	0.237	0.184	0.196	0.207
Claude 3.5 Haiku	0.328	0.225	0.257	0.588	0.604	0.595	0.297	0.235	0.248	0.211	0.167	0.176	0.180
Llama 3.3 70B	0.280	0.194	0.221	0.587	0.594	0.590	0.289	0.251	0.254	0.201	0.177	0.177	0.181

Table B.3: Detailed results for all model and metric combinations on the human questions dataset.