Hof
University
University of
Applied Sciences

Evaluating the Comprehension of Technical Illustrations by Visual Language Models

Master thesis

at the

University of Applied Sciences Hof

Faculty of Computer Science

Study program Applied Research in Computer Science

submitted with:

Prof. Dr. René Peinl

Alfons-Goppel-Platz 1

95028 Hof

submitted by:

Sarang Ravi Chouguley

Bahnhofstr 29

95028 Hof

Hof, 01.05.2025

table of contents

List of figures

List of tables

List of abbreviations

TICQA	Technical Illustration Comprehension Question Answer
VSI	Visual Sequence Interpretation
TCI	Tools and Components Identification
SWR	Safety Warning Recognition
VLM	Vision Language Model
LLM	Large Language Model
LM	Language Model
MLLM	Multimodal Large Language Model

1 Introduction

1.1 Context

Humans have always been fascinated with machines and tools that can assist them in everyday life. From mechanical machines like “calculating clock” and “Turing machine”(*Turing Machines on JSTOR*, 1984) to today’s digital computers and AI models, our quest to build machines that can learn like us, perform tasks like us, think like us and communicate like us has only intensified. In the 1960s, ELIZA (Weizenbaum, 1983) demonstrated that simple pattern-matching techniques could mimic human like conversation, but it wasn’t until the advent of deep learning and the transformer architecture that models like BERT(Devlin et al., 2019), GPT-3(Brown et al., 2020), and ChatGPT(*Introducing ChatGPT | OpenAI*, n.d.) achieved truly human-like fluency.

Building on these breakthroughs, researchers fused multiple modalities like vision, sight and hearing into unified models. This resulted in computer models called MLM (Multimodel Language Models) and VLMs (Vision Language Models) that can process across multiple modalities. VLMs such as GPT-4o(*Hello GPT-4o | OpenAI*, 2024) and LLaVa(H. Liu et al., 2023) leverage joint image–text pre-training to tackle vision-language tasks ranging from text based image generation, image analysis to visual question answering. Building such models requires constant evaluation to reveal strengths, weaknesses and challenges. For this purpose researchers have designed various datasets and benchmarks evaluating VLMs on different aspects.

Yet a crucial evaluation remains unaddressed: can VLMs reliably interpret images, graphics and illustrations, specifically technical illustrations present in product manuals—exploded-views, safety icons, and assembly sequences? These graphics are designed to convey intricate assembly sequences, hazard warnings, and maintenance procedures without heavy textual explanations. Exploring where SOTA VLMs performance stand on such illustrations can reveal new strengths and areas for application.

This thesis, first introduces TICQA (Technical Illustration Comprehension QA), a dataset designed to probe three core capabilities—Visual Assembly/Disassembly Interpretation, Tools & Components Identification, and Safety Warning Recognition—in VLMs. And then evaluates SOTA VLMs on this dataset. By redacting textual cues and employing a combination of MCQ and panel-based evaluation methodology, TICQA isolates pure visual reasoning over technical diagrams and exposes where today’s models excel—and where they still fall short.

1.2 Motivation and Goal

Technical illustrations help convey complex information in form of compact graphics. These illustrations appear almost everywhere, from school textbooks to scientific journals. Product manuals use them for safety warnings, assembly and operation instructions, and troubleshooting. To comprehend them, VLMs must recognise text, symbols, and visual cues; reason about spatial relations; and apply background knowledge. Successful comprehension of these illustrations would both demonstrate multi-faceted reasoning and enable many illustration-based applications.

Where simple graphics (charts, photos) contain one element, technical illustrations integrate text, symbols, and diagrams—and demand context-sensitive interpretation. Most datasets and benchmarks focus on either holistic evaluation or task-specific fine-grained evaluation. Examples of holistic benchmarks include MMBench (Xu et al., 2023), MMMU (Yue et al., 2024), MMStart(Chen et al., 2024), and OCRBench (Y. Liu et al., 2024). Although necessary for general VLM evaluation, they don't target technical illustrations; thus, results on them would be misleading. There is need for a comprehensive and accurate evaluation of VLMs on a dataset that measures performance on tasks comprising technical-illustration comprehension.

To accurately measure VLMs' performance in technical-illustration comprehension, VLMs should be tested for OCR, background knowledge, and spatial reasoning; they should also distinguish symbols, tools, and actions in manuals. In other words, a good dataset would measure whether VLMs truly *understand* symbol meanings, link assembly sequences, and identify tools and components. Based on this idea, the goal for this thesis is defined as follows:

Goal: Develop a custom dataset and evaluation framework to measure VLMs' OCR accuracy, background-knowledge reasoning, spatial-relation inference, symbol discrimination, assembly-sequence linking, and tool/component identification in product-manual illustrations.

1.3 Objectives and Contribution

This thesis pursues three primary objectives. First, the design of **TICQA** (Technical Illustration Comprehension QA), a benchmark specifically tailored to probe VLM understanding of product-manual graphics. TICQA is organised into three complementary tasks—**Visual Sequence Interpretation (VSI)**, **Tools & Components Identification (TCI)**, and **Safety Warning Recognition (SWR)**—and employs a rigorous **redaction protocol** that strips away all textual cues from illustrations, forcing models to rely on genuine visual reasoning.

Second, use of a **mixed evaluation setting** that allows automated evaluation of different answer types. For open-ended SWR and short-answer TCI tasks, I leverage a **panel of medium-sized LLM judges** (PoLL) to rate semantic correctness. For the multiple-choice VAI task, I employ exact-match scoring to calculate sequence-interpretation accuracy.

Third, I conducted an **empirical study** of nine state-of-the-art Vision-Language Models (all top performing and under 20 B parameters) drawn from the latest OpenVLM leaderboard. By benchmarking these models on TICQA's three tasks, I (a) expose their strengths and weaknesses in domain-specific schematics, and (b) analyse the datasets strengths and weaknesses.

Together, these contributions fill a critical gap in multimodal evaluation, establish a new dataset for assessing technical-illustration comprehension, and open a path forward for more robust, visually grounded AI systems.

1.4 Research Questions and Hypothesis

To guide the investigation of VLM comprehension on technical illustrations, I pose three research questions:

- **RQ1**: How accurately can state-of-the-art VLMs infer the correct assembly/disassembly sequence in assembly diagrams when textual cues are redacted?
- **RQ2**: To what extent can these models identify tools and components purely from visual shape cues, without relying on overlaid labels?
- **RQ3**: How effectively do VLMs recognise and describe safety warnings and hazard icons in open-ended questions?

From these questions, I formulate two hypotheses:

- **H1 (VLM vs. Human)**: VLMs will underperform humans on all the tasks.
- **H2 (With vs. Without context)**: Adding additional contextual information yields higher performance and consistency than without context questions.

These RQs and hypotheses frame my empirical study of nine VLMs under 20 billion parameters, isolating visual reasoning from textual shortcuts and assessing both model performance and dataset methodology.

1.5 Outline

The rest of the work is structured as follows: Chapter 2 provides background on the evolution of language, multimodal, and vision–language models, outlining foundational architectures and established evaluation methods. Chapter 3 then details the design of the TICQA dataset, including the three task subsets (VAI, TCI, SWR), the redaction protocol, and quality-control procedures. Chapter 4 describes the experimental setup: model selection criteria, hardware environment, prompt engineering, and the panel-LLM plus human evaluation framework. Chapter 5 presents empirical results, correlation analyses with existing benchmarks, and hallucination diagnostics. Chapter 6 presents key findings, highlighting model strengths, failure modes, and implications of including context. Lastly, Chapter 7 concludes with a summary of contributions, discusses limitations, and proposes directions for future research.

2 Chapter 2: Related Work

This chapter provides an in-depth review of the background and evolution of Vision-Language Models (VLMs). We begin by tracing the development of language models (LMs), from their early stages as statistical models to the introduction of neural language models, and finally, the rise of large language models (LLMs). The subsequent sections focus on the emergence of multimodal language models (MLMs), which integrate text, images, and other modalities, leading to the development of VLMs. The final section explores the evaluation techniques and benchmarks for these advanced AI models, highlighting the latest datasets and approaches used to assess the capabilities of VLMs in real-world applications. This chapter lays the foundation for understanding the work done in this thesis.

2.1 Language Models

A Language model (LM) is a computational model that predicts the probability of next token (e.g., a character word) based on previous tokens. LMs have been developed to perform tasks requiring comprehension of human language; these fall under the domain of natural language processing (NLP). This section reviews the history of language models, including different phases of development and the current landscape.

The development of modern-day language models can be tracked to four major phases in NLP(Zhao et al., 2023) : Statistical Language Models (SLM) (*Foundations of Statistical Natural Language Processing - Christopher Manning, Hinrich Schutze - MIT Press*, 1998; Jelinek, 1976), Neural Language Models (NLM) (Bengio et al., 2003; Mikolov et al., 2010), Pre-trained language models (PLM) (Devlin et al., 2019), and Large Language Models (LLM) (Shanahan, 2024a).

Statistical Language Model (SLM):

Statistical Language Models use the statistical learning methods developed in the 1990s. SLMs are word-prediction models that predict the next word based on preceding context. SLMs were widely used for information retrieval (IR) (X. Liu & Croft, 2005) and natural language processing (NLP) tasks like speech recognition (Thede & Harper, n.d.). Popular SLMs types include the n-gram, maximum entropy and skip-gram models. N-gram models are based on the Markov assumption (Chung, 1960) (i.e., probability of the next word in a sequence depends only on a fixed sized window of previous words). However, n-gram models suffer from the problem of zero probability (i.e zero probability for unseen n-grams). To address this issue various methods like Good-Turing discounting (Gale & Sampson, 1995) and back-off estimation(Katz, 1987) were developed. Later, maximum entropy and skip-gram models were introduced to overcome the data sparsity problem of n-gram models. Even with these developments, all SLMs faced issues like inability to model complex patterns and small context windows. Neural network-based models later addressed these limitations.

Neural Language Model (NLM):

The introduction of neural network models, specifically recurrent neural networks (RNN) (Bengio et al., 1994) and LSTM (Long Short-Term Memory)(Hochreiter & Schmidhuber, 1997) , which could handle sequence data effectively, shifted the focus from SLMs to Neural Language Models (NLM). Another important development during this period was development of distributed representation of words (Bengio et al., 2003) and word2vec (Mikolov et al., 2013). Unlike statistical word prediction models, models based on distributed representation of words could capture more complex word relationships. This combination improved the performance of NLMs on NLP tasks like speech recognition, semantic analysis, and machine translation (Collobert et al., 2011). NLMs focused on learning features and relationships of text data instead of just word-sequenced modeling. As a result, NLMs can generalise better on unseen sequences, capture semantic similarities and excel in complex NLP tasks. While NLMs showed task-specific improvements, they struggled to generalise across a broad range of NLP tasks. This changed with the development of pre-trained language models (PLM).

Pre-trained Language Model (PLM)

ELMo (Peters et al., 2018), which is a bidirectional LSTM (biLSTM) model trained on large corpus of text, was one of the first pre-trained language models. Elmo introduced a new type of "deep contextualised word representation" (Peters et al., 2018). ELMo could model the complex characteristics of word use and variability of word use across different linguistic contexts. However, training RNNs and LSTM based networks on large amounts of data was largely restricted by the sequential nature of these networks. However, the introduction of the transformer architecture (Vaswani et al., 2017) solved this problem by enabling parallelisation. BERT(Devlin et al., 2019) was one of the first model trained on this architecture to catch attention of researchers. The context aware representations learned by BERT showed very effective general purpose semantic understanding. This breakthrough in training language models on large text corpus laid the foundation for now widely used "pre-training and fine-tuning" paradigm. Subsequent studies developed better models such as GPT-2 (Radford et al., n.d.) and BART (Lewis et al., 2019) and refined pre-training strategies (Fedus et al., 2022; Sanh et al., 2021; T. Wang et al., 2022). Fine-tuning became a crucial step for adapting PLMs to various NLP tasks. However, it was the release of ChatGPT by OpenAI in the late 2022 which caught the attention of the world.

Large Language Model (LLM)

ChatGPT, which is a chatbot application based on GPT-3 (175B-parameter language model), surprised the world with its ability to perform wide variety of NLP tasks and communicate effortlessly with users. ChatGPT was able to solve tasks it hadn't been specifically trained on, unlike previous PLMs like

GPT-2 and BART. This capability of large-scale PLMs became known as “emergent abilities” (Wei et al., 2022). Researchers also discovered that scaling model size or training data of PLMs (according to scaling law (Kaplan et al., 2020)) lead to improved performance on downstream tasks and better generalisation. These larger PLMs became known as large language models (LLMs) (Shanahan, 2024b) due to their enormous size. One of the key abilities of LLMs is in-context learning (Brown et al., 2020). In-context learning allows LLMs to solve complex tasks with only task-specific instructions and few-task specific examples (few-shot) or none at all (zero-shot) as prompt (i.e. input). This enables them to generalise across diverse tasks, such as machine translation, question answering, and domain adaptation, without additional training. This few-shot and zero-shot prompting came to be know as prompt engineering, which allows users to design specific prompt texts to guide LLMs in generating desired responses or complete specific tasks, becoming the standard for interacting with LLMs.

Large Language Models (LLMs) have ushered in the era of generative artificial intelligence (Gen AI), where computational models are beginning to exhibit signs of general intelligence. Inspired by the success of LLMs, various other AI models are being developed to tackle tasks that require human-level capabilities, including image understanding, language comprehension, reasoning, and multimodal tasks. These advancements are no longer confined to the realm of natural language processing (NLP) but are increasingly expanding across diverse domains, marking a significant leap forward in AI's ability to perform complex, human-like tasks. As the field continues to evolve, these innovations promise to transform a wide array of industries and redefine the boundaries of what artificial intelligence can achieve.

2.2 Multimodal Language Models

Following the development of LLMs, the next wave of development in the field of AI was the combining of multiple modalities like text, image, audio, etc into LLM resulting in Multimodal Language Models (MLM). These models can process, understand, and generate text, image and/or audio to solve real world tasks. This section explores the concept of multimodality, the phases of multimodal research leading to the emergence of multimodal language models.

According to Turk (2014), multimodality is the ability to convey or interpret information across several distinct modalities. With respect to computers, multimodal data refers to a combination of variety of data types — such as images, numeric values, text, audio, etc. This fusion of data allows for more comprehensive representation of information in computers. Because humans naturally excel at processing information presented in multiple mode, it is both logical and advantageous to design machines that emulate the same behaviour.

Phases of multimodal research

Single modality (1980-2000):

The origins of modality research can be traced back to the 1980s, when early image and speech recognition systems were developed based on statistical algorithms and image-processing techniques. The work done during this period laid the foundations for future work in this domain. For example, development of hidden Markov models (HMMs) (Bahl et al., 1986) to improve accuracy and reliability of speech recognition technology or development of Eigenfaces (Satoh & Kanade, 1997), a face recognition method, which could extract facial features to recognise individuals based on statistical patterns in face images (Lien et al., 1998).

Modality conversion (2000-2010):

This period witnessed a lot of research focused on studies of human-computer interactions. Several important researches and projects like AMI project (Kraaij et al., 2005) in 2001, CALO (Tur et al., 2010) in 2003, and SSP (Vinciarelli et al., 2008) in 2008 which aimed at simulating human behaviour in computers were started during this period. The AMI proposed utilising computers to record and process meeting data. The aim was to develop technologies that could process audio, video and text data from meetings. CALO, which stands for "Cognitive Assistant that Learns and Organises", influenced the development of Siri and aimed to create an intelligent virtual assistant capable of understanding and responding to human language while performing tasks. Social Signal Processing (SSP) aimed to analyse non-verbal clues such as facial expressions, gestures and voice tones to understand social interactions and facilitate more natural human-computer interaction. The work done during this period advanced research towards fusion of modalities.

Modality fusion (2010-2020): At this stage, the combination of deep learning techniques and systems which could handle multiple modalities led to further advancements in this domain. In 2011, one of the first multimodal deep learning algorithms was introduced (Ngiam et al., 2011) which proposed the use of deep neural networks to learn features across multiple modalities, specifically audio and video. It enhanced the performance of models in tasks like image classification, speech recognition and video analysis. In 2012, another multimodal learning algorithm, based on deep Boltzmann machines (DBMs) (Hinton & Salakhutdinov, 2012), was introduced. In 2016, a neural image captioning algorithm with semantic attention was developed which could generate human-like descriptive captions for images by analysing the visual contents, enhancing automated image understanding and interpretation (You et al., 2016). By the end of the 2020s, various models had been developed that could process and generate multimodal data. This was pushed to the next level by the advent of large-scale models in the early 2020s.

Huge multimodal models (2020-present):

The rapid development of large-scale models trained on huge datasets opened up new opportunities for multimodal models. In 2021, CLIP (Contrastive Language Image Pre-training) (Radford et al., 2021) was released. CLIP could understand and correlate both textual and visual data, enabling it to perform tasks like zero-shot image classification on unseen categories and retrieve images based on text queries. Moving towards image generation, DALL-E (Ramesh et al., 2021) was released and could generate high-quality graphics and images from text prompts. Then, in 2023, KOSMOS-1 (S. Huang et al., 2023) was released by Microsoft, which could perceive text and images, follow instructions and learn in-context. This allowed KOSMOS-1 to perform tasks like visual question answering. Another notable work was the combination of LLMs and image transformers in PALM-E (Driess et al., n.d.). These models excelled in tasks like object detection, image understanding and visual question answering. Apart from models which focused on text and image modalities, several other multimodal models which combined audio, video or text were also developed.

All these developments resulted in the development of multimodal large language models (MLLM) which excelled in all 3 modalities task: natural language , vision and audio. These models marked the beginning of a new era in AI, where machines could seamlessly integrate and solve complex real-world challenges across multiple modalities.

2.3 Vision Language Models

Vision Language Models (VLMs) mark a significant advancement in AI by bridging the gap between visual perception and natural language understanding. These models integrate computer vision capabilities with language processing to enable AI systems that can comprehend, reason and communicate information in both visual and textual modalities. Previous sections introduced LLMs and their expanded version MLMs. This section explores VLMs, a type of MLM, specifically their architecture and training.

At the very core, VLMs are deep neural networks trained on huge corpus of text and images. To fully grasp their capabilities, it is important to understand their fundamental components and architecture. The two fundamental components of VLMs are a text encoder and an image encoder.

A text encoder converts tokenised text into vector embeddings, a numerical representation which capture the semantic and positional relationship between tokens. This allows text encoders to process input sequences and generate rich contextual internal representations (Bostrom & Durrett, 2020 ;Peters et al., 2018). A decoder, which is essentially the same as the encoder, generates coherent output sequences using encoder representations and previously generated tokens. In a large language model, only text is converted into embeddings, whereas in a VLM, both image and text are converted into embeddings.

An image encoder performs the same function as a text encoder but for images. It captures the visual properties of an image, such as colours, shapes, and textures, and converts them into vector embeddings. While earlier versions of VLMs used convolutional neural networks (CNNs) for feature extraction, modern VLMs use a vision transformer (ViT) (Dou et al., 2021). While CNN-based encoders tokenise an image by dividing it into grids and extracting grid-based features, ViT-based encoders use patch-based methods, which involve extracting linear projections from the image by dividing it into patches (Yang et al., 2022).

VLMs combine image and text encoders in their architecture for both training and generation. The architecture of VLMs can be divided into two types: encoder-only and encoder-decoder. An encoder-only architecture uses just the transformer's encoder module, whereas an encoder–decoder architecture incorporates both the encoder and the decoder modules. Examples of encoder-only models include CLIP (Ramesh et al., 2022) and ALBEF (J. Li et al., 2021) , while an example of an encoder-decoder model is SimVLM (Z. Wang et al., 2021). Models such as CLIP, BLIP, and ALIGN (Jia et al., 2021) use both an image encoder and a text encoder. This allows models to align image and text embeddings, effectively aligning and capturing cross-modal relationships between image and text. Another important part of the architecture is the cross-attention mechanism. The cross-attention mechanism allows dynamic interaction between visual and textual features by enabling tokens from one modality (e.g., vision) to influence tokens from the other modality (e.g., text) (H. Lin et al., 2022).

Once the architecture is designed, the next step is training. Training enables VLMs to learn both individual and cross-modal features of images and text. The selection of learning objectives plays important role during training, as they define the focus of model. Some of the examples are learning objectives for VLMs are image-text contrastive learning (ITC), masked language modeling (MLM), masked vision modeling (MVM), and image-text matching (TM) (Rao et al., 2023). Based on these learning methods, training can be categorised into two categories: training with contrastive learning and training with masking.

Contrastive learning maps the image and text embeddings from their respective encoders into a shared embedding space. The VLM is trained on datasets of image-text pairs with the objective of minimising the distance between the embeddings of matching pairs and maximising the distance for non-matching pairs. Masking allows VLMs to predict randomly masked parts of input text or images. In masked language modeling, VLMs learn to fill in missing words in a text caption, given an incomplete image.

Initially, it made sense to train models like KOSMOS-1 from scratch, but more recent models use pre-trained encoders due to the resource-intensive nature of training from scratch. With the availability of pre-trained LLMs and vision encoders, VLMs can now be built using these pre-trained encoders along with a mapping layer that aligns the visual representations of an image to the LLM's input space. One example of such a model is LLaVA (H. Liu et al., 2023b), which uses the Vicuna LLM and CLIP ViT as pre-trained text and image encoders, respectively.

VLMs represent one of the most advanced AI models capable of simultaneously understanding and processing data across different modalities. This multimodal capability provides several key advantages over unimodal approaches. Firstly, VLMs enhance knowledge learning by integrating both visual and textual domains. The combination of visual features and textual descriptions significantly enriches the model's understanding of both text and images. Secondly, VLMs can handle more complex tasks that require visual understanding, as well as improvements to existing tasks such as image captioning. Thirdly, architectural advancements in VLMs—particularly the shift toward using pre-trained vision encoders and powerful language models—have enabled more complex reasoning about multimodal content. Modern approaches, such as used in LLaVA, demonstrate how leveraging pre-trained components enhances transfer learning and domain adaptation."

As VLMs continue to evolve, further improvements in multimodal fusion techniques, enhanced training, and other areas are expected. These advancements will broaden the applicability of VLMs across domains such as healthcare, education, creative industries, and scientific research. They will open new possibilities for human-AI interaction through systems that can truly see and communicate.

2.4 Evaluation of AI models

Evaluation of AI models is essential for ensuring the reliability, effectiveness, and safety of these systems. It also helps identify strengths, weaknesses, biases, and gaps to optimise performance in real-world applications. AI models like LLMs and VLMs are essentially 'black boxes' because their internal decision-making processes are unknown and not easily understandable. While it's possible to analyse their inputs (data fed into the model) and outputs (predictions made by the model), the intricate computations and logic behind a given output remain hidden. For this reason, accurate evaluation has always been a challenge for such models. In the following section, I summarise the evolution of evaluation techniques and datasets for AI models.

Traditional machine learning and deep learning models are evaluated on datasets tailored to their specific characteristics and applications. For example, techniques like k-fold cross-validation , holdout validation, and bootstrap are used to split datasets into training and testing sets. Metrics like accuracy, precision, recall, F1-score, and mean squared error (MSE) are used to measure the performance of machine learning and deep learning models (Browne, 2000). The evaluation of NLP algorithms or models can be loosely grouped into two types: automatic/manual evaluation, and intrinsic/extrinsic evaluation (Clark et al., 2010).

Automatic and manual evaluation:

The most straightforward way to evaluate an NLP model (or algorithm) is to recruit human subjects and ask them to assess the system's output based on predetermined criteria. This is known as manual evaluation, as it involves humans manually entering input into the system and judging the output.

Although manual evaluation is the best method for determining a model's usefulness, it has two significant limitations: it is often inconsistent and slow. As the number of evaluation tasks increases, these limitations significantly impact the development process. Therefore, many researchers prefer automatic evaluation, at least in the early stages. Automatic evaluations provide a computational way to evaluate models based on predefined criteria, metrics, and strategies. These computational methods can be programmed to run and evaluate models automatically. However, automatic evaluations can barely surpass the accuracy and relevance of manual evaluations. Hence, to maintain a balance between automatic and manual evaluation, regular studies are conducted to measure their correlation.

Intrinsic and extrinsic evaluation:

Intrinsic and extrinsic evaluations are another form of evaluation. In an intrinsic evaluation, system output is directly evaluated in terms of a set of predefined criteria based on the desired functionality of the model (Clark et al., 2010). Intrinsic evaluations help in model training by providing insights into the model's training performance. In an extrinsic evaluation, model output is assessed based on its impact on a task external to the model itself. Extrinsic evaluations are important for understanding the impact and limitations of trained models in real-world scenarios. Generally, intrinsic evaluations tend to be automatic, and extrinsic evaluations tend to be manual. However, as the complexity and number of real-world scenarios increase, this rule is less strictly followed.

As computer systems evolve from simple NLP task solvers to generalized AI systems, extrinsic evaluations have also grown and evolved. Since, internally, these AI systems are still neural networks using similar training strategies, intrinsic evaluations have not evolved as significantly as extrinsic evaluations. Below, I discuss datasets, benchmarks and techniques for extrinsic evaluation of LLMs and VLMs.

Datasets for evaluation of LLMs

Since the initial idea behind the development of language models was to enhance the performance of computational models on NLP tasks, many of the early datasets focused on evaluating performance on existing NLP tasks such as text classification, sentiment analysis, and translation. However, as these models improved on traditional NLP tasks and expanded their reach to more complex ones, newer datasets, metrics, and benchmarks emerged.

Although there is no formal definition of what constitutes a traditional NLP task, tasks such as semantic analysis, text classification, natural language inference, summarisation, and translation may be considered as traditional NLP tasks. Benchmarks such as (Bang et al., 2023; Laskar et al., 2023; Liang et al., 2022) evaluate models on semantic analysis, (Peña et al., 2023) measure performance on text classification, and (P. Wang et al., 2023) provide evaluation on summarisation and translation.

As traditional NLP models evolved into LLMs, new abilities, tasks, applications, datasets, and benchmarks emerged. These benchmarks were designed to accommodate the complexities of evaluating LLMs like different response types, inaccuracies, hallucinations, etc. One of the first prominent benchmarks for LLMs was BIG-bench (Srivastava et al., 2022); it consisted of 204+ language tasks across fields including linguistics, common sense, science, and more. BIG-bench provided a comprehensive mechanism for evaluating LLMs. This evaluation revealed strengths and weaknesses such as reasoning ability, advanced QA, hallucination, bias, and more in LLMs. To measure these strengths and weaknesses, additional datasets and benchmarks were released. To measure robustness in LLMs, defined as model stability when faced with unexpected input, benchmarks like (Bang et al., 2023) (Nie et al., 2019) were designed. Factuality, another important factor in LLMs, refers to the accuracy of answers generated by LLMs based on real-world information. This was the focus of benchmarks like (Kwiatkowski et al., 2019). Another important domain was Ethics and Bias, which measures harmful content such as toxic language, hate speech, and insults, as well as biases like stereotypes related to demographic identities. These vulnerabilities were assessed by benchmarks such as (B. Wang et al., 2023).

Datasets for evaluation of VLMs

With the evolution of LLMs into VLMs, the existing benchmarks proved insufficient for evaluating these models. There was an increasing need for benchmarks and datasets that evaluate VLMs on vision and language-vision tasks. One of the earliest applications of VLMs was image captioning, and some existing datasets like MSCOCO (T. Y. Lin et al., 2014) and Nocaps (Agrawal et al., 2019) were utilised to measure exactly this. One of the first benchmarks for VLMs was VLUE (Vision-Language Understanding) (Zhou et al., 2022) , which measured VLMs' performance on five tasks: Image-Text Retrieval, Visual Reasoning, Visual Grounding, Visual Question Answering, and Image Captioning. Visual Question Answering (VQA) became a major focus for VLMs, with benchmarks like A-OKVQA (Schwenk et al., 2022) , Adversarial VQA (L. Li et al., 2021), and OK-VQA (Marino et al., 2019) focusing on its evaluation.

As VLMs began to improve in their training and architecture, new tasks such as multimodal perception, multimodal recognition, and multimodal reasoning emerged. Multimodal recognition cane further divided into focused tasks like concept recognition, action recognition, attribute recognition, OCR, and more. Multimodal perception involves object localisation (identifying spatial positions of objects in a scene), object relations (understanding spatial relationships between objects), and object interaction understanding. Whereas, multimodal reasoning includes visual relation reasoning, in addition to common sense reasoning (J. Huang & Zhang, n.d.). Most of today's benchmarks, such as MM-Bench (Xu et al., n.d.), MMT-Bench (Ying et al., 2024), MM-Vet (Yu et al., 2023) and GQA (Hudson & Manning, 2019), focus on measuring performance on these tasks for a holistic evaluation of VLMs.

Other benchmarks, such as POPE (Pham & Schott, 2024), HallusionBench (Guan et al., 2024), and OCRBench (Y. Liu et al., 2024), focus on specific tasks like hallucination and OCR.

Evaluation Techniques

In addition to the benchmarks and datasets, evaluating LLMs and VLMs requires special evaluation mechanisms and metrics.

Evaluation methods:

Evaluation methods for Large Language Models (LLMs) and Vision-Language Models (VLMs) are diverse, reflecting the complexity and variety of tasks these models perform. Manual evaluation, often considered the gold standard, involves human annotators rating outputs for qualities like correctness, relevance, and safety, providing nuanced insights but being costly and time-intensive. LLM-based evaluation, or "LLM-as-a-judge," (Shen et al., 2023) leverages a strong language model to assess the outputs of other models, offering scalable and flexible evaluation for subjective or open-ended tasks, though it introduces potential biases from the evaluating model itself. PoLL (Panel of LLM Judges) involves employing a set of LMs to compare the outputs and ground truth, which allows cheap, automatic, unbiased evaluation (Verga et al., 2024).

Multiple-choice questions (MCQs) are widely used for their simplicity and efficiency, especially in knowledge-based benchmarks, but research shows they may not fully capture a model's capabilities in open-ended or real-world scenarios and can be sensitive to answer order. Therefore, MCQs are best suited for general knowledge testing, while long-form generation (LFG) or open-ended questions, often evaluated manually or by LLMs, provide deeper insights into model reasoning and robustness. Overall, effective evaluation typically combines several methods-manual, automatic, LLM-based, PoLL-based, and MCQ-to achieve a comprehensive understanding of model performance across tasks and modalities.

Evaluation metrics:

Automatic evaluation uses metrics such as BLEU (Papineni et al., n.d.) , ROUGE (C.-Y. Lin, n.d.), METEOR (Banerjee & Lavie, n.d.) , and CIDEr (dos Santos et al., 2021) for VLMs, especially in tasks like image captioning, where generated text is compared to ground truth references for precision and recall. For LLMs, automatic methods include reference-based metrics (comparing outputs to known answers) and reference-free metrics (assessing qualities like fluency or safety without ground truth), with techniques such as perplexity and semantic similarity.

In this chapter, I presented the evolution of language models, culminating in the development of Vision-Language Models (VLMs). I discussed the progression from statistical and neural language models to large-scale LLMs, followed by the integration of multiple modalities in multimodal language models. The rise of VLMs, which combine visual and textual data for enhanced understanding and reasoning, marks a significant leap in AI's capabilities. Furthermore, I reviewed the challenges and advancements in evaluating these models, with a focus on the specialised benchmarks and evaluation techniques designed for assessing VLMs. As AI continues to evolve, the development of robust evaluation methods and datasets will remain essential for ensuring the reliability, fairness, and performance of these powerful models in diverse applications.

3 Chapter 3: TICQA Dataset

This chapter introduces the TICQA dataset, a mix of open-question and closed-question VQA dataset with 250 samples from 42 consumer product manuals. It begins by outlining the motivation for TICQA, highlighting the need for a specialised dataset that addresses the challenges of technical illustration question answering. The chapter then presents the dataset's guidelines, detailing the various sub-tasks it encompasses. Additionally, it describes the creation methodology employed in curating the TICQA dataset, along with the quality checks that were implemented to ensure its robustness and reliability.

3.1 Motivation: Limitation of existing datasets

Technical illustrations are complex information graphics comprising text, images, and symbols. They are typically found in documents like product manuals, school textbooks, and technical artefacts. Since this work focuses on technical illustrations found in product manuals, I analysed existing datasets (and benchmarks) related to document VQA, OCR, and the comprehensive evaluation of VLMs. I found that, first, almost no dataset (or benchmark) focuses on technical illustrations, and second, those that do are not very effective at measuring the visual abilities required for truly understanding technical diagrams. Below, I present my findings.

The majority of document datasets—such as DocVQA (Mathew et al., 2021), InforgraphicVQA (Mathew et al., 2022), MP-DocVQA (Kang et al., 2024), DocGenome (Xia et al., 2024), and PDF-VQA (Ding et al., 2024)—focus almost exclusively on industry reports, infographics, scientific figures, slide decks, or book covers. While these datasets include complex layouts and some diagrammatic content, none are tailored to the dense combination of annotated diagrams, exploded views, symbols, and procedural steps that characterise true technical illustrations in product manuals.

Likewise, large-scale OCR and form-understanding datasets (e.g., OCRBench) (Y. Liu et al., 2024) excel at evaluating text recognition, layout parsing, and key-value extraction. However, they do not measure deeper visual reasoning over schematics or multi-component diagrams. Their questions are typically fact-oriented (e.g., "What's the invoice number?") rather than inference-driven (e.g., "What is the correct sequence of actions?"). As a result, these datasets miss the multimodal interplay of symbolic icons and technical language that technical-illustration understanding demands.

Newer "comprehensive" benchmarks, such as SEED-Bench (B. Li et al., 2023) , SEED-Bench-2-Plus (B. Li et al., 2024), and MMDocBench (F. Zhu et al., 2024), aggregate across multiple domains and provide useful stress tests for general VLM capabilities. However, they still lack dedicated subtasks for assembly diagrams, exploded-view reasoning, or sequential procedural logic.

Finally, a handful of technical collections (e.g., E-manual, TechQA, the Technical Illustrations and PM209 dataset) do exist, however, they either focus on pure textual QA (E-manual and TechQA) or provide only basic and surface level questions like “How to operate the device shown in image ?”(the Technical Illustrations and PM209 dataset) (Castelli et al., 2019; Hussein, 2024; Nandy et al., 2021; L. Zhang et al., 2023).

In summary, no existing dataset rigorously evaluates the specialised visual skills required to fully understand the complex semantics of technical illustrations in product manuals.

3.2 Guidelines for dataset

Based on the above analysis, I formulate the following guidelines for building TICQA.

1. Comprehensive representation of technical illustrations

The dataset should include a wide range of technical illustrations that reflect variety of diagrams found in product manuals. These should include exploded view, assembly instructions, components diagram and operating procedures.

2. Complex and Inference-driven questioning

The dataset should include questions that require inference and reasoning beyond simple fact-based queries. For example, questions like “What is the correct sequence of actions to assemble this component ?” Or “ Which part of the diagram corresponds to this label ?” will test the model’s ability to interpret spatial reasoning, sequential reasoning, and contextual understanding of technical diagrams.

3. Hierarchical structure of tasks

The dataset should support hierarchical tasks with increasing complexity. For example, it should start with basic questions, such as identifying parts of a diagram or recognising symbols, and progress to more complex tasks, such as reasoning about the assembly steps or understanding dynamic interactions between components. This hierarchical approach will help evaluate models at different levels of understanding, from basic recognition to complex problem-solving.

5. Redaction, ground truth and context

To enable accurate evaluation, each illustration in dataset should have extra text redacted, annotated with ground truth and context information. Redaction of text would force the model to use visual cues rather than cheating by relying on text clues. This will ensure accurate evaluation of the model’s visual abilities based on ground truth.

6. Diverse product manual domains

The dataset should include technical illustrations from a wide range of product domains, such as electronics, mechanical systems, appliances and furniture. This diversity will ensure that models evaluated on the dataset can generalise across various types of technical diagrams.

3.4 Defining sub-tasks for dataset

For a structured evaluation of technical illustration comprehension, I propose following 3 sub-tasks for TICQA:

Visual Sequence Interpretation (VSI)

VSI involves interpreting assembly, disassembly and step-by-step diagrams to determine the correct order of sequence. This task measures model's visual ability to focus on cues such as arrows, shading, spatial positioning and using it to identify the correct sequence. An example use case is when a user sees an illustration showing how a gear, washer, and bolt fit together. The VLM explains which part goes first, second, and so on, by analysing the arrows and relative positioning in the diagram.

Tools and Components Identification (TCI)

TCI requires identifying parts (e.g., spring, gasket, lever) or tools (e.g., hex wrench, screwdriver) solely from their shapes or visual features and matching them across different views (top, side, cross-section) without relying on text labels. An example use case is when a user points to a part in an exploded diagram and asks, 'Which tool is this?' The VLM visually identifies it and correlates it with a shape in the illustration.

Safety Warning Recognition (SWR)

SWR involves recognising safety warnings or precautions (e.g., wear gloves, disconnect power, fire hazard), whether universal or product-specific, based on symbols, actions, and objects. An example use case is when a diagram includes a small lightning bolt icon near a cable. The VLM indicates that it is an electrical hazard and advises disconnecting power before proceeding.

3.5 Methodology for dataset creation

Overview of dataset creation techniques

Dataset generation approaches for vision-language models typically follow manual, semi-automatic, or fully automatic workflows. Manual generation relies on human annotators—either experts or crowdsourcing platforms such as Amazon MTurk—manually collecting and processing data. Semi-automatic methods automate data collection but retain manual annotation. Fully automatic pipelines automate both collection and annotation using synthetic-data techniques, supplemented by quality-control mechanisms.

Examples:

- Comprehensive datasets such as MMT-Bench (Ying et al., 2024), MMBench (Xu et al., 2023), and MM-Vet (Yu et al., 2023) rely on **expert annotations**.
- Datasets like GQA (Hudson & Manning, 2019), SEED-Bench (B. Li et al., 2023), and CRiC (Gao et al., 2023) generate **synthetic samples**.
- VCR (T. Zhang et al., 2024) and VQA (Goyal et al., 2017) use **crowdsourced annotation** via Amazon MTurk.

Manual expert annotation yields high-quality data but scales slowly. Synthetic-data approaches enable rapid scaling at the cost of lower quality.

Selected Approach

I adopted a semi-automatic pipeline to prioritise dataset quality. A recent analysis of the HellaSwag benchmark (Chizhov et al., 2025) revealed widespread quality issues—typos, low quality questions and multiple invalid MCQ answers—that can misrepresent model performance. To avoid such pitfalls, I focused on data quality over volume.

Dataset Collection and Filtering:

1. Retrieval:
    - Gathered product-manual PDFs from ManualLib and ManualsOnline.
    - Criteria:
        - Diverse product categories and manufacturers.
        - Predominance of illustrations over text.
        - English-language content.
        - Inclusion of safety, assembly, and operation sections.
    - Initial set: 70 manuals (consisting of 2500 pages) covering 40 products.
2. Illustration Extraction:
    - Converted PDFs to images using the pdf2image Python library.

- Fine-tuned a ViT-based object-detection model (google/vit-base-patch16-224-in21k) on 200 manually annotated images.
- Applied the tuned model to crop 3,000 candidate images. Figure 1 presents some of the illustrations cropped by the model.

3. Screening:
    - Manually reviewed crops, discarding 2,750 ambiguous or irrelevant images.
    - Final selection: 250 high-quality illustrations.
4. Processing:
    - Further processed the 250 images according to specific task requirements resulting in 229 images for dataset (some images were combined into a single illustration).

Refer Annex A2 for details on manuals used for the dataset.

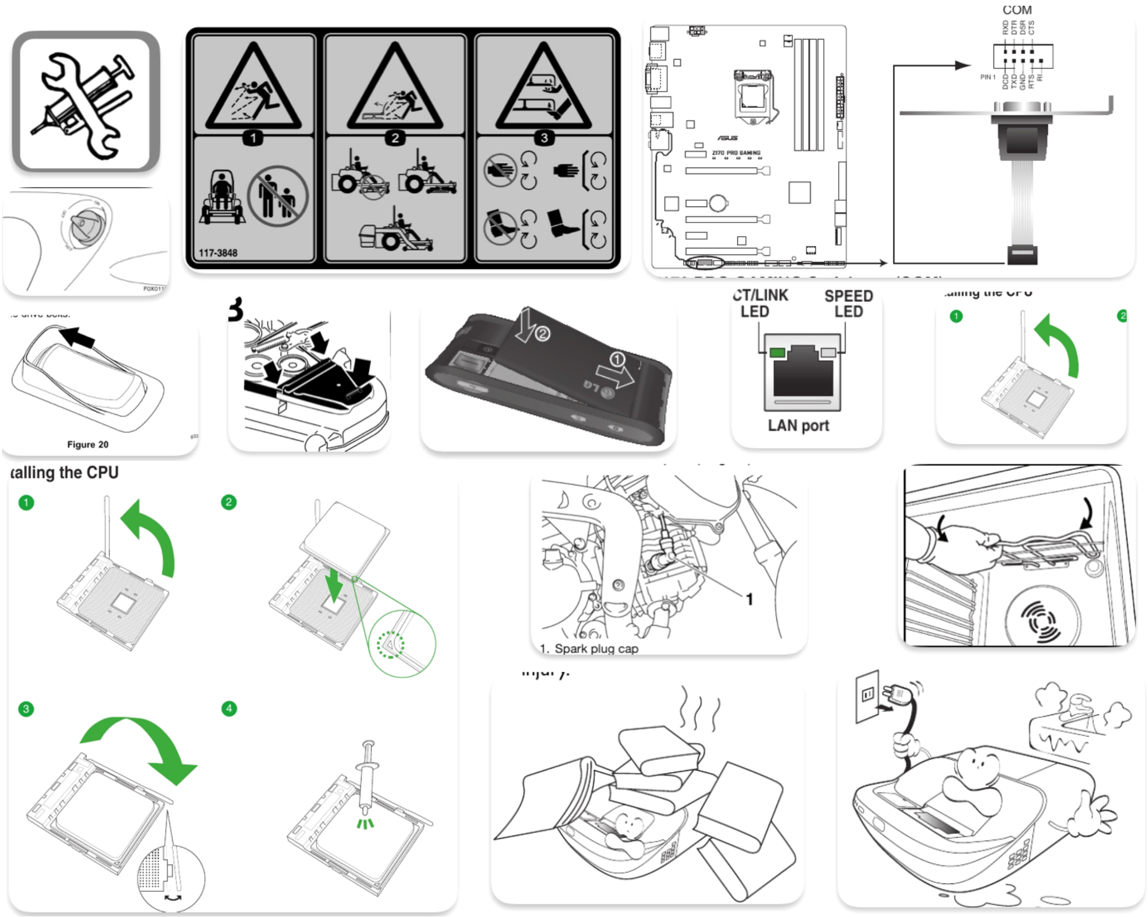


Figure 1: Cropped images from image detection model. (Source: own presentation)

3.6 Sub-tasks Details

For a each sample, an image is paired with a question, ground truth and context. The dataset is divided into 3 subsets based on tasks defined earlier: VSI, TCI and SWR. Figure 2 presents a sample from the dataset for each task.

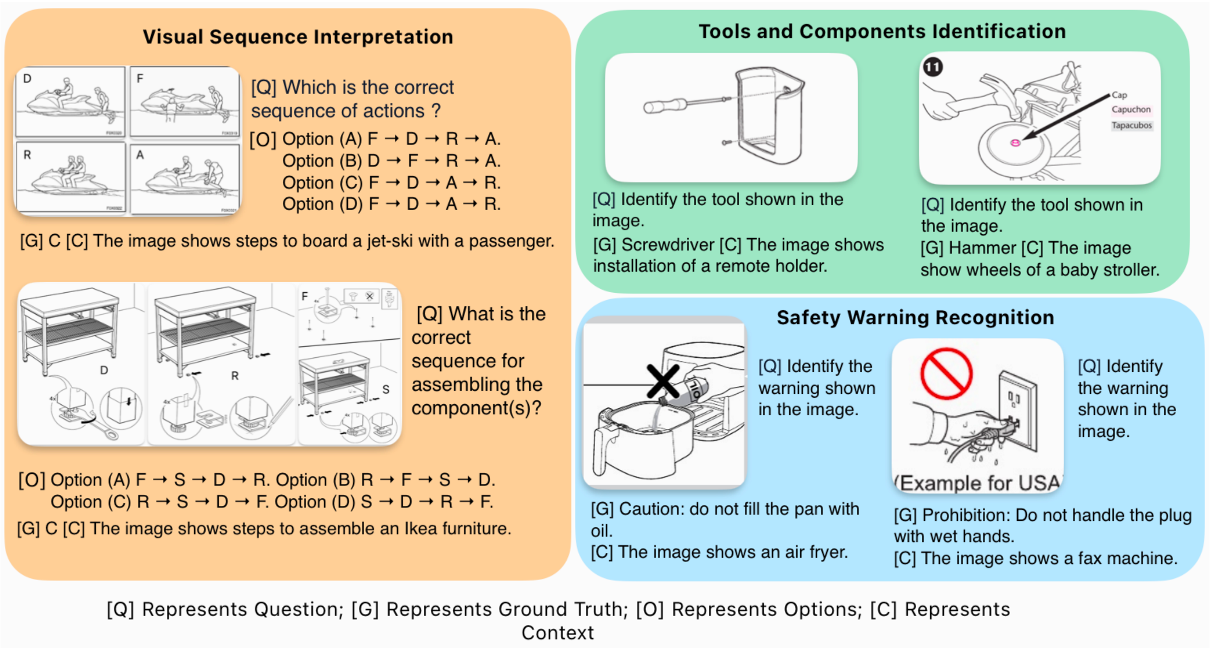
VSI subset:

This subset comprises **50** illustrations—each paired with a contextual information, a ground-truth answer, a set of MCQ options and a question—extracted and post-processed from product manuals.

Key steps and features:

- **Post-processing:** Any text, numbering or other non-visual cues that allows the model to answer the question without visual reasoning are redacted. Numeric labels are replaced with randomly ordered letters to avoid numerical sequence bias.
- **Context:** Supplies necessary metadata (e.g., product name, action performed) to interpret the illustration.
- **Question template:**
  “What is the correct sequence for assembly/disassembly of the [component(s)]? Use the context if needed.”, “What is the correct sequence of actions ?”
  These templates are adapted per illustration to generate relevant questions. For certain illustrations (e.g., exercise_bikes_page_9_crop_0) which do not fit in these templates, appropriate questions are used.
- **Illustration grouping:**
  - **Single image** for most items.
  - **Composite images** (2–3 frames) for complex assemblies (e.g., IKEA products).
- **Response format:** Multiple-choice (A–D). One correct sequence; three distractors crafted to be plausible yet non-obvious.
- **Ground truth:** Manually generated (e.g., Option (A)F → B → N → G. Option (B) F → B → G → N. Option (C) B → G → F → N. Option (D) B → F → G → N.)

The rationale for using an MCQ format instead of open-ended responses is that sequence illustrations in product manuals typically consist of a series of unlabelled images with little or no description (e.g., IKEA instructions). Manually interpreting these sequences risks misalignment or errors, and automatic generation would require access to a large VLM (e.g., GPT-4o), which wasn’t available. By overlay-



Visual Sequence Interpretation

[Q] Which is the correct sequence of actions ?

[O] Option (A) F → D → R → A.
Option (B) D → F → R → A.
Option (C) F → D → A → R.
Option (D) F → D → A → R.

[G] C [C] The image shows steps to board a jet-ski with a passenger.

[Q] What is the correct sequence for assembling the component(s)?

[O] Option (A) F → S → D → R. Option (B) R → F → S → D.
Option (C) R → S → D → F. Option (D) S → D → R → F.

[G] C [C] The image shows steps to assemble an Ikea furniture.

Tools and Components Identification

[Q] Identify the tool shown in the image.
[G] Screwdriver [C] The image shows installation of a remote holder.

[Q] Identify the tool shown in the image.
[G] Hammer [C] The image show wheels of a baby stroller.

Safety Warning Recognition

[Q] Identify the warning shown in the image.
[G] Caution: do not fill the pan with oil.
[C] The image shows an air fryer.

[Q] Identify the warning shown in the image.
[G] Prohibition: Do not handle the plug with wet hands.
[C] The image shows a fax machine.

[Q] Represents Question; [G] Represents Ground Truth; [O] Represents Options; [C] Represents Context

Figure 2: Samples from the 3 tasks in dataset. (Source: own presentation)

ing randomised letter labels on the diagrams and framing MCQs around them, we can generate accurate ground truth quickly and reliably.

TCI subset:

This subset consists of **100** illustrations, each paired with a contextual information, a ground-truth answer, and a question.

Key steps and features:

- **Post-processing:** None, all the illustrations are used as it is from the cropped samples.
- **Context:** Supplies necessary metadata (e.g., product name, action performed) to interpret the illustration.
- **Question template:**
  “Identify the [tool/component/accessary] shown in the image”, “Name the tool to adjust the [item] shown in the image.”
  These templates are adapted per illustration to generate relevant questions.
- **Illustration grouping:**
  - **Tool identification:** 73% of the samples are related to tool identification.

    - **Component/accessary identification:** 27% of the samples are related to component or accessory identification.

- **Response format:** One word (or multiple words in case of tools).
- **Ground truth:** One word (or multiple words in case of tools), curated from manual text descriptions. (e.g., Screwdriver, Hammer, etc.)

SWR subset:

This subset is composed of **100** illustrations of different safety warnings, precautions and prohibitions present in product manuals.

Key steps and features:

- **Post-processing:** None, all the illustrations are used as it is from the cropped samples.
- **Context:** Supplies necessary metadata (e.g., product name, action performed) to interpret the illustration.
- **Question template:**
  "What safety warning or hazard is shown in this image?"
- **Illustration grouping:**
    - **Warning:** 22% of the samples are related to warning (serious or extreme hazard) identification.
    - **Caution:** 6% of the samples are related to caution (less serious hazard) identification.
    - **Prohibition:** 72% of the samples are related to prohibition (actions not permitted) identification.
- **Response format:** One to two sentence description of the hazard.
- **Ground truth:** One to two sentence description of the hazard. (e.g., Always turn off the generator when refuelling.)

3.7 Quality Assurance Procedures

To maintain a high-quality dataset, following checks were applied:

1. Text Redaction

- Removed all non-essential text (labels, captions) from images so VLMs forced to reply on visual cues for answer. This was verified via trial runs that any remaining text could cue the answer, and therefore redacted accordingly.

2. Visual Clarity

   - Inspected every image for blur, occlusion, or excessively small details. Retained only those with clear, legible visual cues.

3. Contextual Metadata

   - Supplied brief context (e.g. product/component name, action performed) when image content alone could be ambiguous to prevent hallucination and anchors the model's interpretation.

4. Prompt Engineering

   - One would assume that providing an image would force VLMs to answer according to the image, however it was observed during trial runs that sometimes model would give answers based on their language understanding rather than from the input image. Hence, to avoid this, questions are framed to explicitly direct attention to the image (e.g. include "shown in the image" phrase).

Together, these steps guarantee that model performance reflects true visual comprehension of technical illustrations, avoiding responses from language or common-sense knowledge.

This chapter, introduced the TICQA dataset, motivated by the shortcomings of existing VQA and comprehensive benchmarks for technical illustrations, and established six core design principles—comprehensive diagram coverage, inference-driven questions, hierarchical task structure, strict redaction, precise ground truth, and domain diversity. I then detailed the semi-automatic creation pipeline—combining manual selection of product manuals, ViT-based object detection, and rigorous quality checks—to curate 250 high-quality QA pairs. Building on this foundation, I defined three focused subtasks (Visual Sequence Interpretation, Tools & Components Identification, Safety Warning Recognition) and adopted an MCQ format along with single word and few sentence description to isolate visual reasoning. This dataset not only fills a critical gap in technical-illustration QA but also provides the robust benchmark on which the remainder of this thesis will build: in the chapters ahead, I will use TICQA to evaluate current vision-language models and identify their weaknesses across each subtask.

4 Evaluation of VLMs

This chapter presents details on experiments conducted to test VLMs on TICQA dataset. First, I discuss the evaluation environment settings, then I present the model choice criteria and details of the VLMs chosen. Then I discuss 2 experiment settings: with and without context, followed by evaluation mechanism for each task.

4.1 Environment Settings

To conduct experiments in this study, JupyterHub environment provided by Institute of Information Systems is utilised. The GPU used for this study is Nvidia A100 40GB. The software configuration is as follows: CUDA 12.1, PyTorch 2.5, Python 3.11 and Huggingface transformers v4.51.3. While generating responses from VLMs, the sampling parameter (do_sample) was set to false, to generate highly deterministic responses.

4.2 Model choice

With rapid progress in the field, new VLMs appear almost weekly. To keep pace, the research community has created several leaderboards—e.g., the Open VLM Leaderboard by OpenCompass (*Open VLM Leaderboard - a Hugging Face Space by Opencompass*, n.d.), Vision Arena by WildVision, and the SEED-Bench Leaderboard by TencentAILab. I chose the Open VLM Leaderboard because it is the most popular and up-to-date, reporting scores on benchmarks such as MMBench and MMStar. To fit available 40 GB GPU, I restrict the study to models under 20 billion parameters. Furthermore, I include only the 10 highest-ranked non-quantised models to ensure a representative set of state-of-the-art VLMs. The 10 selected benchmarks are as follows:

Benchmarks selected overview:

1. **MMBench_V11**: A large-scale, objective evaluation benchmark for assessing the multimodal understanding capabilities of VLM. It contains over 3k MCQ questions spanning 20 fine-grained ability dimensions (Xu et al., n.d.).

2. **MMStar**: A high-quality, vision-indispensable multi-modal benchmark designed to rigorously evaluate the capabilities of VLMs. It contains 1.5k meticulously curated samples, each sample requires genuine visual reasoning (L. Chen et al., 2024).

3. **MMMU_Val**: The validation split of MMMU benchmark, targeting expert-level reasoning in disciplines like engineering, medicine and humanities. It uses college-level diagrams, charts, and equati-

ons to asses deliberate reasoning beyond basic perception. With 11k+ questions, MMMU_Val stresses domain-specific knowledge integration (Yue et al., 2023).

4. **OCRBench**: Specialising in text localisation and reasoning, OCRBench, consists of 29 datasets on OCR tasks like text recognition, scene-text centric VQA, document oriented VQA, key information extraction and handwritten mathematical expression recognition (Y. Liu et al., 2024).

5. **AI2D**: AI2D evaluates diagram understanding through 5k+ science diagrams annotated with blobs, arrows, and text boxes. Its focus on syntactic parsing and multi-choice questions tests model’s ability to interpret educational graphics, though limited tasks diversity restricts broader applicability (Hiippala et al., 2021).

6. **HallusionBench**: This benchmark targets hallucination detection. It employs 1.1k yes/no questions to identify contradictions between images and generated text (Guan et al., 2024).

7. **SEEDBench_IMG**: A subset of SEEDBench containing 19k samples, it assesses general visual reasoning through synthetic and human-annotated data (B. Li et al., 2023).

8. **MMVET**: MMVET evaluates VLMs along 6 foundational vision-language capabilities including recognition, knowledge, spatial awareness and OCR (Yu et al., 2023).

9. **RealWorldQA**: This benchmark focuses on practical visual QA, using real-world images like product labels and street signs to simulate user applications (*Xai-Org/RealworldQA · Datasets at Hugging Face*, n.d.).

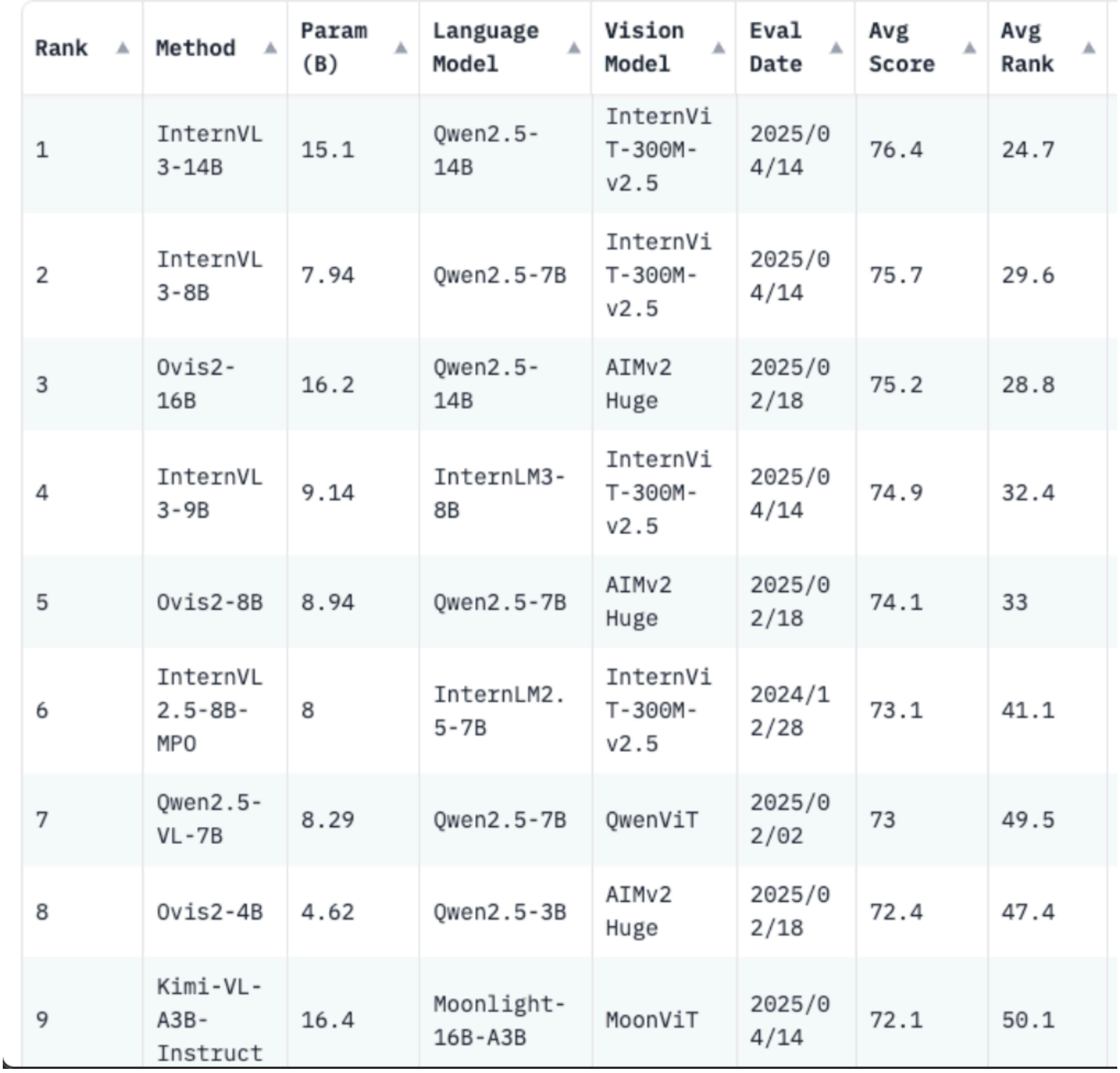
10. **POPE**: Like HallusionBench this benchmark also targets hallucination in multimodal language models. However it uses 9k adversarial samples to quantify object hallucination rates, using precision-oriented metrics (Y. Li et al., 2023).

4.3 Selected Models

Based on the above criteria, Figure 3 shows available VLMs on OpenVLM leaderboard as of 20th April 2025. Note that I used the AWQ-quantised variant of InternVL3-14, as the full-precision version would not fit on a 40 GB GPU. Table x summarises the details of selected models. Below is a brief overview of each of the selected models.

InternVL3-14B, 8B and 9B (J. Zhu et al., 2025)

Developed by OpenGVLab, internVl3-14B is the latest model in the InternVL series released on 11th April 2025. It uses Variable Visual Position Encoding (V2PE) to enhance long-context understanding. Supports tool usage, GUI agents, industrial image analysis and 3D vision. For architecture it uses “ViT-MLP-LLM” paradigm with dynamic resolution (448*448 tiles) and multi-image video support. For language part is utilises Qwen2.5-14B and InternViT-300M-v2.5 for vision part. The 8B parameter variant uses Qwen2.5-7B for language part while 9B parameter variant utilises InternLM3-8b-instruct.



Rank ▲	Method ▲	Param (B) ▲	Language Model ▲	Vision Model ▲	Eval Date ▲	Avg Score ▲	Avg Rank ▲
1	InternVL3-14B	15.1	Qwen2.5-14B	InternViT-300M-v2.5	2025/04/14	76.4	24.7
2	InternVL3-8B	7.94	Qwen2.5-7B	InternViT-300M-v2.5	2025/04/14	75.7	29.6
3	Ovis2-16B	16.2	Qwen2.5-14B	AIMv2 Huge	2025/02/18	75.2	28.8
4	InternVL3-9B	9.14	InternLM3-8B	InternViT-300M-v2.5	2025/04/14	74.9	32.4
5	Ovis2-8B	8.94	Qwen2.5-7B	AIMv2 Huge	2025/02/18	74.1	33
6	InternVL2.5-8B-MPO	8	InternLM2.5-7B	InternViT-300M-v2.5	2024/12/28	73.1	41.1
7	Qwen2.5-VL-7B	8.29	Qwen2.5-7B	QwenViT	2025/02/02	73	49.5
8	Ovis2-4B	4.62	Qwen2.5-3B	AIMv2 Huge	2025/02/18	72.4	47.4
9	Kimi-VL-A3B-Instruct	16.4	Moonlight-16B-A3B	MoonViT	2025/04/14	72.1	50.1

Figure 3: Top 9 open sourced Model on Open VLM leaderboard under 20B parameters

InternVL2.5-8B (Z. Chen et al., 2024)

Predecessor to InternVL3, this VLM was release on 15th November 2025. It introduced signifiant enhancements in training and testing strategies as well as data quality. It uses InternLM2.5-7B as language model and InternViT-300M-v2.5 as vision model .

Kimi-VL-A3B-Instruct (Team et al., 2025)

Release most recently by MonshotAI team on 10th April 2025, Kimi is first in their series of VLMs. It differs from all other VLMs in architecture also as it uses Mixture-of-Experts (MoE) with total 16B parameters, 3B activated for inference. It also has long-context window of 128k tokens for long videos/ documents. It is optimised for OCR, agent tasks and multimodal reasoning.

Qwen2.5-VL-7B

Qwen2.5 is part of Qwen series developed by researchers at Alibaba cloud released in January 2025. Its key features include recognising common objects such as flowers, animals along with highly accurate analysis of texts, charts, icons, graphics and layouts within images. It is agent in nature, which means it can reason and dynamically use direct tools. It can also localise objects in an image by generating bounding boxes or points and provide JSON output attributes along with coordinates. It supports structured output generation. It uses Qwen2.5-7B as language model along with QwenViT as vision model.

Ovis2-16B, 8B and 4B (Lu et al., 2024)

Introduced by AI team at Alibaba International Digital Commerce Group on 31st May 2025. Its key features include optimised training strategies for small-scale models enabling them to achieve higher capacity density, enhanced Chain-of-Thought (CoT) reasoning abilities through combination of instruction tying and preference learning, video and multi-image processing, multilingual support beyond English and Chinese and improved structured data extraction from complex visual elements like tables and charts. The 16B and 8B variant uses Qwen2.5-14B as language model and AIMv2 Huge as Vision model, while 4B variant utilises Qwen2.5-3B as language model.

4.5 Experiments

A study in late 2024 (Amara, K. et al., 2024) investigated the influence of context (i.e. external information apart from the input image) in VQA. They found that "complimentary information improves answer and reasoning quality, while contradictory information harms model performance and confidence". Inspired from these results, I included contextual information in the dataset and performed evaluation experiments in two settings: **with and without contextual information** to understand the effects of external context information on VLM. The results of these experiments can help to verify if external information is really needed for technical illustration comprehension in VLMs. In response, VLM is asked to provide answer and rational behind the answer (i.e. reasoning) in specified structured JSON format. Below I discuss the experiment details including the prompt template.

Experiment 1: With Context

For experiment 1, external context was added along with the question and instructions in VLM prompt. The prompt template used for each sub-task is as follows:

Prompt template for VSI:

```
[Question]

Context: [Context]

Options: [Options]

Choose from the given options. Explain your reasoning. Take help of context if needed.

Note:

- Respond only in valid json.

- Example Response: {'answer': 'A', 'reasoning': 'Explain your reasoning here.' }
```

Prompt template for SWR:

```
[Question]

Take help of extra information if needed. Explain your reasoning.

Note:

- Respond only in valid json.

- Example Response: {'answer': 'Warning. Do not pull the plug', 'reasoning': 'Explain your reasoning here’}.

Extra Information: [context]
```

Prompt template for TCI:

```
[Question]

Context: [Context].
```

Take help of context if needed.

Note:

- *Respond only in valid json.*
- *Example Response: {'answer': 'Tool/Accessary name', 'reasoning': 'Explain your reasoning here'}*

```

**Experiment 2: Without Context**

For experiment 2, no external context was provided along with the question and instructions in VLM prompt. The prompt template used for each sub-task is as follows:

**Prompt template for VSI:**

```

[Question]

Options: [Options]

Choose from the given options. Explain your reasoning.

Note:

- *Respond only in valid json.*
- *Example Response: {'answer': 'A', 'reasoning': 'Explain your reasoning here.' }*

```

**Prompt template for SWR:**

```

[Question] Explain your reasoning.

Note:

- *Respond only in valid json.*
- *Example Response: {'answer': 'Warning. Do not pull the plug', 'reasoning': 'Explain your reasoning here'}.*

~~~
```
~~~

Prompt template for TCI:

~~~
```
[Question]
Note:
- Respond only in valid json.
- Example Response: {'answer': 'Tool/Accessary name', 'reasoning': 'Explain your reasoning here'}
```
~~~

4.6 Evaluation Methodology

For evaluation of VLM response 2 different techniques: **exact match** for MCQs and **semantic match** using PoLL for open-ended question are used. Below are the evaluation details per task.

Evaluation of VSI task response:

For the VSI, each model must output the correct MCQ option (A, B, C, or D). I evaluate the performance using exact-match to calculate accuracy: each VLM is prompted with the image, question, answer choices, and context, and asked to return the chosen option along with its reasoning. Exact matches between the model's prediction and the ground truth yield the accuracy score, while the provided rational is used to verify the response manually for a small subset of examples.

Evaluation of SWR task response:

For the SWR, each VLM must produce an open-ended description of the safety warning shown in the illustration. While manual match of answer and ground truth can be a natural choice for evaluation, it is time-consuming, prone to bias and inconsistent; single-LLM evaluation also suffers from bias and high cost . Therefore, I adopt the PoLL (Panel of LLMs) technique [cite] , leveraging three medium-sized, open-source models—phi-4 (14B, AWQ), Mistral-Small-24B-Instruct (AWQ), and Qwen2.5-32B-Instruct-AWQ—as judges.

Evaluation procedure:

1. **Extract answer and reasoning:** Prompt the VLM with to generate structured output, extract its answer and reasoning from the response.

2. **Judge the answer:** For each response, provide the answer, ground truth to each judge model, asking it to label the response as “True” (i.e. correct) or “False” (i.e. incorrect) based on semantic similarities. Ask the judges to also provide their rational for evaluation, which is later used to verify the evaluation manually for a small subset of examples.

3. **Aggregate the evaluation of panel:** Apply max-pooling over the three binary judgements to determine the final correctness and compute accuracy.

4. **Validation:** Review a small subset of evaluations to verify evaluation consistency.

Prompt for LLM judge:

```
You will be given a Reference answer and a Provided Answer. Judge whether the Provided Answer is correct by comparing it to the Reference Answer. The answer’s question is related to safety information, hence while evaluating judge on the basis of semantic meaning and not exact match. Extra information is ok. If the Provided Answer is correct say exactly "True", otherwise say "False". Provide response in a valid following json structure {'answer': 'your answer', 'reason': 'your reason'}
```

Evaluation of TCI task response:

For the TCI subset, each model must output a concise name (one to two words) for the tool or component shown. Although string-based metrics like ROUGE-L or BLEU could be applied, they struggle with synonymous labels (e.g., “hex wrench” vs. “Allen key,” “Phillips screwdriver” vs. “cross-head screwdriver”). Hence, I again employ the PoLL (Panel of LLMs) technique with the same three judges —phi-4 (14B, AWQ), Mistral-Small-24B-Instruct (AWQ), and Qwen2.5-32B-Instruct-AWQ. The same evaluation pipeline consisting of answer and reasoning — extraction, answer evaluation based on semantic similarities, aggression of binary evaluations and validation —is used, with difference in prompt template for judge.

Prompt for LLM judge:

```
You will be given a Reference answer and a Provided Answer. Judge whether the Provided Answer is correct by comparing it to the Reference Answer. Differently formatted answer, similar names, and alternative spellings should all be considered the same. Extra information is ok. If the Provided An-
```

```
swer is correct say exactly "True", otherwise say "False". Provide response in a valid following json structure {'answer': 'your answer', 'reason': 'your reason’}
```

4.7 Human Baseline

Understanding how humans perform on a given dataset and comparing it with VLM’s performance could reveal hidden traits and bias. For this reason, I human baseline is also generated for TICQA dataset. For generating the baseline, group of 4 students (to moderate bias) are given the dataset converted into MCQ format. They are asked to choose the correct option for all the 250 samples. The human accuracy baseline is created by taking the average of scores in each task. Single experiment is conducted for human baseline — with context, since the focus was on understanding influence of contextual information in VLMs, not humans.

In this chapter, I described the evaluated of nine leading vision-language models across the three TICQA subtasks—Visual Sequence Interpretation, Tools & Components Identification, and Safety Warning Recognition—using tailored metrics for each. For sequence tasks, I measured exact-match accuracy on MCQs. For tool identification and safety warnings, I applied the PoLL (Panel of LLMs) approach, aggregating judgments from three medium-sized language models to accommodate synonymous labels and open-ended descriptions.

A comprehensive analysis of these evaluation results will be presented in the next chapter. There, I will delve into performance breakdowns by subtask, model and experiment setting.

5 Results and Analysis

This chapter presents results of VLM evaluation on TICQA dataset. It also discusses different analysis of the results.

5.1 Results

The table 1 compares nine vision-language models (VLMs) plus a human baseline on three subtasks—Tool and Component Identification (TCI), Safety Warning Recognition (SWR) and Visual Sequence Interpretation (VSI)—reporting both "with context" and "without context" accuracies as well as weighted overall score. Reported average score from OpenVLM leaderboard on selected benchmarks (MMBench_V11, MMStar, MMMU_Val, OCRbench, AI2D, HallusionBench, MMVet, SeedBench_IMG, RealWorldQA, Pope) is also included for SOTA performance reference.

Under contextual prompting, Ovis2-16B leads the group with a 50.0% overall accuracy (52.0/40.0/66.0 on TCI/SWR/VSI), while the smallest model, Internvl2.5-8b-MPO, trails at just 25.2%. Removing context causes a modest drop for all models: Ovis2-16B falls to 45.6% overall and Internvl2.5-8b-MPO to 23.2%. Notably, Kimi-VL-A3B-Instruct excels on TCI (49.0%) but suffers on SWR (19.0%), and Qwen2.5-7B-Instruct achieves the highest VSI score (70.0%) with context. Despite these advances, all models remain well below the human baseline of 87.6% overall accuracy.

At a glance, the **Human Baseline** establishes a robust performance on TCI (95.8 %) and SWR (98.5 %) accuracies, significantly outperforming all VLMs. However, in the VSI accuracy, the Human Baseline unexpectedly registers the lowest at 38.0 %. Among the VLMs, **Ovis2-16B** leads the group with 50.0 % and 45.6 % overall accuracy in both with and without the context settings. Similarly, **Ovis2-8B** closely follows in with the context settings with 47.6 %, but falls behind **Ovis2-4B** (40.4 %) in without the context experiment.

In the VSI task, **Qwen2.5-7B-Instruct** tops the "with context" setting at 70.0%, followed closely by **Ovis2-8B** at 68.0% and **Ovis2-16B** at 66.0%; smaller models lag behind (e.g. Internvl2.5-8b-MPO at 50.0%). Without context, **Ovis2-16B** leads at 68.0%, with **Qwen2.5-7B** at 66.0% and **InternVL3-8B** at 64.0%, while **Ovis2-4B** dips lowest to 48.0%**.** With context, **Kimi-VL-A3B-Instruc**t (42.0 %) is the worst performer.

In the TCI, Ovis2-16B and Kimi-VL-A3B-Instruct are the clear leaders: Ovis2-16B scores 52.0 % with context (46.0 % without), and Kimi-VL-A3B-Instruct follows at 49.0 % (43.0 % without). Both InternVL3-8B and Ovis2-8B sit in the middle (28.0 % , 20.0 % and 49.0 % , 43.0 % with and without respectively), while the lightest model, Internvl2.5-8b-MPO, lags far behind at just 19.0 % with context (17.0 % without).

In SWR task, **Ovis2-16B** leads the group at 40.0 % with context (34.0 % without), closely followed by **Ovis2-8B** at 36.0 % (25.0 % without) and **InternVL3-8B**, which jumps from 18.0 % without to 32.0 % with context. **Qwen2.5-7B-Instruct** and **Ovis2-4B** land in the low-30s under context, but **Kimi-VL-A3B-Instruct** remains stuck at 19.0 % (actually improving to 23.0 % when context is removed).

Figure 3. Shows a scatter plot for comparing TICQA scores and leaderboard scores.

5.2 Analysis

Below I present a deeper analysis of the above results.

5.2.1 With and without context performance gains

Analysing the accuracies of with and without context settings reveals a mixed pattern. The biggest gains because of contextual information are in **Qwen2.5-7B-Instruct** (gains +4% on TCI, VSI and +17% on VSI, boosting overall by +9.2%) while **Kimi-VL-A3B-Instruct** actually loses accuracy in SWR (−4%) and VSI (−16%), resulting in a −2.4% drop overall despite +6% on TCI.

Top beneficiaries of context:

Qwen2.5-7B-Instruct sees the largest single jump +17 pt gain on Safety Warning Recognition (17 to 34 %), plus solid +6 pt jumps on both Tools and Components Identification and Visual Sequence Interpretation, yielding the highest overall lift at +9.2 pts. In other words, context almost doubles its ability to recognise safety warnings and meaningfully helps with identifying tools/components and interpreting visual sequences.

InternVL3-8B exhibits a massive gain on Safety Warning Recognition (+14 pts, from 18 % to 32 %), along with an +8 pt gain on Tools and Components Identification (20 to 28 %). Those lifts drive its overall accuracy up by +7.6 pts—even though its Visual Sequence Interpretation score dips by 6 pts. This suggests InternVL3-8B leverages context most effectively for spotting safety warnings, while potentially over-fitting or misinterpreting when piecing together sequential visual steps.

Ovis2.8B also shows a noticeable +11 pt gain on Safety Warning Recognition (25 to 36 %) and picks up +6 pts on Tools & Components Identification (49 to 55 %) as well as +8 pts on Visual Sequence Interpretation (60 to 68 %), yielding an overall boost of +8.4 pts. This show that Ovis2-8B is good at leveraging contextual cues across all facets of technical illustration comprehension—boosting not just low-level recognition of safety warnings and tool components but also higher-level visual sequence interpretation—making it one of the most context-sensitive models in the suite.

Model	With context				Without context				LS*
	TCI	SWR	VSI	O*	TCI	SWR	VSI	O*	
Internvl2.5-8b-MPO	19.0	19.0	50.0	25.2	17.0	16.0	50.0	23.2	73.1
InternVL3-14B-AWQ	24.0	25.0	62.0	32.0	14.0	25.0	58.0	27.2	**76.4**
InternVL3-8B	28.0	32.0	58.0	35.6	20.0	18.0	64.0	28.0	75.7
InternVL3-9B	25.0	29.0	60.0	33.6	21.0	20.0	62.0	28.0	74.9
Kimi-VL-A3B-Instruct	49.0	19.0	42.0	35.6	43.0	23.0	58.0	38.0	72.1
Ovis2-16B	**52.0**	**40.0**	66.0	**50.0**	**46.0**	**34.0**	**68.0**	**45.6**	75.2
Ovis2-4B	48.0	33.0	44.0	41.2	43.0	**34.0**	48.0	40.4	72.4
Ovis2-8B	49.0	36.0	68.0	47.6	43.0	25.0	60.0	39.2	74.1
Qwen2.5-7B-Instruct	35.0	34.0	**70.0**	41.6	31.0	17.0	66.0	32.4	73.0
Human baseline	95.7	**98.5**	37.0	87.6	95.7	**98.5**	37.0	87.6	NA

Table 1: Accuracy score of VLMs. (Source: own presentation)

All the values are in %. Highest values are marked in bold and lowest values are underlined. *O represents overall accuracy calculated as weighted average. Highest scores are highlighted in bold, while lowest are underlined. *LS represents OpenVLM leaderboard average score on selected benchmarks. Leaderboard scores are from 20th April 2025.

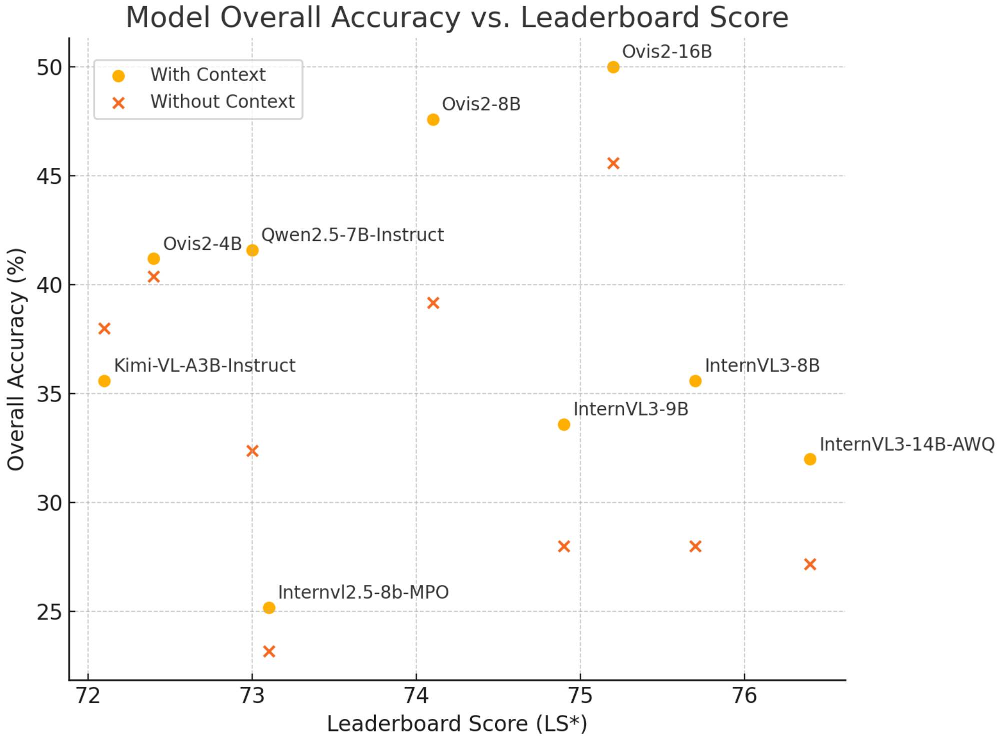


Figure 4: Scatter plot for model overall accuracy vs leaderboard score. (Source: own presentation)

Moderate, balanced gains:

InternVL3-14B-AWQ picks up +10 pts on Tools and Components Identification (14 to 24 %) and +4 pts on Visual Sequence Interpretation (58 to 62 %), while Safety Warning Recognition remains flat, for a net +4.8 pt overall increase. This model uses context primarily to sharpen its concept identification and sequential interpretation, without affecting warning recognition.

InternVL3-9B enjoys +4 pts on Tools and Components Identification and +9 pts on Safety Warning Recognition but loses 2 pts on Visual Sequence Interpretation, for a +5.6 pt overall gain. Its context advantage is clearly in spotting warnings and basic components rather than chaining visual steps.

Ovis2-16B benefits equally on Tools and Components Identification and Safety Warning Recognition (+6 pts each) but falls 2 pts on Visual Sequence Interpretation, netting +4.4 pts overall. Like InternVL3-9B, context mainly aids low-level recognition tasks.

Model	TCI gain	SWR gain	VSI gain	Overall gain
Internvl2.5-8b-MPO	*2.0*	3.0	0.0	2.0
InternVL3-14B-AWQ	***10.0***	0.0	4.0	4.8
InternVL3-8B	8.0	14.0	-6.0	7.6
InternVL3-9B	4.0	9.0	-2.0	5.6
Kimi-VL-A3B-Instruct	6.0	*-4.0*	*-16.0*	*-2.4*
Ovis2-16B	6.0	6.0	-2.0	4.4
Ovis2-4B	5.0	-1.0	-4.0	0.8
Ovis2-8B	6.0	11.0	***8.0***	8.4
Qwen2.5-7B-Instruct	4.0	***17.0***	4.0	***9.2***

Table 2: Accuracy gains in VLMs with and without context. (Source: own presentation)

Small or negative effects:

Internvl2.5-8b-MPO shows only a +2 pt lift on Tools and Components Identification and +3 pts on Safety Warning Recognition, with zero change in Visual Sequence Interpretation—yielding a modest +2 pt overall gain. Its representational capacity may be too limited to make full use of added context.

Ovis2-4B barely budges: +5 pts on Tools and Components Identification, –1 pt on Safety Warning Recognition, –4 pts on Visual Sequence Interpretation, for a mere +0.8 pt overall. Its mixed responses suggest that context can sometimes confuse smaller or less refined architectures.

Kimi-VL-A3B-Instruct is the only model to see an overall drop (–2.4 pts). Although it jumps +6 pts on Tools and Components Identification with context, it plunges –4 pts on Safety Warning Recognition

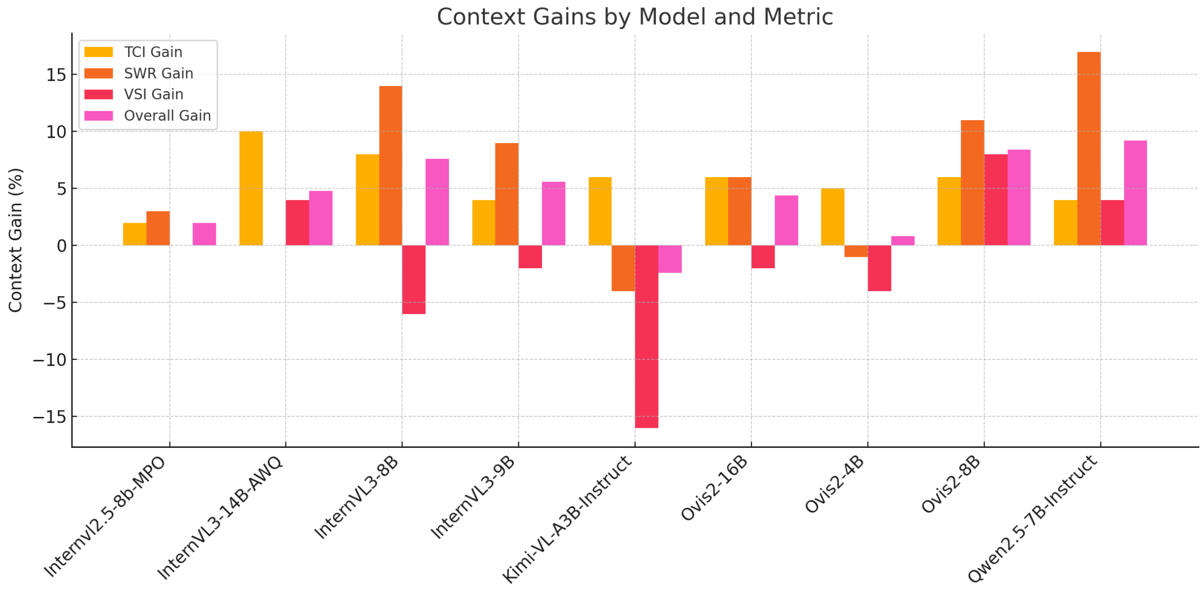


Figure 5: Bar chart representation of context gain analysis. (Source: own presentation)

and a dramatic –16 pts on Visual Sequence Interpretation—indicating that extra background text actually derails its ability to recognise warnings and correctly interpret visual sequences.

To summarise, Safety Warning Recognition benefits most from additional context, showing dramatic improvements across several models. In contrast, Visual Sequence Interpretation is the most challenging task; several models perform worse with extra context, suggesting that too much text can mislead their step-by-step reasoning. Model capacity and architecture strongly influence context sensitivity: larger, more advanced VLMs almost always improve, whereas smaller models often experience negligible or negative effects.

5.2.2 Model characteristics vs overall accuracy

I analysed the model performance on the dataset across model size. Fig. X presents a scatter plot of model size vs overall accuracy.

From the plot it is clearly visible that there is a mixed effect of parameter size and architecture+training on the accuracy. Even through Ovis2-16B having the largest size (16B parameters) performs best (50% accuracy), Kimi-VL-A3B-Instruct (16.4B parameters) and InternVL3-14B-AWQ perform average (38% and 32% respectively). While smaller model Ovis2-4B (4.62B parameters), Ovis2-8B (8B parameters) perform equally or better than larger models InternVL3-8B (7.94), InternVL3-9B (9.14B parameters) and InternVL2.5-8B-MPO (8B parameters). It is also evident that models from same family but newer version (InternVL2.5 and InternVL3 series) perform better, as expected. Special attention should be given to Ovis and internVL series. Smaller or same Ovis models (4B and 8B) score better than InternVL models (8B, 9B) in the same parameters range. This may be attributed to the training differences, as Ovis2 utilised instruction tuning and preference learning to boost reasoning, especially

Chain-of-Thought (CoT) abilities while InternVL adopts a native multimodal pre-training on text and vision data jointly from scratch approach.

5.2.3 Model performance on sub-tasks

Fig. X shows heatmap of each model's accuracy **with context** across the three subtasks—TCI, SWR, and VSI. Visual Sequence Interpretation (VSI) emerges as the strongest subtask, led by Qwen2.5-7B (70 %), Ovis2-8B (68 %), and Ovis2-16B (66 %). Even the weakest VSI model (Kimi-VL at 42 %) outperforms most models on the other tasks. Tools & Components Identification (TCI) exhibits a broader performance range: Ovis2-16B (52 %) and Kimi-VL (49 %) excel, InternVL3-8B/9B hover around 25–28 %, and Internvl2.5-8B-MPO trails at 19 %. Safety Warning Recognition (SWR) proves most challenging overall, with Ovis2-16B (40 %), Ovis2-8B (36 %), and InternVL3-8B (32 %) at the top, while Kimi-VL slips to just 19 %.

Across these tasks, the Ovis2 series stands out for its balanced performance, particularly at 8 B and 16 B. Qwen2.5-7B specialises in sequence reasoning, trading off mid-range TCI/SWR performance (32–34 %) for the highest VSI score. Kimi-VL inverts the usual pattern—strong in TCI but weak in SWR and only moderate in VSI. InternVL3 variants occupy the middle ground with consistent, if unspectacular, results, while Internvl2.5-8B-MPO's limited capacity is reflected in its uniformly low scores. It is safe to conclude that the easiest task is **Visual Sequence Interpretation**—its average accuracy hovers around 58 % (42–70 %)—while the hardest is **Safety Warning Recognition**, with average scores near 29 % (19–40 %).

In this chapter, I presented different analysis of the evaluation results. The analysis included are performance gains with and without context, model characteristics vs overall accuracy and model performance on sub-tasks. These analysis reveal key findings presented in the next chapter.

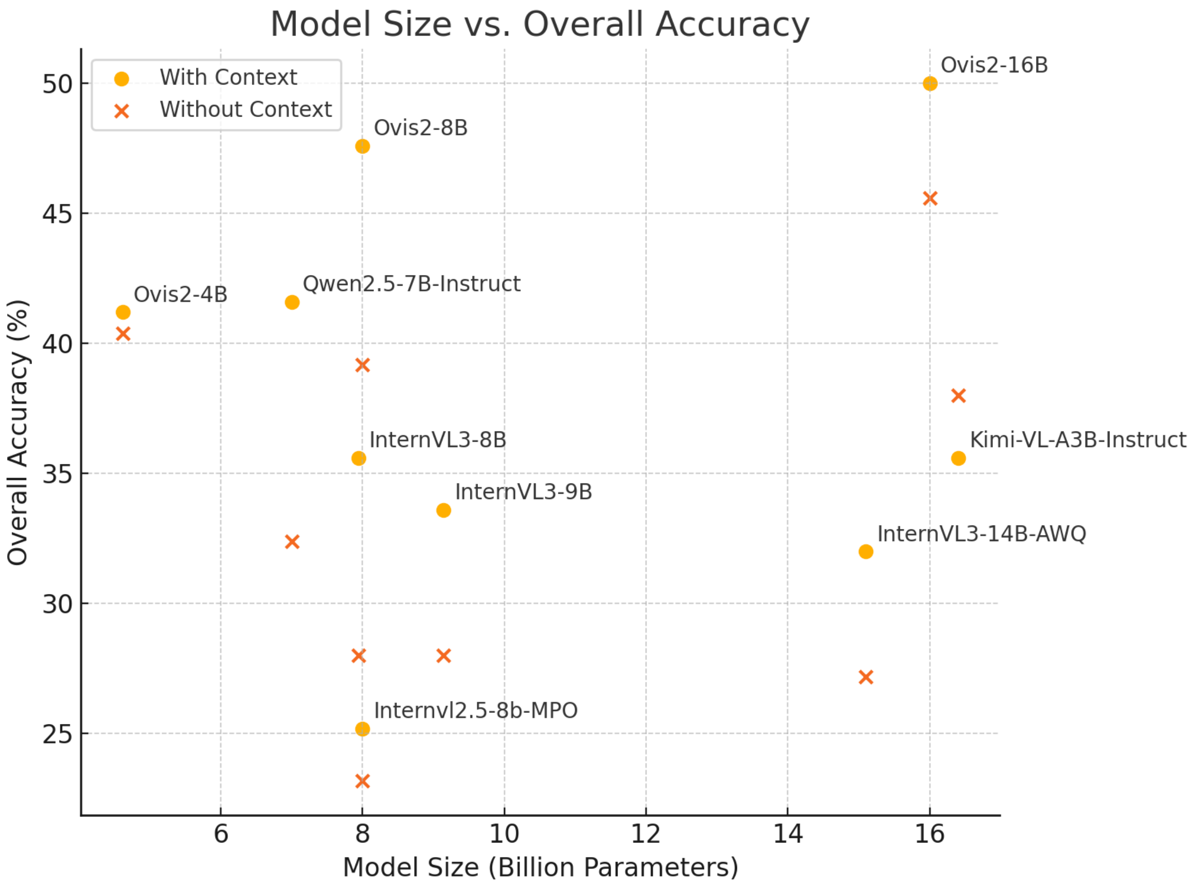


Figure 6: Scatter plot of model size vs overall accuracy. Circle markers are used for with context accuracy and cross markers for without-context accuracy. (Source: own presentation)

7. Findings

This chapter presents the key findings of the work based on the results and analysis of nine VLMs evaluation on TICQA.

7.1 Finding 1

Providing context to VLMs has mixed effects

Adding textual context generally boosts performance on recognition tasks—particularly Safety Warning Recognition and Tools & Components Identification—but can hinder models on Visual Sequence Interpretation when they're overloaded with extra text. The effect is capacity-dependent: larger, instruction-tuned architectures typically use, whereas smaller models often see negligible or even negative impacts. Moreover, no model benefits uniformly; each displays a unique trade-off between improved low-level recognition and high-level reasoning. These findings point to an real-world use strategy: tailor context length and prompt design to the specific model and subtask—for example, provide rich background when emphasising safety warnings, but streamline prompts when accurate sequence interpretation is required.

7.2 Finding 2

New architecture and training strategies improve model performance

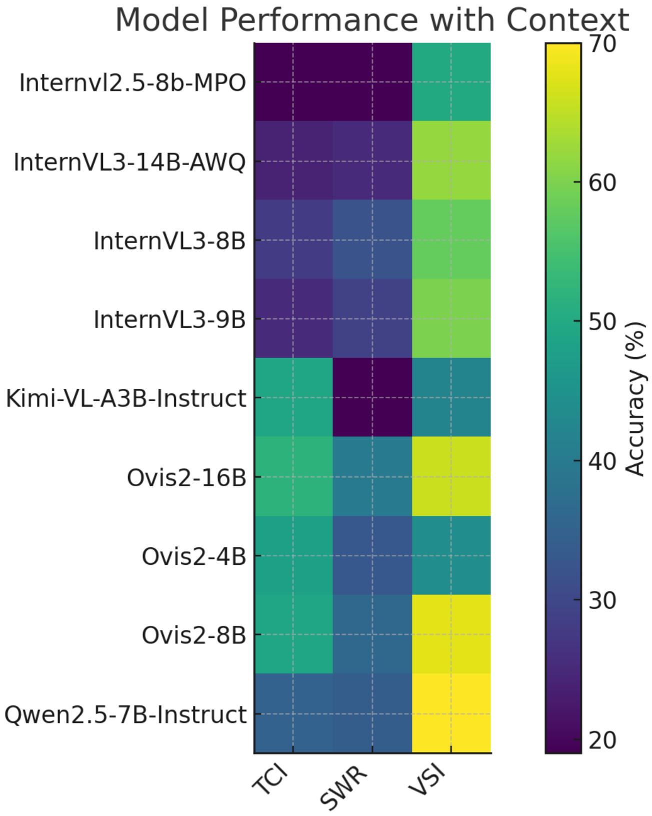


Figure 7: Heatmap of model performance with context across tasks. Darker cells indicate lower scores and brighter cells higher scores. (Source: own presentation)

Models with new training strategies like instruction tuning and preference learning to boost reasoning, especially Chain-of-Thought (CoT) abilities perform better than models relying on pre-training on joint (text + visual) data from scratch.

7.3 Finding 3

Visual Sequence interpretation is easier than Safety Warning Recognition and Tools Identification

Interpretation of visual sequence in technical illustrations is easier for VLMs as compared to recognising safety warnings or identifying tool. However, this might be due to the number of samples in VSI task (50) and the nature of the task (MCQ), further study needs to be done to verify this.

7.4 Finding 4

Humans perform better overall, however VLMs are better in Visual Sequence Interpretation.

Evaluating humans on the TICQA dataset, reveals humans perform exceptionally well (95 % accuracy) in recognising various safety warning and are also able to identify almost all (98.5 % accuracy) safety warnings. However, they perform mediocre (37 % accuracy) on visual sequence interpretation. Also all VLMs surpass human baseline in VSI task, while performing average on other two tasks. This is especially surprising, since VLMs performed better on average without context on this task. This likely reflects the complexity of the VSI task: removing textual cues makes it challenging for humans but helps VLM.

This chapter presented 4 findings:

- Providing context to VLMs has mixed effects
- New architecture and training strategies improve model performance
- Visual Sequence interpretation is easier than Safety Warning Recognition and Tools Identification
- Humans perform better overall, however VLMs are better in Visual Sequence Interpretation.

8. Conclusion

8.1 Summary

In this thesis, I set out to bridge a critical gap in multimodal evaluation by focusing on the comprehension of technical illustrations found in product manuals. I began by motivating the importance of these domain-specific graphics—sequence diagrams that guide assembly and disassembly, annotated diagrams that identify tools and components, and standardised symbols that convey safety and hazard information. Recognising that existing benchmarks largely overlook such schematics, I defined "technical illustration comprehension" as a distinct capability encompassing spatial reasoning, symbol recognition, and visual cues understanding.

To study this concept, I designed **TICQA**, a novel dataset structured around three complementary tasks:

1. **Visual Sequence Interpretation (VSI)**, where models select the correct sequence from multiple-choice options;
2. **Tools & Components Identification (TCI)**, which requires naming a tool or part from a redacted image;
3. **Safety Warning Recognition (SWR)**, demanding open-ended descriptions of hazard symbols and precautions.

A rigorous redaction protocol removed all textual labels, forcing models to rely purely on visual understanding. For evaluation, I combined exact-match scoring on VSI with a **Panel of LLM Judges (PoLL)** —a collection of medium-sized language models—and tested the dataset with university students to establish human baseline.

The empirical study involved nine state-of-the-art Vision Language Models, each under 20 billion parameters. The results were illuminating: VLMs not only matched but in many cases surpassed human performance on the assembly MCQs, yet they fell noticeably short on open-ended safety descriptions and component naming, resulting in overall average performance compared to humans. This observation supports **H1** to some extent. The inclusion of contextual information improved on average 4.5 % accuracy in VLMs, validating **H2**. However, special care must be taken per task while adding context, since including context does not yield uniform gains across tasks.

Together, these findings demonstrate both the promise and the current limits of VLMs in processing technical illustrations, while establishing TICQA as a good starting point for future dataset development in this domain.

8.2 Answers to Research Questions

RQ1: How accurately can state-of-the-art VLMs infer the correct assembly/disassembly sequence in assembly diagrams when textual cues are redacted?

Answer: The study shows that VLMs **perform pretty well** (with an avg. accuracy of 60.0 %) even when textual clues are not present in the illustrations. This points to a presence of strong sequence reasoning abilities in current medium sized (<20B param) VLMs.

RQ2: To what extent can these models identify tools and components purely from visual shape cues, without relying on overlaid labels?

Answer: The study indicates that current VLMs **find it hard** (on avg. 36.5 % accuracy) to identify tools and components present in technical graphics without the help of text.

RQ3: How effectively do VLMs recognise and describe safety warnings and hazard icons in open-ended questions?

Answer: The study points that VLMs under 20B parameters are **not good** (40.6 % avg. accuracy) at recognising safety warning illustrations.

8.3 Limitations

While TICQA represents a significant step forward, it is important to acknowledge its constraints. First, the dataset's **scope**—42 manuals yielding 250 annotated samples—captures only a fraction of the vast variability in industry graphics and product categories. As a result, model performance may not generalize to highly specialised domains (e.g., aerospace schematics or biomedical device diagrams).

Secondly, the **redaction protocol**, though necessary to isolate visual reasoning, may overcorrect by removing contextual cues that humans often rely on—such as brief text labels or numeric callouts—thus creating an artificial evaluation environment. In practice, humans interpret combined text-image content, so models fine-tuned exclusively on redacted diagrams might underperform in integrated scenarios.

Thirdly, the **generation of dataset** using semi-automatic techniques, while efficient than manual strategies, still needs improvement: the chosen method allows for collection of cropped images from product manuals, however the complexity of the task demands manual intervention at later stages, annotation and verification.

Fourthly, the inclusion of only **English manuals**, retracts the study to English language only. Many studies have found that their is a huge bias towards English language due to larger availability of text corpora, future studies on this task with focus on multilingual or languages other than English can strengthen multimodal multilingual model development.

Finally, **hardware constraints** limited my experiments to VLMs under 20 billion parameters. Leading larger open-source models—many exceeding 100 billion parameters —remain untested, leaving open the question of how scale and optimisation techniques might alter performance on technical graphics. Recognising these limitations helps frame TICQA's results and highlights areas for improvements.

8.4 Outlook

Looking forward, several avenues can extend and enrich this work. **Expanding TICQA** to include more samples, manuals, a wider variety of products, and multilingual content would improve its representativeness and challenge models with diverse symbols and standards. Also defining and including more tasks like VSI, TCI and SWR would promote robust evaluation on comprehension of technical illustrations.

On the evaluation side, developing **hybrid metrics** that blend panel-LLM judgments with lightweight human-in-the-loop validation may strike a better balance between scalability and reliability. Evaluating the **reasoning** behind answer to identify hallucinations and misinterpretations could provide deeper analysis of model behaviour. Especially, measuring **model uncertainty** and a**ttention attribution** could provide more insights on each modality's contribution to the final answer and reasoning.

From a model-development perspective, fine-tuning VLMs on combined text-and-image instructional corpora—rather than exclusively redacted visuals—may yield systems that more closely mirror real-world user experiences. Integrating VLMs into **interactive assistance tools**, where models provide step-by-step guidance and real-time safety feedback, could revolutionise technical support, reducing user errors and improving accessibility.

Ultimately, advancing VLM comprehension of technical illustrations holds promise not only for consumer product manuals but also for industrial automation, virtual maintenance environments, and educational platforms. By charting this course, TICQA lays the groundwork for AI systems that truly “see” and interpret the visual instructions that undermine modern technology.

Bibliography

Amara, K., Klein, L., Lüth, C., Jäger, P., Strobelt, H., & El-Assady, M. (2024). Why context matters in VQA and Reasoning: Semantic interventions for VLM input modalities. arXiv preprint arXiv:2410.01690.

Agrawal, H., Desai, K., Wang, Y., Chen, X., Jain, R., Johnson, M., Batra, D., Parikh, D., Lee, S., & Anderson, P. (2019). nocaps: novel object captioning at scale (pp. 8948–8957). https://nocaps.org

Bahl, L. R., Brown, P. F., de Souza, P. V., & Mercer, R. L. (1986). MAXIMUM MUTUAL INFORMATION ESTIMATION OF HIDDEN MARKOV MODEL PARAMETERS FOR SPEECH RECOGNITION. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 49–52. https://doi.org/10.1109/ICASSP.1986.1169179

Banerjee, S., & Lavie, A. (n.d.). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., Do, Q. V., Xu, Y., & Fung, P. (2023). A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. 675–718. https://doi.org/10.18653/v1/2023.ijcnlp-main.45

Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C., Ca, J. U., Kandola, J., Hofmann, T., Poggio, T., & Shawe-Taylor, J. (2003). A Neural Probabilistic Language Model. Journal of Machine Learning Research, 3(Feb), 1137–1155.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning Long-Term Dependencies with Gradient Descent is Difficult. IEEE Transactions on Neural Networks, 5(2), 157–166. https://doi.org/10.1109/72.279181

Bostrom, K., & Durrett, G. (2020). Byte Pair Encoding is Suboptimal for Language Model Pretraining. Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020, 4617–4624. https://doi.org/10.18653/v1/2020.findings-emnlp.414

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 2020-December. https://arxiv.org/pdf/2005.14165

Browne, M. W. (2000). Cross-Validation Methods. Journal of Mathematical Psychology, 44(1), 108–132. https://doi.org/10.1006/JMPS.1999.1279

Castelli, V., Chakravarti, R., Dana, S., Ferritto, A., Florian, R., Franz, M., Garg, D., Khandelwal, D., McCarley, S., McCawley, M., Nasr, M., Pan, L., Pendus, C., Pitrelli, J., Pujar, S., Roukos, S., Sakrajda, A., Sil, A., Uceda-Sosa, R., … Zhang, R. (2019). The TechQA Dataset. Proceedings of the Annual Meeting of the Association for Computational Linguistics, 1269–1278. https://doi.org/10.18653/v1/2020.acl-main.117

Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., & Zhao, F. (2024). Are We on the Right Way for Evaluating Large Vision-Language Models? https://arxiv.org/pdf/2403.20330

Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Cui, E., Zhu, J., Ye, S., Tian, H., Liu, Z., Gu, L., Wang, X., Li, Q., Ren, Y., Chen, Z., Luo, J., Wang, J., Jiang, T., Wang, B., … Wang, W. (2024). Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling. https://arxiv.org/pdf/2412.05271

Chizhov, P., Nee, M., Langlais, P.-C., & Yamshchikov, I. P. (2025). What the HellaSwag? On the Validity of Common-Sense Reasoning Benchmarks. https://arxiv.org/pdf/2504.07825

Chung, K. L. (1960). Markov Chains with Stationary Transition Probabilities. Markov Chains with Stationary Transition Probabilities. https://doi.org/10.1007/978-3-642-49686-8

Clark, A., Fox, C., & Lappin, S. (2010). The Handbook of Computational Linguistics and Natural Language Processing. The Handbook of Computational Linguistics and Natural Language Processing. https://doi.org/10.1002/9781444324044;JOURNAL:JOURNAL:BOOKS;WGROUP:STRING:PUBLICATION

Collobert, R., Weston, J., Com, J., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research, 12, 2493–2537.

Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North, 4171–4186. https://doi.org/10.18653/V1/N19-1423

Ding, Y., Ren, K., Huang, J., Luo, S., & Han, S. C. (2024). PDF-MVQA: A Dataset for Multimodal Information Retrieval in PDF-based Visual Question Answering. https://arxiv.org/pdf/2404.12720

dos Santos, G. O., Colombini, E. L., & Avila, S. (2021). CIDEr-R: Robust Consensus-based Image Description Evaluation. W-NUT 2021 - 7th Workshop on Noisy User-Generated Text, Proceedings of the Conference, 351–360. https://doi.org/10.18653/v1/2021.wnut-1.39

Dou, Z. Y., Xu, Y., Gan, Z., Wang, J., Wang, S., Wang, L., Zhu, C., Zhang, P., Yuan, L., Peng, N., Liu, Z., & Zeng, M. (2021). An Empirical Study of Training End-to-End Vision-and-Language Transformers. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2022-June, 18145–18155. https://doi.org/10.1109/CVPR52688.2022.01763

Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., ichter, brian, Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., … Florence, P. (n.d.). PaLM-E: An Embodied Multimodal Language Model.

Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. Journal of Machine Learning Research, 23(120), 1–39. http://jmlr.org/papers/v23/21-0998.html

Foundations of Statistical Natural Language Processing - Christopher Manning, Hinrich Schutze - Google Books. (1998). MIT Press. https://books.google.de/books?hl=en&lr=&id=YiFDxbEX3SUC&oi=fnd&pg=PR16&dq=foundation+of+statistical+natural+language+processing&ots=v0ullCfJOO&sig=XdesAdDrj31uqsZVDIYCt-1rnzo&redir_esc=y#v=onepage&q=foundation%20of%20statistical%20natural%20language%20processing&f=false

Gale, W. A., & Sampson, G. (1995). Good-Turing Frequency Estimation Without Tears. Journal of Quantitative Linguistics, 2(3), 217–237. https://doi.org/10.1080/09296179508590051;WGROUP:STRING:PUBLICATION

Gao, D., Wang, R., Shan, S., & Chen, X. (2023). CRIC: A VQA Dataset for Compositional Reasoning on Vision and Commonsense. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(5), 5561–5578. https://doi.org/10.1109/TPAMI.2022.3210780

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D., & Tech, V. (2017). Making the v in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering (pp. 6904–6913). http://visualqa.org/

Guan, T., Liu, F., Wu, X., Xian, R., Li, Z., Liu, X., Wang, X., Chen, L., Huang, F., Yacoob, Y., Manocha, D., & Zhou, T. (2024). HallusionBench: An Advanced Diagnostic Suite for Entangled Language Hallucination and Visual Illusion in Large Vision-Language Models (pp. 14375–14385). https://github.com/tianyi-lab/HallusionBench.

Hello GPT-4o | OpenAI. (n.d.). Retrieved 30 April 2025, from https://openai.com/index/hello-gpt-4o/

Hiippala, T., Alikhani, M., Haverinen, J., Kalliokoski, T., Logacheva, E., Orekhova, S., Tuomainen, A., Stone, M., & Bateman, J. A. (2021). AI2D-RST: a multimodal corpus of 1000 primary school science diagrams. Language Resources and Evaluation, 55(3), 661–688. https://doi.org/10.1007/S10579-020-09517-1/FIGURES/11

Hinton, G. E., & Salakhutdinov, R. R. (2012). A Better Way to Pretrain Deep Boltzmann Machines. Advances in Neural Information Processing Systems, 25.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/NECO.1997.9.8.1735

Huang, J., & Zhang, J. (n.d.). A Survey on Evaluation of Multimodal Large Language Models.

Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., Lv, T., Cui, L., Mohammed, O. K., Patra, B., Liu, Q., Aggarwal, K., Chi, Z., Bjorck, N., Chaudhary, V., Som, S., SONG, X., & Wei, F. (2023). Language Is Not All You Need: Aligning Perception with Language Models. Advances in Neural Information Processing Systems, 36, 72096–72109.

Hudson, D. A., & Manning, C. D. (2019). GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering (pp. 6700–6709).

Hussein, A. (2024). Technical illustrations. https://www.kaggle.com/datasets/ahmedhgabr/technical-illustration

Introducing ChatGPT | OpenAI. (n.d.). Retrieved 30 April 2025, from https://openai.com/index/chatgpt/

Jelinek, F. (1976). Continuous Speech Recognition by Statistical Methods. Proceedings of the IEEE, 64(4), 532–556. https://doi.org/10.1109/PROC.1976.10159

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q. V, Sung, Y., Li, Z., & Duerig, T. (2021). Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision (pp. 4904–4916). PMLR. https://proceedings.mlr.press/v139/jia21b.html

Kang, L., Tito, R., Valveny, E., & Karatzas, D. (2024). Multi-page Document Visual Question Answering Using Self-attention Scoring Mechanism. 219–232. https://doi.org/10.1007/978-3-031-70552-6_13

Kaplan, J., McCandlish, S., Henighan OpenAI, T., Brown OpenAI, T. B., Chess OpenAI, B., Child OpenAI, R., Gray OpenAI, S., Radford OpenAI, A., Wu OpenAI, J., & Amodei OpenAI, D. (2020). Scaling Laws for Neural Language Models. https://arxiv.org/pdf/2001.08361

Katz, S. M. (1987). Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. IEEE Transactions on Acoustics, Speech, and Signal Processing, 35(3), 400–401. https://doi.org/10.1109/TASSP.1987.1165125

Kraaij, W., Hain, T., Lincoln, M., Mccowan, I., Carletta, J., Kraaij, W., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., Post, W., Reidsma, D., & Wellner, P. (2005). The AMI meeting corpus. International Workshop on Machine Learning for Multimodal Interaction, Springer, 28–29. www.amiproject.org,

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M. W., Dai, A. M., Uszkoreit, J., Le, Q., & Petrov, S. (2019). Natural Questions: A Benchmark for Question Answering Research. Transactions of the Association for Computational Linguistics, 7, 453–466. https://doi.org/10.1162/TACL_A_00276/43518/NATURAL-QUESTIONS-A-BENCHMARK-FOR-QUESTION

Laskar, M. T. R., Bari, M. S., Rahman, M., Bhuiyan, M. A. H., Joty, S., & Huang, J. X. (2023). A Systematic Study and Comprehensive Evaluation of ChatGPT on Benchmark Datasets. Proceedings of the Annual Meeting of the Association for Computational Linguistics, 431–469. https://doi.org/10.18653/v1/2023.findings-acl.29

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Proceedings of the Annual Meeting of the Association for Computational Linguistics, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

Li, B., Ge, Y., Chen, † Yi, Ge, Y., Zhang, R., & Shan, Y. (2024). SEED-Bench-2-Plus: Benchmarking Multimodal Large Language Models with Text-Rich Visual Comprehension. https://arxiv.org/pdf/2404.16790

Li, B., Rui, 1⋆, 1⋆, W., Wang, G., Ge, Y., Ge, Y., Shan, Y., & Lab, T. A. (2023). SEED-Bench: Benchmarking Multimodal LLMs with Generative Comprehension. https://arxiv.org/pdf/2307.16125

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., & Hoi, S. C. H. (2021). Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. Advances in Neural Information Processing Systems, 34, 9694–9705. https://github.com/salesforce/ALBEF.

Li, L., Lei, J., Gan, Z., & Liu, J. (2021). Adversarial VQA: A New Benchmark for Evaluating the Robustness of VQA Models (pp. 2042–2051).

Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W. X., & Wen, J. R. (2023). Evaluating Object Hallucination in Large Vision-Language Models. EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings, 292–305. https://doi.org/10.18653/v1/2023.emnlp-main.20

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C. D., Ré, C., Acosta-Navas, D., Hudson, D. A., … Koreeda, Y. (2022). Holistic Evaluation of Language Models. Annals of the New York Academy of Sciences, 1525(1), 140–146. https://doi.org/10.1111/nyas.15007

Lien, J. J., Cohn, J. F., Kanade, T., & Li, C. C. (1998). Automated facial expression recognition based on FACS action units. Proceedings - 3rd IEEE International Conference on Automatic Face and Gesture Recognition, FG 1998, 390–395. https://doi.org/10.1109/AFGR.1998.670980

Lin, C.-Y. (n.d.). ROUGE: A Package for Automatic Evaluation of Summaries.

Lin, H., Cheng, X., Wu, X., & Shen, D. (2022). CAT: Cross Attention in Vision Transformer. Proceedings - IEEE International Conference on Multimedia and Expo, 2022-July. https://doi.org/10.1109/ICME52920.2022.9859720

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8693 LNCS(PART 5), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023a). Visual Instruction Tuning. Advances in Neural Information Processing Systems, 36, 34892–34916. https://llava-vl.github.io

Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023b). Visual Instruction Tuning. Advances in Neural Information Processing Systems, 36, 34892–34916. https://llava-vl.github.io

Liu, X., & Croft, W. B. (2005). Statistical Language Modeling For Information Retrieval. Annu. Rev. Inf. Sci. Technol.

Liu, Y., Li, Z., Huang, M., Yang, B., Yu, W., Li, C., Yin, X.-C., Liu, C.-L., Jin, L., & Bai, X. (2024). OCRBench: on the hidden mystery of OCR in large multimodal models. Iss, 67(13). https://doi.org/10.1007/s11432-024-4235-6

Lu, S., Li, Y., Chen, Q.-G., Xu, Z., Luo, W., Zhang, K., & Ye, H.-J. (2024). Ovis: Structural Embedding Alignment for Multimodal Large Language Model. https://arxiv.org/pdf/2405.20797

Marino, K., Rastegari, M., Farhadi, A., & Mottaghi, R. (2019). OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge (pp. 3195–3204).

Mathew, M., Bagal, V., Tito, R., Karatzas, D., Valveny, E., & Jawahar, C. V. (2022). InfographicVQA (pp. 1697–1706).

Mathew, M., Karatzas, D., & Jawahar, C. V. (2021). DocVQA: A Dataset for VQA on Document Images (pp. 2200–2209). https://www.industrydocuments.ucsf.edu/

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings. https://arxiv.org/pdf/1301.3781

Mikolov, T., Karafiát, M., Burget, L., Jan, C., & Khudanpur, S. (2010). Recurrent neural network based language model. Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010, 1045–1048. https://doi.org/10.21437/INTERSPEECH.2010-343

Nandy, A., Sharma, S., Maddhashiya, S., Sachdeva, K., Goyal, P., & Ganguly, N. (2021). Question Answering over Electronic Devices: A New Benchmark Dataset and a Multi-Task Learning based QA Framework. Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021, 4600–4609. https://doi.org/10.18653/v1/2021.findings-emnlp.392

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal Deep Learning.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., & Kiela, D. (2019). Adversarial NLI: A New Benchmark for Natural Language Understanding. Proceedings of the Annual Meeting of the Association for Computational Linguistics, 4885–4901. https://doi.org/10.18653/v1/2020.acl-main.441

Open VLM Leaderboard - a Hugging Face Space by opencompass. (n.d.). Retrieved 30 April 2025, from https://huggingface.co/spaces/opencompass/open_vlm_leaderboard

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (n.d.). BLEU: a Method for Automatic Evaluation of Machine Translation.

Peña, A., Morales, A., Fierrez, J., Serna, I., Ortega-Garcia, J., Puente, Í., Córdova, J., & Córdova, G. (2023). Leveraging Large Language Models for Topic Classification in the Domain of Public Affairs. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 14193 LNCS, 20–33. https://doi.org/10.1007/978-3-031-41498-5_2

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1, 2227–2237. https://doi.org/10.18653/v1/n18-1202

Pham, N., & Schott, M. (2024). H-POPE: Hierarchical Polling-based Probing Evaluation of Hallucinations in Large Vision-Language Models. https://arxiv.org/pdf/2411.04077

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision (pp. 8748–8763). PMLR. https://proceedings.mlr.press/v139/radford21a.html

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (n.d.). Language Models are Unsupervised Multitask Learners. Retrieved 30 April 2025, from https://github.com/codelucas/newspaper

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical Text-Conditional Image Generation with CLIP Latents. https://arxiv.org/pdf/2204.06125

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-Shot Text-to-Image Generation (pp. 8821–8831). PMLR. https://proceedings.mlr.press/v139/ramesh21a.html

Rao, J., Shan, Z., Liu, L., Zhou, Y., & Yang, Y. (2023). Retrieval-based Knowledge Augmented Vision Language Pre-training. MM 2023 - Proceedings of the 31st ACM International Conference on Multimedia, 5399–5409. https://doi.org/10.1145/3581783.3613848

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Le Scao, T., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N. V., … Rush, A. M. (2021). Multitask Prompted Training Enables Zero-Shot Task Generalization. ICLR 2022 - 10th International Conference on Learning Representations. https://arxiv.org/pdf/2110.08207

Satoh, S., & Kanade, T. (1997). Name-it: Association of face and name in video. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 368–373. https://doi.org/10.1109/CVPR.1997.609351

Schwenk, D., Khandelwal, A., Clark, C., Marino, K., & Mottaghi, R. (2022). A-OKVQA: A Benchmark for Visual Question Answering Using World Knowledge. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 13668 LNCS, 146–162. https://doi.org/10.1007/978-3-031-20074-8_9

Shanahan, M. (2024a). Talking about Large Language Models. Communications of the ACM, 67(2), 68–79. https://doi.org/10.1145/3624724;PAGE:STRING:ARTICLE/CHAPTER

Shanahan, M. (2024b). Talking about Large Language Models. Communications of the ACM, 67(2), 68–79. https://doi.org/10.1145/3624724;PAGE:STRING:ARTICLE/CHAPTER

Shen, C., Cheng, L., Nguyen, X. P., You, Y., & Bing, L. (2023). Large Language Models are Not Yet Human-Level Evaluators for Abstractive Summarization. Findings of the Association for Computational Linguistics: EMNLP 2023, 4215–4233. https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.278

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., Lewkowycz, A., Agarwal, A., Power, A., Ray, A., Warstadt, A., Kocurek, A. W., Safaya, A., Tazarv, A., … Wu, Z. (2022). Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. https://arxiv.org/pdf/2206.04615

Team, K., Du, A., Yin, B., Xing, B., Qu, B., Wang, B., Chen, C., Zhang, C., Du, C., Wei, C., Wang, C., Zhang, D., Du, D., Wang, D., Yuan, E., Lu, E., Li, F., Sung, F., Wei, G., … Lin, Z. (2025). Kimi-VL Technical Report. https://arxiv.org/pdf/2504.07491

Thede, S. M., & Harper, M. P. (n.d.). A Second-Order Hidden Markov Model for Part-of-Speech Tagging.

Tur, G., Stolcke, A., Voss, L., Peters, S., Hakkani-Tur, D., Dowding, J., Favre, B., Fernandez, R., Frampton, M., Frandsen, M., Frederickson, C., Graciarena, M., Kintzing, D., Leveque, K., Mason, S., Niekrasz, J., Purver, M., Riedhammer, K., Shriberg, E., … Yang, F. (2010). The CALO meeting assistant system. IEEE Transactions on Audio, Speech and Language Processing, 18(6), 1601–1611. https://doi.org/10.1109/TASL.2009.2038810

Turing Machines on JSTOR. (n.d.). Retrieved 30 April 2025, from https://www.jstor.org/stable/24969370

Turk, M. (2014). Multimodal interaction: A review. Pattern Recognition Letters, 36(1), 189–195. https://doi.org/10.1016/J.PATREC.2013.07.003

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. Advances in Neural Information Processing Systems, 30.

Verga, P., Hofstätter, S., Althammer, S., Su, Y., Piktus, A., Arkhangorodsky, A., Xu, M., White, N., & Cohere, P. L. (2024). Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. https://arxiv.org/pdf/2404.18796

Vinciarelli, A., Pantic, M., Bourlard, H., & Pentland, A. (2008). Social signal processing: State-of-the-art and future perspectives of an emerging domain. MM'08 - Proceedings of the 2008 ACM International Conference on Multimedia, with Co-Located Symposium and Workshops, 1061–1070. https://doi.org/10.1145/1459359.1459573

Wang, B., Chen, W., Pei, H., Xie, C., Kang, M., Zhang, C., Xu, C., Xiong, Z., Dutta, R., Schaeffer, R., Truong, S. T., Arora, S., Mazeika, M., Hendrycks, D., Lin, Z., Cheng, Y., Koyejo, S., Song, D., & Li, B. (2023). DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models. Advances in Neural Information Processing Systems, 36. https://arxiv.org/pdf/2306.11698

Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., Cao, Y., Liu, Q., Liu, T., & Sui, Z. (2023). Large Language Models are not Fair Evaluators. https://arxiv.org/pdf/2305.17926

Wang, T., Roberts, A., Hesslow, D., Scao, T. Le, Chung, H. W., Beltagy, I., Launay, J., & Raffel, C. (2022). What Language Model Architecture and Pretraining Objective Works Best for Zero-Shot Generalization? (pp. 22964–22984). PMLR. https://proceedings.mlr.press/v162/wang22u.html

Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., & Cao, Y. (2021). SimVLM: Simple Visual Language Model Pretraining with Weak Supervision. ICLR 2022 - 10th International Conference on Learning Representations. https://arxiv.org/pdf/2108.10904

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., & Fedus, W. (2022). Emergent Abilities of Large Language Models. https://arxiv.org/pdf/2206.07682

Weizenbaum, J. (1983). ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine. Communications of the ACM, 26(1), 23–28. https://doi.org/10.1145/357980.357991;PAGE:STRING:ARTICLE/CHAPTER

xai-org/RealworldQA · Datasets at Hugging Face. (n.d.). Retrieved 30 April 2025, from https://huggingface.co/datasets/xai-org/RealworldQA

Xia, R., Mao, S., Yan, X., Zhou, H., Zhang, B., Peng, H., Pi, J., Fu, D., Wu, W., Ye, H., Feng, S., Wang, B., Xu, C., He, C., Cai, P., Dou, M., Shi, B., Zhou, S., Wang, Y., … Qiao, Y. (2024). DocGenome: An Open Large-scale Scientific Document Benchmark for Training and Testing Multi-modal Large Language Models. Arxiv. https://arxiv.org/pdf/2406.11633

Xu, C., Hou, X., Liu, J., Li, C., Huang, T., Zhu, X., Niu, M., Sun, L., Tang, P., Xu, T., Cheng, K.-T., & Guo, M. (n.d.). MMBench: Benchmarking End-to-End Multi-modal DNNs and Understanding Their Hardware-Software Implications.

Yang, T., Wang, Y., Lu, Y., & Zheng, N. (2022). Visual Concepts Tokenization. Advances in Neural Information Processing Systems, 35, 31571–31582. https://github.com/thomasmry/VCT

Ying, K., Meng, F., Wang, J., Li, Z., Lin, H., Yang, Y., Zhang, H., Zhang, W., Lin, Y., Liu, S., Lei, J., Lu, Q., Chen, R., Xu, P., Zhang, R., Zhang, H., Gao, P., Wang, Y., Qiao, Y., … Shao, W. (2024). MMT-Bench: A Comprehensive Multimodal Benchmark for Evaluating Large Vision-Language Models Towards Multitask AGI. Proceedings of Machine Learning Research, 235, 57116–57198. https://arxiv.org/pdf/2404.16006

You, Q., Jin, H., Wang, Z., Fang, C., & Luo, J. (2016). Image Captioning With Semantic Attention (pp. 4651–4659).

Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., & Wang, L. (2023). MM-Vet: Evaluating Large Multimodal Models for Integrated Capabilities. Proceedings of Machine Learning Research, 235, 57730–57754. https://arxiv.org/pdf/2308.02490

Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., … Chen, W. (2023). MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI. https://doi.org/10.1109/CVPR52733.2024.00913

Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., … Chen, W. (2024). MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 9556–9567). https://mmmu-benchmark.github.io/

Zhang, L., Hu, A., Zhang, J., Hu, S., & Jin, Q. (2023). MPMQA: Multimodal Question Answering on Product Manuals. Proceedings of the AAAI Conference on Artificial Intelligence, 37(11), 13958–13966. https://doi.org/10.1609/AAAI.V37I11.26634

Zhang, T., Wang, S., Li, L., Zhang, G., Taslakian, P., Rajeswar, S., Fu, J., Liu, B., & Bengio, Y. (2024). VCR: A Task for Pixel-Level Complex Reasoning in Vision Language Models via Restoring Occluded Text. https://arxiv.org/pdf/2406.06462

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., … Wen, J.-R. (n.d.). A Survey of Large Language Models. Retrieved 27 March 2025, from https://www.bing.com/new

Zhou, W., Zeng, Y., Diao, S., & Zhang, X. (2022). VLUE: A Multi-Task Multi-Dimension Benchmark for Evaluating Vision-Language Pre-training (pp. 27395–27411). PMLR. https://proceedings.mlr.press/v162/zhou22n.html

Zhu, F., Liu, Z., Ng, X. Y., Wu, H., Wang, W., Feng, F., Wang, C., Luan, H., & Chua, T. S. (2024). MMDocBench: Benchmarking Large Vision-Language Models for Fine-Grained Visual Document Understanding. https://arxiv.org/pdf/2410.21311

Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., Gao, Z., Cui, E., Wang, X., Cao, Y., Liu, Y., Wei, X., Zhang, H., Wang, H., Xu, W., … Wang, W. (2025). InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. https://arxiv.org/pdf/2504.10479

Annex

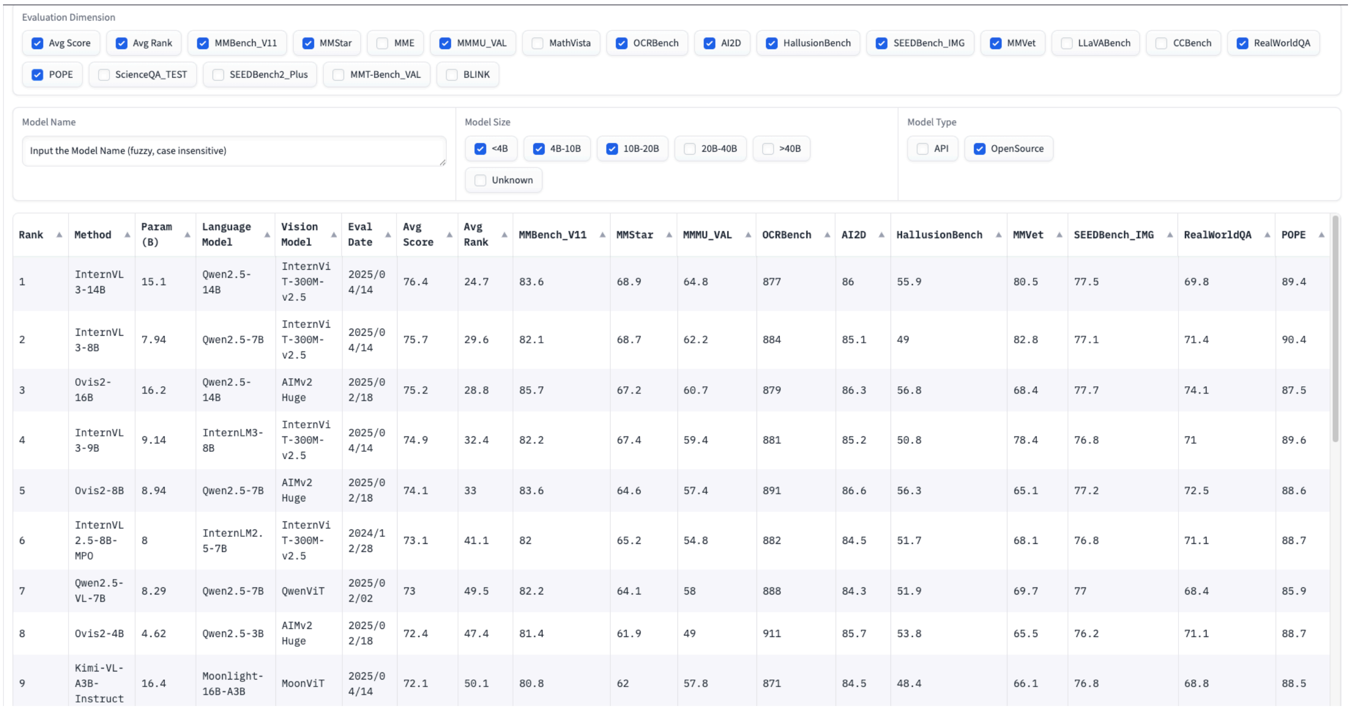
A1: Product Manuals details

Product Name	Company	Manual	Subset
Air conditioner	LG	air_conditioner0, air_conditioner2	TCI
Air Fryer	Philips	air_fryer	SWR
Baby stroller	Graco	baby_stroller	TCI, VSI
Blender	Kenwood	Blender	VSI
Blower	Husqvarna	blower	SWR
Boat	Yamaha	boat	SWR, VSI
Cabinet	Ikea	Ikea1	TCI, VSI
Coffee Machine	Siemens, Citiz	coffee_machine1, coffee_machine0	TCI, VSI
Digital Camera	Canon	Camera0	TCI, VSI
Dishwasher	Ikea, Beko	Dishwasher1, ikea_dishwasher0, Dishwasher0	TCI, VSI
Electric Toothbrush	Philips	Toothbrush0	SWR
Exercise bike	Schwinn	exercise_bikes, exercise_bike1	TCI, VSI
Fax machine	Brother	Fax	SWR
Gaming Chair	GTPlayer	Chair0	TCI, VSI
Generator	Yamaha	Generator	SWR, VSI
Glass shower door	Anzzi	glass_shower_door	TCI
Grill	Kenmore	grill	SWR
Jetski	Yamaha	Jetski	SWR, VSI
Keyboard	NZXT	Keyboard	TCI
Kitchen air ventilator	Miele, Ikea	Miele0, ikea_ventilator	TCI
Microwave	GE	Microwave0	TCI, VSI
Monitor	LG	Monitor	SWR
Motherboard	Asus	Motherboard1	VSI
Oven	Ikea	Ikea_oven0	TCI

Product Name	Company	Manual	Subset
Projector	BenQ	Projector0	SWR
Refrigerator	Electrolux	Fridge0, fridge1	TCI, VSI
Robot Vacuum Cleaner		robotVaccum0	VSI
Security Camera	Cisco	security_camera1	SWR, VSI
Shower Handel	Ikea	ikea_shower0, ikea_shower1	TCI
Sink	Ikea	ikea_sink0	TCI
Sitting Lawn Mover	Toro	lawn_mover0	SWR, VSI
Snowmobile	Yamaha	Snowmobile	SWR
Sofa	Ikea	sleeper_sofa0	TCI
Television	LG	Television1	SWR
Washing Machine	Electrolux	washing_machine1	SWR
Washing Machine	Samsung	washing_machine4, washing_machine5	TCI
Water Pump	Yamaha	Pump	SWR
Work Bench	Ikea	Ikea0	SWR, VSI

Table: Details of product manuals used in the dataset

A2: OpenVLM leaderboard score



Evaluation Dimension

☑ Avg Score ☑ Avg Rank ☑ MMBench_V11 ☑ MMStar ☐ MME ☑ MMMU_VAL ☐ MathVista ☑ OCRBench ☑ AI2D ☑ HallusionBench ☑ SEEDBench_IMG ☑ MMVet ☐ LLaVABench ☐ CCBench ☑ RealWorldQA ☑ POPE ☐ ScienceQA_TEST ☐ SEEDBench2_Plus ☐ MMT-Bench_VAL ☐ BLINK

Model Name

Input the Model Name (fuzzy, case insensitive)

Model Size

☑ <4B ☑ 4B-10B ☑ 10B-20B ☐ 20B-40B ☐ >40B ☐ Unknown

Model Type

☐ API ☑ OpenSource

Rank	Method	Param (B)	Language Model	Vision Model	Eval Date	Avg Score	Avg Rank	MMBench_V11	MMStar	MMMU_VAL	OCRBench	AI2D	HallusionBench	MMVet	SEEDBench_IMG	RealWorldQA	POPE
1	InternVL3-14B	15.1	Qwen2.5-14B	InternViT-300M-v2.5	2025/04/14	76.4	24.7	83.6	68.9	64.8	877	86	55.9	80.5	77.5	69.8	89.4
2	InternVL3-8B	7.94	Qwen2.5-7B	InternViT-300M-v2.5	2025/04/14	75.7	29.6	82.1	68.7	62.2	884	85.1	49	82.8	77.1	71.4	90.4
3	Ovis2-16B	16.2	Qwen2.5-14B	AIMv2 Huge	2025/02/18	75.2	28.8	85.7	67.2	60.7	879	86.3	56.8	68.4	77.7	74.1	87.5
4	InternVL3-9B	9.14	InternLM3-8B	InternViT-300M-v2.5	2025/04/14	74.9	32.4	82.2	67.4	59.4	881	85.2	50.8	78.4	76.8	71	89.6
5	Ovis2-8B	8.94	Qwen2.5-7B	AIMv2 Huge	2025/02/18	74.1	33	83.6	64.6	57.4	891	86.6	56.3	65.1	77.2	72.5	88.6
6	InternVL2.5-8B-MPO	8	InternLM2.5-7B	InternViT-300M-v2.5	2024/12/28	73.1	41.1	82	65.2	54.8	882	84.5	51.7	68.1	76.8	71.1	88.7
7	Qwen2.5-VL-7B	8.29	Qwen2.5-7B	QwenViT	2025/02/02	73	49.5	82.2	64.1	58	888	84.3	51.9	69.7	77	68.4	85.9
8	Ovis2-4B	4.62	Qwen2.5-3B	AIMv2 Huge	2025/02/18	72.4	47.4	81.4	61.9	49	911	85.7	53.8	65.5	76.2	71.1	88.7
9	Kimi-VL-A3B-Instruct	16.4	Moonlight-16B-A3B	MoonViT	2025/04/14	72.1	50.1	80.8	62	57.8	871	84.5	48.4	66.1	76.8	68.8	88.5

Fig: OpenVLM Leaderboard as of 20th April 2025

statement

I hereby declare that I have produced this work independently and without using any aids other than those specified. The thoughts taken directly or indirectly from external sources are marked as such. This is especially true for software-generated texts. To the best of my knowledge, the work has not yet been submitted in the same or a similar form to any other examining authority and has not yet been published.

Hof, **01.05.2025**