
CS 159 Project Report

Stavya Arora

Sani Deshmukh

Isha Goswami

Nora Xiao

1 Introduction

Spot the Difference is a popular game played worldwide that challenges players' keen perceptual skills by asking them to identify differences between two seemingly identical images. In fact, an 11-year old girl from Germany can find the different dot in a circle of hundreds of colored dots immediately after viewing a specific section of the image. When GPT-4o is prompted with finding the difference between the two images shown in Figure 1, it actually does not perform well and only names some differences correctly, misses others, and mentions items that are not different between the images.



Figure 1: The differences are: the hands on the clock, the pin in the lady's hair, the shape of the handles on the cabinet, the item on the shelf, the item the cat is holding, the socks on the girl's feet, the cookie in the bottom right corner, and the number of pink objects on the far right.

Clearly, there is room for improvement here for the analysis of details in images. The game could merely serve as a benchmark for visual understanding and attention in multimodal AI models, testing object permanence, scene continuity, and visual memory. Apart from the simple game that tests children's perceptual abilities, the fine-grained visual attention to detail behind spot the difference has implications for medical imaging, surveillance and security, satellite imagery, and art authentication, among other domains. This requires LLMs not to just detect visual elements, but also to compare spatial relationships in a semantic context. Moreover, this benchmark could help improve the robustness of AI in critical applications such as autonomous driving, where quickly spotting subtle but significant visual changes is crucial.

The task is challenging for current models and agents because it requires object comparison from two different images, rather than just object detection. The general-purpose reasoning models struggle with the pixel-level precision required for finding differences across nearly identical images. Throughout our exploration, we find that models fail to find the 'needle in the haystack' or *what* the actual minute differences are between objects that are detected as different. Along the same lines, conversion of image regions into tokens can be inconsistent and yield inconsistent outputs. When an LLM is simply prompted to find the differences for the same image pair, it might identify certain differences in one trial, but in the next it identifies another set of differences. This stems from the probabilistic nature of LLM decoding and the lack of determinism in visual token alignment. The inconsistency has especially important consequences for reliability in high-stakes environments where predictable and reproducible performance is imperative.

Persistent visual memory and the current attention mechanisms for images are inherently not robust in LLMs or even most vision models. Importance is given to regions of images or patterns that

are significant based on learned patterns. Highly detailed images might have differences in regions that are not usually 'important.' While humans can scan images and keep track of positions and appearances of objects, LLMs are not good enough at 'remembering' details from one image and comparing them against the second.

We aim to address these technical challenges by developing a specialized agent for image difference detection by integrating precise prompting techniques for GPT-4o with the modular visual reasoning framework of Visual Programming (VisProg) [7]. Inspired by the sub-hypothesis model for agentic falsification in Huang *et al.* [6], we aim to structure the image comparison as a sequence of sub-hypothesis validations. Our research will investigate whether decomposing visual difference detection into targeted sub-hypotheses with GPT-4o leads to better performance on challenging comparison tasks that can be carried out with VisProg programs. By combining structured hypothesis testing with visual analysis we seek to advance AI systems towards more human-like capabilities in detailed visual comparison tasks.

2 Background and previous work

Visual reasoning has emerged as a critical challenge in AI, particularly in tasks like visual question answering (VQA) and image comparison. Early neural-network approaches directly mapped images and textual queries to answers [1], which yielded impressive pattern recognition but often struggled with complex, compositional questions requiring stepwise reasoning [2]. Structured reasoning approaches, where models explicitly decompose tasks into sub-steps, have significantly improved performance and interpretability. For example, Johnson *et al.* (2017) devised a model that generates an explicit symbolic program for each step and then executes it with specialized neural modules, achieving better generalization on the CLVR visual reasoning benchmark [2].

Another line of research targets the comparison of two images directly. A representative task is learning to "spot the difference" between image pairs, as in surveillance or medical change detection. Jhamtani and Berg-Kirkpatrick (2018) introduced a dataset of nearly identical image pairs (e.g. sequential surveillance frames) and trained models to generate textual descriptions of their differences [3]. Their approach aligned latent object-level changes with words, outperforming naive attention-based captioners. However, such models were still fundamentally learned mappings from image differences to text, lacking an explicit reasoning or hypothesis-testing mechanism.

Recent advances in transformer-based architecture have been applied to visual change detection tasks. The Visual change Transformer (VcT) [8] addresses change detection by mining unchanged background information, but still operates on the pixel-level. While it highlights the importance of leveraging changed and unchanged regions, we hope to implement a difference detection with semantic-level understanding of objects and their differences.

Building on the idea of explicit reasoning, several recent methods use structured decomposition or symbolic planning to tackle visual content. These approaches often employ large language models (LLMs) as "planners" that break down a visual query into sub-tasks, which are then executed by vision modules. For instance, the VisProg system by Subramanian *et al.* (2023) uses an LLM to generate a small program in response to a question [4]. Another example is ViperGPT [5], which integrates code generation into visual reasoning. ViperGPT treats a complex query as a programming problem, prompting an LLM to produce Python code that calls appropriate vision functions to obtain the answer. Beyond program generation, researchers have also looked at hypothesis-driven reasoning agents. One such approach, POPPER, proposes an AI agent that embraces Karl Popper's philosophy of falsification for hypothesis testing [6]. Such agentic reasoning systems highlight the importance of decomposition and verification.

Recent iterative refinement frameworks have shown promise in visual reasoning: MMC [9] uses Monte-Carlo Tree Search to systematically explore reasoning trajectories for general VQA tasks, while Duan *et al.* [10] demonstrate a solely LLM-driven approach that iteratively refines specific, spatial grounding for UI design. We synthesize these approaches for spot-the-difference tasks, combining LLM-based hypothesis generation with structured visual search to identify and verify differences between images.

Despite significant progress, there remains a gap in fine-grained image comparison capabilities. Existing vision-language models are biased toward holistic understanding and language fluency rather

than detailed differential analysis. In medical imaging, an algorithm comparing a patient’s scan before and after treatment must not only detect minute changes but also explain their nature (e.g. "the lesion has reduced in size"). Current models offer either low-level change detection with no semantic explanation, or high-level explanations that overlook critical details. This motivates the proposed research: extending multimodal systems with structured reasoning and hypothesis decomposition capabilities to identify and explain differences between images.

3 Research Question and Approach

Our goal is to investigate how to build a pipeline that accurately and comprehensively detects visual differences between image pairs. This task, while intuitive for humans, remains challenging for general-purpose multimodal models. Specifically, we aim to determine how prompting strategies and structured reasoning influence performance in fine-grained visual comparison. To explore this, we utilize GPT-4o and VisProg to evaluate three approaches that vary in the degree of structure and symbolic reasoning they incorporate:

1. **Plain LLM Prompting:** We test the model’s performance when asked to spot differences directly, using no prior guidance or scaffolding. This serves as a baseline to understand the default capabilities of GPT-4o in unconstrained image comparison.
2. **Hypothesis-style LLM Prompting:** We prompt GPT-4o to generate discrete, semantically grounded visual questions. These questions isolate specific aspects of the image (such as object presence, quantity, or color) to guide more systematic comparison.
3. **LLM + VisProg:** We pair GPT-4o with VisProg, prompting the model to convert visual questions into symbolic DSL code. This allows us to decompose the comparison into executable steps that provide interpretable, grounded answers. We specifically selected VisProg for its explicit reasoning traceability and GPT-4o for its advanced language modeling capabilities, hypothesizing that their combination could leverage structured symbolic reasoning alongside rich semantic understanding.

By analyzing how each of these strategies handles complexity, visual grounding, and consistency, we aim to better understand the limitations of end-to-end prompting and the potential of structured symbolic reasoning in multimodal tasks.

3.1 What is VisProg?

VisProg is a modular visual programming framework designed to execute symbolic programs that perform visual reasoning over images. It enables structured question answering using a set of interpretable operations such as object detection, existence checking, attribute comparison, and more. Programs in VisProg are written in a domain-specific language (DSL) and can operate on raw images through a sequence of grounded steps.

The key advantage of VisProg is its transparency and compositional reasoning capability. Rather than relying on opaque neural predictions, it provides a structured execution trace, making it especially suitable for tasks like visual difference detection.

4 Experiments and Results

To address our research question, we construct a three-phase experimental pipeline, grounded in the "spot the difference" task. Our goal is to assess the performance and reliability of different prompting strategies and toolchains for automated image comparison.

Our pipeline involves:

- GPT-4o as the base language model.
- VisProg as the visual reasoning engine.
- Three comparison strategies: (1) plain prompting, (2) hypothesis-style prompting, and (3) symbolic execution via VisProg.

4.1 Phase 1: Plain GPT-4o Prompting

In the first baseline, we ask GPT-4o directly to "spot the differences" between two input images.



Figure 2: Input images used in the examples throughout the report. The differences are the 4 small boats in the water in the right image that are not in the left image.

The model implicitly generates sub-questions and returns outputs in natural language. This approach simulates a basic user query without structured decomposition. It operates under a *zero-shot* prompting paradigm, where GPT-4o receives no examples or intermediate guidance—only the direct instruction to identify visual differences. While this method is fast and requires minimal prompt engineering, it yields highly variable responses. The same image pair may produce different sets of identified differences across runs, highlighting the model’s reliance on internal heuristics rather than visual consistency. Furthermore, the lack of structure means some responses include hallucinated differences, vague comparisons, or redundancies. This motivates our move toward more guided prompting strategies in subsequent phases.

Spot the Differences

1. Sky and Clouds:

- Image 1 shows more detailed clouds and a layered sky.
- Image 2 has a flat, cyan-tinted sky with no cloud patterns.

2. Color Tone and Lighting:

- Image 1 is cooler and more naturally lit.
- Image 2 has higher contrast, more magenta/peach saturation in the land and water.

3. Foreground Boat:

- The wake pattern behind the boat differs slightly between the two images.
- The boat also appears marginally brighter in Image 2.

4. Water Shade:

- The water is more turquoise/bright in Image 2.
- In Image 1, it's deeper and more navy-colored.

5. Opera House Shadows:

- Shadows on the sails and roof of the Opera House vary in sharpness and intensity.
- More sunlight reflection in Image 2.

6. Building Details in the Background:

- Certain buildings (e.g., the long structure on the harbor's far side) are differently lit or tinted between the images.
- More structure contrast in Image 2.

7. Vegetation and Trees:

- The trees and green areas are greener and more shadowed in Image 1.
- In Image 2, they are warmer and more yellow-toned.

Preprocessing: Difference Heatmap

While exploring GPT-4o’s performance in Phase 1, we found that the model often hallucinated differences, omitted actual changes, or produced inconsistent outputs across repeated runs on the same image pair. This variability stemmed from the lack of grounding in specific visual regions and the model’s reliance on internal heuristics. To address this limitation, we introduced a difference heatmap as a preprocessing step to guide downstream prompting. The heatmap highlights regions of significant pixel-level change between the two images, combining color and structural differences. We aligned and normalized the input images, computed a weighted blend of RGB and grayscale differences, and applied a threshold with gamma correction to filter out noise. The final binary map

localizes areas of visual change and is used in Phase 2 and Phase 3 to anchor question generation and restrict GPT-4o’s focus to semantically meaningful regions. This preprocessing step was critical for improving the precision and relevance of generated questions in the structured prompting pipeline.

4.2 Phase 2a: Hypothesis-Style GPT-4o Prompting

In this approach, we guide GPT-4o to generate hypothesis-style, structured questions rather than open-ended answers. This encourages visual grounding and logical coverage of the image contents.

Example questions:

Is there a ferry in both images?
 Are the number of sailboats equal?
 Is the water the same color?
 Does each image contain a red building?

These questions improve relevance and structure. The answers are generated by GPT-4o using vision models. While hallucinations are reduced, the number of validated differences remains similar.

Question	LEFT	RIGHT	Different?
How many boats are visible?	2	7	Yes
Ferry color?	Yellow	White/Blue	Yes
People on ferry deck?	Undeterminable	Unknown	Yes
Opera house shell shape?	Spherical sections	Sails	Yes
Sail orientation?	Northeast	Outward	Yes
Red building visible?	Not clearly	No	Yes
Domes on opera house?	Multiple sails	0	Yes
Boat shape?	Rectangular	Various sizes	Yes
Total Differences Found	8		

Plain GPT-4o Prompting Results - Differences Identified

Here, we also extended this to *few-shot* prompting, where we prepended the instruction with 1–2 example image pairs (described textually) and their associated question lists. These input–output pairs served to implicitly teach the model the correct structure, level of granularity, and visual focus expected in its outputs. The specific prompts are described in detail in our Appendix section. Example questions:

Is a small boat present near the water’s edge?
 How many sails are visible on boats?
 Is any boat moving in the water?
 Are there flags near the building?

This few-shot format improved question consistency, better aligned outputs with the heatmap regions, and reduced vague or redundant phrasing.

Question	LEFT	RIGHT	Different?
How many boats are visible?	2	5	Yes
What is the color of the boat?	White, blue	White	Yes
How many sails visible?	One sail	Two	Yes
Orientation of nearest visible boat?	Counterclockwise	Diagonal left	Yes
Water clear of traffic?	Yes	No	Yes
Visible ladders on dock	Unknown	No	Yes
Total Differences Found	6		

Few Shot Prompting Results - Differences Identified

Although the hypothesis-driven questions improved visual grounding, the accuracy still depended heavily on the quality of object detection by GPT-4o’s internal visual system, which often misidentified or overlooked subtle differences.

4.3 Phase 2b: Chain of Thought Prompting: Sub-Hypothesis Expansion

To simulate scientific reasoning and falsification, we introduced a new layered prompting strategy inspired by *chain-of-thought* (CoT) prompting. Rather than relying on GPT-4o to answer visual difference questions directly, we prompted it to think step-by-step by expanding each high-level question into multiple sub-questions targeting specific visual attributes. This mirrors the agentic reasoning process described in POPPER [6], where broad hypotheses are recursively decomposed into more granular tests. The sub-questions focus on quantifiable aspects like object presence, color, shape, or quantity. By making each reasoning step explicit, this form of CoT prompting allows us to externalize the model’s latent reasoning process into interpretable units, which can then be verified independently. While this approach increased the number of testable differences and provided better semantic coverage, it also introduced challenges: subtle variations in wording or focus often led to inconsistent or overlapping answers, highlighting the tradeoff between reasoning granularity and output stability. The detailed sub-questions significantly reduced the ambiguity in identifying differences but increased sensitivity to minor semantic misinterpretations, which inflated the count of detected differences erroneously.

Main question:

How many boats are visible?

Generated sub-questions:

Are there any sailboats in the image?

Is there a dock with boats attached to it?

Are there any small boats near the shoreline?

These expanded questions were the ones tested via GPT-4o’s VQA interface.

Question	LEFT	RIGHT	Different?
Is there a dock with boats attached to it?	Yes	No	Yes
How many boats are within 500 meters of Opera House?	One	Five	Yes
Water is clear or has any visible hue, such as green or brown?	Blue	Clear	Yes
How many people are in proximity to the Opera House?	Many	Unknown	Yes
Are there structures or objects blocking the view of the sky?	No	Yes	Yes
How many sail-like structures are visible on the Opera House?	Seven	Six	Yes
How many distinct white sail structures?	Six	Five	Yes
Is the boat’s hull flat or curved at the bottom?	Flat	Curved	Yes
Total Differences Found	8		

Chain of Thought Prompting Results - Differences Identified

4.4 Phase 3: GPT-4o + VisProg Symbolic Execution

Here, we fully utilize VisProg. GPT-4o is responsible for:

- Generating atomic, specific questions about the image pair.
- Translating those questions into VisProg DSL code.
- Interpreting the execution results.

Each question is compiled into executable VisProg code by GPT-4o in a separate prompt using symbolic modules such as FIND, COUNT, VQA, and EXISTS.

4.4.1 Direct Prompting for GPT-4o VisProg Generation

First, we tried a method similar to that in phase 2a, where GPT-4o was prompted to generate questions focused on the regions of the image that are different based on the heatmap of pixel differences. The questions generated by GPT-4o were similar to those in phase 2a and an example of the corresponding VisProg code is shown.

Question: How many boats are visible in the water?

```

VisProg program (for left image):
boats = FIND(image=LEFT, object="boats")
num_boats = COUNT(region=boats)
result = RESULT(var=num_boats)
FIND
COUNT
RESULT

```

The program generated for the right image is the exact same except the image argument would be 'image=RIGHT' instead.

Question	LEFT	RIGHT	Different?
How many boats visible in the water?	8	13	Yes
Ferry color?	blue	white	Yes
Shape of sails on building?	curved	round	Yes
How many green areas visible?	3	4	Yes
Total Differences Found	4		

Single-level GPT-4o Prompting for VisProg generation results—only differences shown

4.4.2 Chain of Thought Prompting for VisProg Generation

While keeping the same procedure of prompting GPT-4o for VisProg DSL code, we added follow-up prompting for a few specific questions for each high-level question generated to mimic the sub-hypothesis structure for falsification tests, as described in phase 2b. In this case, GPT-4o was told in the high-level question generation prompt that the questions would be used to generate more specific ones (see the Appendix). Examples of the parent and child questions are shown.

Main question 1: What is the shape of the prominent building's roof?

Sub-questions 1: Are there curved sails on the roof?, Is the roof multi-layered?, Does the roof have any pointed elements?, Are there white elements on top of the roof?

Main question 2: How many boats are in the water?

Sub-questions 2: Is there a large ferry in the bottom center?, Is there a sailboat near the center?, Are there multiple small boats visible?

Since a VisProg program is generated for each sub-question, the resulting process has 10-15 different questions and corresponding VisProg programs.

Question	LEFT	RIGHT	Different?
Large ferry in bottom center?	True	True	No
Sailboat near center?	True	True	No
Multiple small boats visible?	10	17	Yes

Results for sub-question 'How many boats are in the water?'

Across all the sub-questions, 4 differences were identified between the two images with this method.

The modular nature of VisProg inherently provides transparency and interpretability, suggesting substantial room for performance enhancement through refining symbolic modules and their interaction with GPT-4o.

5 Discussion and conclusion

Our research systematically evaluated three distinct methods for visual difference detection: (1) plain prompting, (2) hypothesis-style prompting, and (3) structured symbolic reasoning using VisProg. Each approach demonstrated unique strengths and shortcomings.

5.1 Plain Prompting

Plain GPT-4o prompting exhibited ease of implementation and rapid output but suffered from inconsistency, frequent hallucinations, and a lack of reliability. Despite occasionally high recall, the inability to consistently verify the results significantly limits this method’s practical applicability, particularly in sensitive, high-stakes scenarios.

5.2 Hypothesis-style Prompting

Hypothesis-style prompting improves precision and structure. Our initial approach with the chain of thought prompting method (Phase 2b) using sub-hypothesis expansion achieved high coverage, producing up to 3× more testable questions. However, this came at the cost of increased false positives: many sub-questions introduced subtle wording variations or ambiguous concepts, leading GPT-4o to produce inconsistent answers. For instance, two semantically similar questions like “Is the object red?” and “What color is the object?” might receive mismatched answers due to the model’s nondeterminism or subtle visual differences. In one trial, 28 differences were identified — a gross overestimate — highlighting the need for better filtering, normalization, and grounding of sub-hypotheses in heatmap-relevant regions. Despite this, the sub-hypothesis technique provides a valuable tool for decomposing comparisons in a more structured and extensible way.

5.3 Chain of Thought (CoT) Prompting

Introducing the chain-of-thought prompting method significantly expanded the number of verifiable sub-questions, allowing detailed exploration of image differences. Although powerful, this method increased complexity, occasionally producing overly granular or semantically ambiguous questions leading to conflicting responses. Notably, this method highlighted GPT-4o’s tendency towards semantic drift, reinforcing the necessity for clear, contextually grounded sub-question generation.

5.4 VisProg Integration

The VisProg-based symbolic pipeline offers the most reliable and interpretable outcomes for this particular image, though its accuracy depends highly on the coverage and specificity of the DSL modules. Improving the DSL pipeline relies heavily on fine tuning the API prompts and to first get the right questions, and then to generate accurate answers for these questions. The experiments and analysis of the pipeline are converted more in the following sections.

5.5 Implications and Future Directions

Our findings demonstrate the potential and limitations of structured visual reasoning for multi-modal AI. The superior interpretability and reliability offered by the symbolic reasoning pipeline strongly advocate for integrating structured prompting with visual reasoning frameworks in practical applications.

Future research directions include:

- Enhancing VisProg DSL capabilities to handle more sophisticated visual reasoning.
- Developing automated prompt-engineering frameworks to consistently generate context-specific sub-questions.
- Exploring hybrid models that integrate probabilistic visual attention mechanisms with structured symbolic reasoning to achieve both flexibility and reliability.
- Evaluating performance on domain-specific datasets (e.g., medical imagery or surveillance) to validate practical robustness.

In conclusion, the structured integration of symbolic visual reasoning with GPT-4o significantly enhances multimodal model performance in fine-grained visual tasks, setting a foundational benchmark for future advancements in visually detailed AI reasoning.

Appendix

Link to code: <https://github.com/Sani-Deshmukh/CS-159>

Phase 1 - Zero-shot Prompt:

Please spot the differences between the following two input images.

Phase 2a - Structured Prompt:

```
"You are given two images of the same scene and a difference heatmap. "  
"The difference heatmap is a binary image where the difference  
between the two images is highlighted in white. "  
"Generate AS MANY *specific and visually testable* questions that help  
  
identify differences between them as you can. "  
"Use the difference heatmap to generate questions that focus on the differences  
between the two images. "  
"Cover the following categories:\n"  
"- Presence or absence of objects (e.g. 'Is there a boat?')\n"  
"- Quantity (e.g. 'How many cars are there?')\n"  
"- Color (e.g. 'What is the color of the shirt?')\n"  
"- Shape or orientation (e.g. 'What shape is the object? Is it upright or tilted?')\n"  
"- Action or behavior (e.g. 'Is anyone walking? Is the dog sitting?')\n"  
"Write each question so it can be answered **independently** for each image - do not  
compare directly (e.g. not 'Which image has more trees?').\n\n"  
"Do not combine questions. Do not repeat the same wording.\n\n"  
"Each question should focus on a specific object, color, action, or spatial detail. "  
"Questions should have one word answers. "  
"Respond only with a JSON array of strings (no explanation, no Markdown)."  
)
```

Phase 2a - Few-Shot Prompt: This was added on to the earlier structured prompt from Phase 2a.

"Here are a few examples of good question generation based on image pairs and their difference heatmaps. Each example includes a short description of the two images, what changed (based on the heatmap), and a list of concrete visual questions that could be answered independently for each image."

"EXAMPLE 1:"

"Image 1: A city street with three yellow taxis, a red stop sign, and a pedestrian crossing the street."

"Image 2: The same street, but one taxi is missing, the pedestrian is gone, and the stop sign has turned green."

"Difference heatmap: Highlights the areas where the pedestrian, the missing taxi, and the stop sign are located."

" Example of a few generated questions:"

"["

"How many taxis are visible?"

"What color is the stop sign?"

"Is there a pedestrian on the street?"

"Is the stop sign present?"

"Are there people crossing the street?"

"]"

"EXAMPLE 2:"

"Image 1: A dining table with a blue plate, a fork, a glass of water, and a single lit candle."
 "Image 2: The same table, but the candle is missing and the plate is now red."
 "Difference heatmap: Highlights the center of the table and the candle's position."
 " Example of a few generated questions:"
 "["
 "What is the color of the plate?,"
 "Is the candle lit?,"
 "Is a candle present on the table?,"
 "How many glasses are on the table?,"
 "What items are placed near the center of the table?"
 "]"
 "Now, given two new images of the same scene and a difference heatmap. "...

Phase 2b - Chain of Thought Prompt (Zero Shot): This was added on to the earlier prompt from Phase 2a.

"Given the high-level visual question: \"{question}\"
 "Generate 3 more specific subquestions that test concrete, visual hypotheses. "
 "These should be answerable from an image using object recognition, counting, or attribute detection"
 "Return a JSON array of short questions only. ...

Phase 2b - Chain of Thought Prompt (Few Shot): This was the full prompt for generating sub-questions with examples.

"You are given two images of the same scene and a difference heatmap. "
 "The difference heatmap is a binary image where white pixels indicate regions of change. "
 "You are also given these high-level questions about differences between the images:
 {json.dumps(parent_q, indent=2)}"
 "Your task is to generate as many **specific**, independent, concrete, and visually answerable sub-questions for each high-level question as possible that help identify each minute difference between the images.
 "PURPOSE: The questions will be automatically mapped to symbolic reasoning modules (like VQA, FIND, COUNT, EXISTS). "
 "Your job is to phrase questions that can be answered using those operations, focused ONLY on changed regions.
 " Guidelines:
 "- Focus on **visual attributes** (color, count, presence, shape, size, material, visibility, texture).
 "- Write **one question per object** or concept - split compound questions.
 "- Avoid vague or abstract questions (like 'Is it pretty?').
 "- Do NOT refer to both images at once (no comparisons like 'Is it different?').
 "- DO NOT write symbolic code or refer to modules like FIND or VQA.
 "- DO NOT reference the heatmap or the fact that there are two images.
 "- Make sure the question has a **factual, short answer** (e.g., yes/no, number, or single word).
 "Refinement strategies: "
 " - Add spatial specificity (general → specific location) "

```

" - Add attribute specificity (object → object + color/size/state) "
"- Add quantitative precision (things → exact counts)"
"- Break compound questions into atomic parts"

" Examples:"
" Level 1: Are there any vehicles visible?"
" Level 2: [Is there a car in the image?, "
           "Is there a truck in the image?",
           "Are there any motorcycles visible?]"

" Level 1: What is in the upper portion?"
"Level 2: [What objects are in the upper left quadrant?,"
          "What is in the upper right corner? ,"
          "What is at the very top of the image?]"

"Level 2: Is there a car in the image?"
"Level 3: [Is there a red car visible?,"
          "Is there a car in the foreground?,"
          "Is there a parked car?]"
"Return a JSON object mapping each parent question to its refined versions:"
"{"
  "parent_question_1": ["refined_1", "refined_2", "refined_3"],
  "parent_question_2": ["refined_1", "refined_2"]
"}"

```

Phase 3 - VisProg Structured Prompt: The phase 2a prompt was used to generate questions. The following prompt then converted each of these questions to a VisProg program.

```

"
Generate a VisProg program to answer the question. Use ONLY these functions:
- VQA(image=..., question=...)
- FIND(image=..., object=...)
- COUNT(region=...)
- EXISTS(region=...)
- RESULT(var=...)

IMPORTANT:
- If the question can be answered using VQA, use VQA(image=..., question=...) first.
- Each question should have only one object as subject; Break queries with multiple subject
  down to multiple questions.
- Make sure each object has its own question.
- Always question the more significant object first.
- Format each line as: output_var = FUNCTION(arg1=value1, arg2=value2)
- The final line must always be: result = RESULT(var=...)
- Do not use markdown.

Question: {question}
Image: {image_side}
"

```

References

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [2] Johnson, J. et al. Inferring and Executing Programs for Visual Reasoning. In *CVPR*, 2017. d
- [3] Jhamtani, H. and Berg-Kirkpatrick, T. Learning to Describe Differences Between Pairs of Similar Images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
- [4] Subramanian et al. Modular Visual Question Answering via Code Generation. In *ACL*, 2023.
- [5] Surís D, Menon S, Vondrick C. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.
- [6] Huang K, Jin Y, Li R, Li MY, Candès E, Leskovec J. Automated Hypothesis Validation with Agentic Sequential Falsifications. arXiv preprint arXiv:2502.09858. 2025.
- [7] Gupta T, Kembhavi A. Visual Programming: Compositional visual reasoning without training. arXiv preprint arXiv:2211.11559. 2022.
- [8] Jiang B, Wang Z, Wang X, Zhang Z, Chen L, Wang X, Luo B. VcT: Visual change Transformer for remote sensing image change detection. arXiv preprint arXiv:2310.11417, 2023.
- [9] Liu S, Zhang Z, Hu P, Ma J, Du J, Wang Q, Zhang J, Liu Q, Gao J, Ma F. MMC: Iterative refinement of VLM reasoning via MCTS-based multimodal critique. arXiv preprint arXiv:2504.11009, 2025.
- [10] Duan P, Cheng C-Y, Hartmann B, Li Y. Visual Prompting with Iterative Refinement for Design Critique Generation. arXiv preprint arXiv:2412.16829. 2025.