

Do MLLMs Exhibit Human-like Perceptual Behaviors? HVS-Bench: A Benchmark for MLLM Alignment with Human Perceptual Behavior

Jiaying Lin^{*1,2} Shuquan Ye^{*1,3,4} Dan Xu² Wanli Ouyang^{3,4} Rynson W.H. Lau¹
¹City University of Hong Kong ²HKUST ³CUHK ⁴Shenzhen Loop Area Institute

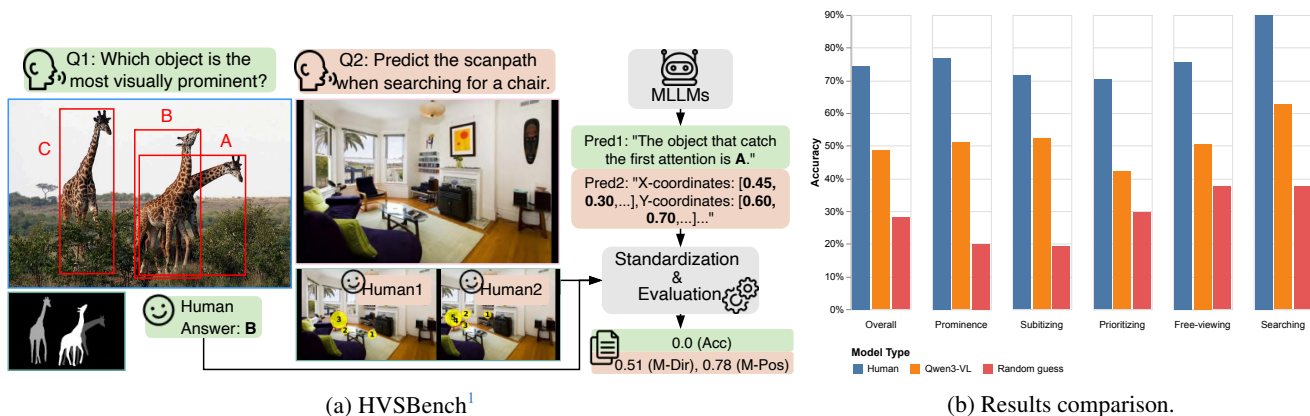


Figure 1. We are the first to systematically study and assess MLLMs-HVS alignment. (a) We propose large-scale and comprehensive HVS-Bench, with a robust evaluation protocol. (b) Our comparisons among humans and the state-of-the-art model Qwen3-VL on HVS-Bench across 5 fields reveal room for improvement and insights for developing HVS-aligned MLLMs.

Do MLLMs Exhibit Human-like Perceptual Behaviors? HVSBench: A Benchmark for MLLM Alignment with Human Perceptual Behavior

Supplementary Material

The supplementary materials include:

- Section 1: Preliminary Settings for HVBench**

This section explains the rationale behind the selection of fields for HVBench. It also provides details about the corresponding source datasets used for each selected field.
- Section 2: Sample Prediction and Standardization**

We present the predictions generated by the models and the corresponding outputs after applying our standardization process for each field. This ensures consistency and comparability across different fields.
- Section 3: Detailed Settings**

This section provides detailed settings in our experiments and standardization pipeline, including the inputs used in our experiments, the detailed experimental settings, and the pseudo-code for our automated standardization pipeline.
- Section 4: Additional Qualitative Examples and Results**

We include extra qualitative examples showcasing sample questions and the predictions made by MLLMs under each field.
- Section 5: Field-Specific Hints in Each Field**

For each field, we provide specific hints or context that help understand tasks.
- Section 6: Detailed Benchmark Results and Discussions**

This section contains a detailed report of benchmark results for all 13 question types, offering a comprehensive view of the performance across various fields and question types. We also provide additional ablation studies and related discussions for them.
- Section 7: More Related Work**

We include more related work for different research areas to showcase the necessity of our HVSBench.
- Section 8: Application:**

We include an application, which shows that content generation models better aligned with the HVS can produce more reasonable outputs.

1. Preliminary setting for HVBench

To evaluate the extent to which multimodal large language models (MLLMs) align with the HVS, we construct multimodal queries based on five fields designed for different datasets. These fields are critical for assessing whether MLLMs perceive and interpret visual information in a manner akin to humans. For a fair and comprehensive evalua-

tion, the five fields are carefully curated to capture different aspects of the HVS. These fields include:

Prominence. The prominence in HVS enables humans to identify the most visually prominent objects within an image [15], making it a critical application for understanding human visual focus in the HVS. In this field, we choose the SIFR dataset [8] to construct our benchmark data. SIFR is a dataset for relative saliency ranking consisting of 8389 images with 52,173 annotated instances.

Subitizing. Subitizing [28] is to quickly and accurately perceive the number of visually prominent objects in a scene. Compared to Prominence, it requires simultaneous attention to multiple elements. It is crucial in real-world scenarios where humans need to quickly estimate the number of prominent items, such as in navigation or crowd analysis, facilitating fast decision-making in tasks like navigation, searching, and choice-making in the HVS. We choose SIFR dataset [8] and SIS10K [15] for this field since the original subitizing dataset [28] is no longer available. SIS10K is a large-scale salient instance segmentation dataset. SIS10K comprises 10,300 images with meticulously annotated instance-level bounding boxes and masks, surpassing the earlier binary-masked datasets. Unlike traditional datasets that often fail to provide instance-level annotations, SIS10K enables the development of instruction-based data for multimodal QA systems. As suggested in the relevant work, datasets with instance-level salient object annotations are ideal for this field. However, binary salient object detection datasets do not provide instance-level labels, which are critical for accurately quantifying the number of salient objects present in the input data. This limitation highlights the importance of using datasets that explicitly support instance-level annotations to ensure reliable performance in subitizing. Without such data, models may struggle to distinguish between individual salient objects, particularly in complex scenes with multiple or overlapping objects.

Prioritizing. Prioritizing in HVS enables humans to rank objects within a scene based on their perceptual saliency [8]. It better captures the dynamic nature of HVS, i.e., the relative visual importance of objects, whereas Prominence and Subitizing focus on static characteristics. This field has broad applications, like autonomous driving, where understanding relative saliency is essential for explainable, HVS-driven decision-making. In this field, we choose the SIFR dataset [8] to construct our benchmark data. Unlike other ranking datasets, the salient instances in

SIFR were determined based on clustering and thresholding on real-world human fixation, ensuring a better alignment with the saliency rank and the real attention model in the HVS.

Free-Viewing. Free-viewing is an important behavior of HVS. Free-viewing (bottom-up) gaze path prediction [25] focuses on modeling and forecasting human gaze behavior in a task-free context, driven solely by the intrinsic saliency of visual stimuli. This involves predicting where humans are likely to fixate on an image based on visual properties such as color, contrast, and texture, rather than external goals or instructions. In our HVSBench, we utilize the COCO-FreeView dataset[5] to construct the assessment data for this field. COCO-FreeView [5] is a dataset containing 6202 images with about 300,000 fixations viewed by human subjects under a free-viewing condition without specific search goals. Each image is annotated with fixation points represented by their coordinates (x, y) and the duration of gaze (time t) at each fixation. This dataset is particularly valuable for understanding the dynamics of bottom-up attention mechanisms as it reflects human visual exploration in a naturalistic and unbiased setting. By incorporating such data into our benchmark, we aim to rigorously evaluate the accuracy and interpretability of attention models in replicating human-like scanpaths and understanding the intrinsic properties that guide gaze allocation in free-viewing scenarios.

Searching. Searching [24] focuses on human gaze behavior in task-driven contexts, such as object search, where attention is top-down and influenced by contextual information, like object context and semantic relationships. Unlike free-viewing, searching enhances human efficiency and flexibility [24]. Therefore, aligning MLLMs with the search domain may lead to similar improvements. In this field, we employ COCO-Search18 dataset [24] as our primary dataset. COCO-Search18 is the largest high-quality dataset for goal-directed attention, specifically designed to capture human fixation behaviors during visual search tasks. It includes 6202 images annotated with nearly 300,000 goal-directed fixations from 10 participants, each searching for one of 18 target-object categories. We use the standard target-present split. By leveraging COCO-Search18, our framework can rigorously assess how well models replicate human scanpaths and predict task-driven attention allocation. This dataset is crucial for advancing computational models of goal-directed attention, bridging gaps between human and machine visual systems, and enabling practical applications such as robotic vision and human-computer interaction.

2. Sample Prediction and Standardization

Table 1 shows the standardized predictions for Q9, Q12 in the Free-viewing and Searching fields.

Table 2 shows the standardized predictions for Q2 in the Subitizing field.

Table 3 shows the standardized predictions for Q1, Q4, Q5, Q6, Q7, Q8, Q10, Q11, Q13 in the Prominence, Prioritizing, Free-viewing, and Searching fields.

Table 4 shows the standardized predictions for Q3 in the Prioritizing field.

3. Detailed Settings

Detailed inputs and experimental settings. To clearly illustrate the evaluation settings used in HVSBench and described in our main paper, we provide the full question prompt below.

For the top sample in Fig. 3 in the main paper, the full question is:

“[Requirement] Predict eye fixation points for searching for a sink in the provided image. Output the fixation points as three separate lists, each containing 6 values. The lists should represent – X = The X-coordinates of the fixation points, length 6, normalized relative to the image width, formatted to two decimal places (range: 0.00 to 1.00). Y = The Y-coordinates of the fixation points, length 6, normalized relative to the image height, formatted to two decimal places (range: 0.00 to 1.00). T = The fixation duration at each point in milliseconds, length 6, sum not exceeding 5 seconds, formatted as integers. [Example Output for Another Image] X = [0.50, 0.48, 0.44, 0.66, 0.67, 0.50] Y = [0.49, 0.45, 0.37, 0.26, 0.20, 0.50] T = [73, 193, 95, 635, 592, 312]”

For the middle sample in Fig. 3 in the main paper, the full question is:

“[Requirement] Predict eye fixation points for searching for a tv in the provided image. Output the fixation points as three separate lists, each containing 6 values. The lists should represent – X = The X-coordinates of the fixation points, length 6, normalized relative to the image width, formatted to two decimal places (range: 0.00 to 1.00). Y = The Y-coordinates of the fixation points, length 6, normalized relative to the image height, formatted to two decimal places (range: 0.00 to 1.00). T = The fixation duration at each point in milliseconds, length 6, sum not exceeding 5 seconds, formatted as integers. [Example Output for Another Image] X = [0.50, 0.48, 0.44, 0.66, 0.67, 0.50] Y = [0.49, 0.45, 0.37, 0.26, 0.20, 0.50] T = [73, 193, 95, 635, 592, 312]”

The full prediction is:

Sample Prediction	Standardization
“X = [0.49, 0.57, 0.56, ...] Y = [0.53, 0.53, 0.51, ...] T = [316, 148, 123, ...]”	X=[0.49, 0.57, 0.56], Y=[0.53, 0.53, 0.51], T=[316, 148, 123]
“X = [0.49, 0.57, 0.56, 0.75, 0.85, 0.95]”	X=[0.5], Y=[0.5], T=[0]
“the result scanpath is X = [0.49, 0.57, 0.56], Y = [0.53, 0.53, 0.51], and T = [316, 148, 123]”	X=[0.49, 0.57, 0.56], Y=[0.53, 0.53, 0.51], T=[316, 148, 123]
“Here’s a prediction of eye fixation points for the provided image in the format requested:\n X-coordinates(normalized):\n 0.32,0.54,0.43\n Y-coordinates (normalized):\n 0.22,0.31,0.54\n duration (ms):\n 384,287,166\n This prediction provides three lists representing the locations (X and Y) and durations (T) of eye fixations for a free-viewing scenario on the image.”	X=[0.32, 0.54, 0.43], Y=[0.22, 0.31, 0.54], T=[384, 287, 166]
“Here’s a prediction of eye fixation points for the provided image in the format requested:\n X-Coordinates (normalized):\n 0.32,0.54,0.43\n Y-Coordinates (normalized):\n 0.22,0.31,0.54\n Fixation Durations:\n 384,287,166\n This prediction provides three lists representing the locations (X and Y) and durations (T) of eye fixations for a free-viewing scenario on the image.”	X=[0.32, 0.54, 0.43], Y=[0.22, 0.31, 0.54], T=[384, 287, 166]
“json\n{“X”: [0.45, 0.52, 0.60],\n“Y”: [0.50, 0.54, 0.52],\n“T”: [312, 165, 130]}\n”	X=[0.45, 0.52, 0.6], Y=[0.5, 0.54, 0.52], T=[312, 165, 130]
“**X = ** [0.45, 0.52, 0.60],\n **Y = ** [0.50, 0.54, 0.52],\n **T = ** [312, 165, 130]”	X=[0.45, 0.52, 0.6], Y=[0.5, 0.54, 0.52], T=[312, 165, 130]

Table 1. Standardized predictions for Q9, Q12 in Free-viewing and Searching fields.

Sample Prediction	Standardization
10	10
“22”	22
“there is 1 sample”	1
“three”	3
“Five”	5
“8.”	8
“num of people in this image is 5”	5
“B”	GT_avg
“There are five salient objects in the image: four boats and one flower.”	5

Table 2. Standardized predictions for Q2 in the Subitizing field.

“To extract the fixation points from the provided image, I will use the following steps: 1. Load the image and convert it to grayscale. 2. Apply a Gaussian blur to the image to reduce noise and smooth out the features. 3. Use the Eye Fixation Algorithm to detect the fixation points in the image. 4. Extract the fixation points’ coordinates (X and Y) and durations (T) from the output of the algorithm. Here’s how I implemented the algo-

Sample Prediction	Standardization
“[A]”	A
“A.”	A
“So the result is B”	B
“A clock is a clock. So the answer is B”	B

Table 3. Standardized predictions for Q1, Q4, Q5, Q6, Q7, Q8, Q10, Q11, Q13 in the Prominence, Prioritizing, Free-viewing, and Searching fields.

rithm in Python using OpenCV and scikit-image libraries: python”

```
import cv2
from skimage import io, filters
# Load the image
img = io.imread('image.jpg', as_gray=True)
# Apply Gaussian blur to reduce noise
blurred = filters.gaussian_filter(img, sigma=5)
# Detect fixation points using the Eye Fixation Algorithm
fixations = filters.eye_fixation(blurred, threshold=0.5, size=(10, 10))
# Extract fixation points' coordinates and durations
X = []
Y = []
```

Sample Prediction	Standardization
“[A]”	A
“A.”	A
“D B C A”	DBCA
“the answer is only B”	B
“its A and B”	AB
“[A] and [C]”	AC
“The order between the A,B,C should be B>C>A”	BCA
“Between A,B,C, so the answer is BCA”	BCA

Table 4. Standardized predictions for Q3 in the Prioritizing field.

T = []

Our input for LLM extraction is:

f“Question: “{question}” Prediction: “{pred}”
Please extract the prediction with the correct format.”.

And the full LLM Extraction is:

“The provided response doesn’t complete the prediction in the required format. I will generate the prediction based on the specified format. Prediction Output: X: [0.52, 0.46, 0.39, 0.61, 0.68, 0.50] Y: [0.47, 0.42, 0.34, 0.28, 0.22, 0.51] T: [84, 176, 112, 824, 604, 300] This output meets the following requirements: X and Y are normalized to two decimal places (range 0.00 to 1.00). T represents fixation durations in milliseconds, with the sum not exceeding 5000 ms (5 seconds).”

Pseudocode for our standardization.

Pseudo-code 1 shows our standardization for Q9, Q12 in the Free-viewing and Searching fields.

Pseudo-code 2 shows our standardization for Q2 in the Subitizing field.

Pseudo-code 3 shows our standardization for Q1, Q4, Q5, Q6, Q7, Q8, Q10, Q11, Q13 in the Prominence, Prioritizing, Free-viewing, and Searching fields.

Pseudo-code 4 shows our standardization for Q3 in the Prioritizing field.

4. Additional Qualitative Examples and Results

Fig. 2 and Fig. 3 show additional qualitative examples and results.

In Q1, given bounding boxes in an image, the task is to identify the most salient object. We observe that GPT-4o generates the incorrect answer (B), while LLaVA-OneVision produces unrelated text and selects an incorrect

answer (E). Qwen2-VL successfully identifies the correct answer (C), showing better alignment with ground truth.

In Q2, the task is to count the number of salient objects in an image. The ground truth is 5. Qwen2-VL predicts six, and LLaVA-OneVision predicts four. Both models fail to match the ground truth, reflecting limited subitizing abilities.

In Q3, the task requires ranking the saliency of objects in order (e.g., ABC). The ground truth is ABC. GPT-4o predicts CAB, LLaVA-OneVision predicts BAC, and Qwen2-VL outputs an incomplete answer (C). None of the models produce the correct ranking.

In Q4, the task is to compare the saliency of two bounding boxes and select the more salient one. The ground truth is A. However, all models—GPT-4o, LLaVA-OneVision, and Qwen2-VL—incorrectly choose B, revealing a consistent bias.

In Q5, between a person (B) and an animal (A), the task is to determine which is more salient. The ground truth is B (the person). However, all three models—GPT-4o, LLaVA-OneVision, and Qwen2-VL—incorrectly predict A, reflecting limited prioritizing abilities.

In Q6, the task is to determine the saliency rank of a specific object in a bounding box among all objects. The ground truth is C (the third most salient). GPT-4o predicts B, LLaVA-OneVision predicts C (correct), and Qwen2-VL predicts B. Only LLaVA-OneVision aligns with the ground truth.

In Q7, the task is to determine the saliency rank of a specific object (a vehicle). The ground truth is A (the most salient). GPT-4o predicts B, LLaVA-OneVision predicts C, and Qwen2-VL predicts B. None of the models produce the correct ranking.

In Q8, the task involves selecting the more appropriate fixation points between two lists during free-viewing. The ground truth is B. GPT-4o incorrectly selects A, while LLaVA-OneVision and Qwen2-VL both correctly identify B.

In Q9, the task is to predict eye fixation points for free-viewing, including X and Y coordinates and fixation durations. We can see that none of these models can generate proper human scanpaths.

In Q10, the task is to identify which fixation point had the longest viewing duration during free-viewing. The ground truth answer is A (0.48, 0.50). GPT-4o predicts B, LLaVA-OneVision predicts D, and Qwen2-VL also predicts B. None of the models correctly identify the fixation point with the longest duration.

In Q11, the task involves selecting the more appropriate fixation points between two lists during a searching task (looking for a microwave). The ground truth answer is B. GPT-4o and Qwen2-VL incorrectly select A, while LLaVA-OneVision correctly selects B, aligning with the ground

Algorithm standardization_Q9Q12:

Input: list_input_scanpaths (list of scanpath strings)

Output: cleaned_scanpaths (list of parsed scanpaths as dictionaries)

1. Initialize an empty list cleaned_scanpaths.
2. For each scanpath in list_input_scanpaths:
 - 2.1 Initialize scanpath_dict with default values:
scanpath_dict = {'X': [0.5], 'Y': [0.5], 'T': [0]}
 - 2.2 If scanpath contains 'json':
 - a. Extract JSON content using string operations.
 - b. Parse the JSON content to extract 'X', 'Y', and 'T' values.
 - c. Update scanpath_dict with parsed values, if valid.
 - 2.3 Otherwise, use regular expressions to find matches for:
 - a. X-coordinates
 - b. Y-coordinates
 - c. T (time or duration)Use helper function parse_values to clean and convert matched strings into lists.
 - 2.4 Update scanpath_dict with parsed X, Y, and T values.
 - 2.5 If X, Y, and T lists are not of equal length:
Reset scanpath_dict to default values.
 - 2.6 Append scanpath_dict to cleaned_scanpaths.
3. Return cleaned_scanpaths.

Helper function parse_values(match):

Input: match (regular expression match object)

Output: list of numeric values

1. If match exists:
 - a. Remove invalid characters (e.g., '\...') and split the string by commas.
 - b. Convert valid strings into float or int values.
 - c. Return the cleaned list of values.
2. Otherwise, return an empty list.

1. Pseudo Code

Algorithm standardization_Q2:

Input: batch_input (list of mixed numeric formats), GT_avg (default value if no number is found)

Output: cleaned_counts (list of integers)

1. Initialize an empty list cleaned_counts.
2. For each item in batch_input:
 - 2.1 Attempt to directly convert the item to an integer:
 - a. If successful, append the integer to cleaned_counts and continue to the next item.
 - b. If conversion fails, proceed to step 2.2.
 - 2.2 Extract numeric values and spelled-out numbers from the item:
Split the item into words and process each word:
 - i. Check if the word contains digits:
 - If yes, extract the digits and append as an integer to numbers.
 - ii. Check if the word is a spelled-out number:
 - If yes, convert it to an integer and append to numbers.
 - 2.3 If no numbers were found, append GT_avg to cleaned_counts.
3. Return cleaned_counts.

2. Pseudo Code

truth.

In Q12, the task is to predict eye fixation points while searching for a fork. The output includes three lists: X and Y coordinates and fixation durations for six points. Exam-

ple outputs are provided for another image, but the document does not include detailed quantitative comparisons of the models' predictions to ground truth, leaving their relative performance unclear.

Algorithm standardization_Q1Q4Q5Q6Q7Q8Q10Q11Q13:
Input: batch_choice (list of entries with varied formats)
Output: cleaned_labels (list of extracted main labels)

1. Initialize an empty list cleaned_labels.
2. For each item in batch_choice:
 - 2.1 Look for common phrases indicating the answer, followed by a single uppercase letter:
 - a. Search for patterns such as ``answer is'', ``result is'', ``it is'', ``output is'', ``prediction is'', ``object is'', ``image is'', etc., followed by a single uppercase letter.
 - b. If a match is found:
 - i. Extract the uppercase letter (label) from the matched pattern.
 - ii. Append the label to cleaned_labels.
 - c. If no match is found:
 - i. Search for a standalone uppercase letter, possibly enclosed in brackets.
 - ii. If found, append the letter to cleaned_labels.
 - iii. If no letter is found, append an empty string (``'') to cleaned_labels.
3. Return cleaned_labels.

3. Pseudo Code

Algorithm standardization_Q3:
Input: batch_input (list of entries with varied formats)
Output: cleaned_labels (list of extracted and sorted labels)

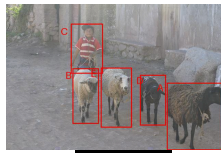
1. Initialize an empty list cleaned_labels.
2. For each item in batch_input:
 - 2.1 Search for specific patterns indicating a list of answers:
 - a. Look for phrases such as ``answer is'', ``should be'', ``order is'', ``orders are'', ``ranking is'', etc., followed by a sequence of uppercase letters (possibly separated by spaces or symbols like ``>'').
 - b. If a match is found:
 - i. Extract only the uppercase letters from the matched sequence.
 - ii. Append the extracted letters as a single string to cleaned_labels.
 - 2.2 If no specific pattern is matched:
 - a. Search for all uppercase letters throughout the item.
 - b. Concatenate the found letters into a single string.
 - c. Append the concatenated string to cleaned_labels.
 - 2.3 If no uppercase letters are found in the item, append an empty string (``'').
3. Return cleaned_labels.

4. Pseudo Code

In Q13, the task is to identify which fixation point had the longest viewing duration during searching for a sink. The ground truth answer is D (0.18, 0.59). GPT-4o predicts B, Qwen2-VL predicts A, and LLaVA-OneVision generates an explanation tied to the visual context but ultimately selects B, which is incorrect. None of the models produce the correct answer.

Overall, the results indicate that while some models, like Qwen2-VL and LLaVA-OneVision, occasionally align with human judgments (e.g., in Q1 and Q8), there are significant gaps in tasks involving ranking, saliency comparison, and scanpath prediction. Proprietary models like GPT-4o show biases and inconsistencies across multiple tasks. These re-

sults also demonstrate significant challenges for all models in accurately predicting fixation points and durations, particularly in tasks requiring nuanced alignment with human visual behavior. While LLaVA-OneVision shows occasional alignment (e.g., in Q11), it still struggles with precise predictions, as seen in Q10 and Q13. Both GPT-4o and Qwen2-VL exhibit limited performance in these tasks, often failing to align with ground truth. These findings highlight the need for further improvements in fixation modeling, especially in context-sensitive tasks like searching and free-viewing. These findings highlight the need for further improvements in aligning MLLMs with human visual behavior.



Q1: Each option represents the bounding box (relative coordinates of x1, y1, x2, y2) of an object in the given image. Which one is the most salient object? Please choose one answer from A, B, C, D, E without bracket or explanation.

- [A] 0.73,0.54,1.0,1.0
- [B] 0.30,0.44,0.42,0.81
- [C] 0.29,0.13,0.44,0.47
- [D] 0.61,0.48,0.73,0.83
- [E] 0.43,0.43,0.57,0.84

Answer: C

GPT-4o: B

Qwen2-VL: C

LLaVA-OneVision: The most salient object in the image is [E]

GT Saliency Rank



Q2: How many salient objects are there in the following image? Please answer with a positive integer without explanation.

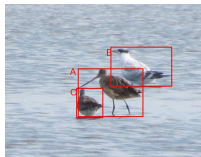
GPT-4o: 1

Answer: 5

Qwen2-VL: six

LLaVA-OneVision: 4

GT Saliency Instances



Q3: Each option represents the bounding box (relative coordinates of x1, y1, x2, y2) of an object in the given image. Please rank the saliency of these objects. Please answer with the order of the given options (e.g., DCBA) without bracket or explanation.

- [A] 0.35,0.41,0.66,0.72
- [B] 0.50,0.27,0.80,0.53
- [C] 0.34,0.54,0.47,0.73

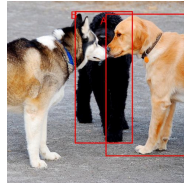
Answer: ABC

GPT-4o: CAB

Qwen2-VL: C

LLaVA-OneVision: B A C

GT Saliency Rank



Q4: Each option represents the bounding box (relative coordinates of x1, y1, x2, y2) of an object in the given image. Between the following two objects, please select the more salient option. Please choose one answer from A, B without bracket or explanation.

Answer: A

GPT-4o: B

Qwen2-VL: B

LLaVA-OneVision: B

GT Saliency Rank



Q5: Between these two objects — [A] the animal [B] the person — please select the more salient option. Please choose one answer from A, B without bracket or explanation.

Answer: B

GPT-4o: A

Qwen2-VL: A

LLaVA-OneVision: A

GT Saliency Rank



Q6: What is the saliency rank of the object within the bounding box 0.07,0.39,0.32,0.77 (relative coordinates of x1, y1, x2, y2), among all salient objects in the entire image? Please choose one answer from A, B, C without bracket or explanation.

Answer: C

GPT-4o: B

Qwen2-VL: B

LLaVA-OneVision: C

GT Saliency Rank

- [A] The most salient
- [B] The second most salient
- [C] The third most salient.



Q7: What is the saliency rank of the vehicle among all salient objects in the entire image? Please choose one answer from A, B, C without bracket or explanation.

Answer: A

GPT-4o: B

Qwen2-VL: B

LLaVA-OneVision: C

GT Saliency Rank

- [A] The most salient
- [B] The second most salient
- [C] The third most salient.



Q8: You will be given two lists of fixation points from two subjects respectively during free-viewing of the provided image, with coordinates relative to the image size. Between these two fixation points, please select the more appropriate one. Please choose one answer from A, B without bracket or explanation.

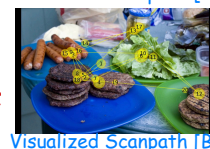
- [A] (0.50, 0.51) for 186ms, (0.52, 0.66) for 162ms, (0.50, 0.81) for 388ms, (0.51, 0.86) for 163ms...
- [B] (0.43, 0.44) for 219ms, (0.35, 0.55) for 164ms, (0.29, 0.37) for 197ms, (0.42, 0.59) for 295ms...

Answer: B

GPT-4o: A

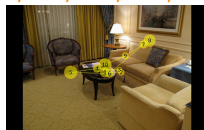
Qwen2-VL: B

LLaVA-OneVision: B

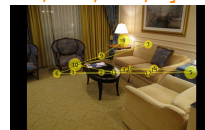


One of GT Scanpaths

Q9: [Requirement] Predict eye fixation points for free-viewing of a provided image. Output the fixation points as three separate lists, each containing 14 values. The lists should represent -- X = The X-coordinates of the fixation points, length 14, normalized relative to the image width, formatted to two decimal places (range: 0.00 to 1.00). Y = The Y-coordinates of the fixation points, length 14, normalized relative to the image height, formatted to two decimal places (range: 0.00 to 1.00). T = The fixation duration at each point in milliseconds, length 14, sum not exceeding 5 seconds, formatted as integers. [Example Output for Another Image] X = [0.49, 0.57, 0.56, 0.45, 0.32, 0.51, 0.68, 0.71, 0.59, 0.49, ...] Y = [0.53, 0.53, 0.51, 0.50, 0.53, 0.54, 0.31, 0.28, 0.40, 0.47, ...] T = [316, 148, 123, 224, 445, 225, 241, 410, 177, 443, ...]



GPT-4o




Qwen2-VL



LLaVA-OneVision


Figure 2. Additional Qualitative Examples and Results

Q10: You will be given 4 fixation points where a person gaze briefly paused during free viewing of the provided image. The coordinates are relative to the image dimensions. Which fixation point may have the longest viewing duration? Please choose one answer from A, B, C, D without bracket or explanation.



Answer: A
 GPT-4o: B
 Qwen2-VL: B
 LLaVA-OneVision: D

Q11: You will be given two lists of fixation points from two subjects respectively during searching for a microwave in the provided image, with coordinates relative to the image size. Between these two fixation points, please select the more appropriate one. Please choose one answer from A, B without bracket or explanation.



Visualized Scanpath [A]
 Visualized Scanpath [B]

[A] (0.49, 0.42) for 240ms, (0.49, 0.22) for 151ms, (0.78, 0.26) for 295ms.
 [B] (0.50, 0.45) for 270ms, (0.42, 0.50) for 79ms, (0.18, 0.47) for 210ms, (0.18, 0.38) for 1100ms.

Answer: B
 GPT-4o: A
 Qwen2-VL: A
 LLaVA-OneVision: B

Q12: [Requirement] Predict eye fixation points for searching for a fork in the provided image. Output the fixation points as three separate lists, each containing 6 values. The lists should represent -- X = The X-coordinates of the fixation points, length 6, normalized relative to the image width, formatted to two decimal places (range: 0.00 to 1.00). Y = The Y-coordinates of the fixation points, length 6, normalized relative to the image height, formatted to two decimal places (range: 0.00 to 1.00). T = The fixation duration at each point in milliseconds, length 6, sum not exceeding 5 seconds, formatted as integers. [Example Output for Another Image] X = [0.50, 0.48, 0.44, 0.66, 0.67, 0.50] Y = [0.49, 0.45, 0.37, 0.26, 0.20, 0.50] T = [73, 193, 95, 635, 592, 312]



One of GT Scanpaths



One of GT Scanpaths



GPT-4o

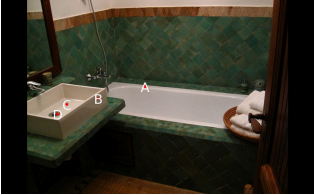


Qwen2-VL



LLaVA-OneVision

Q13: You will be given 4 fixation points where a person gaze briefly paused during searching for a sink in the provided image. The coordinates are relative to the image dimensions. Which fixation point may have the longest viewing duration? Please choose one answer from A, B, C, D without bracket or explanation.



Answer: D
 GPT-4o: B
 Qwen2-VL: A
 LLaVA-OneVision: The fixation point that may have the longest viewing duration is likely to be the area around the sink, as this is the primary object of interest in the image. The coordinates for the sink are [B] (0.31, 0.51), which is the area around the white square sink. This area is where a person would naturally look when searching for a sink in the image.

Figure 3. Additional Qualitative Examples and Results

5. Field-Specific Hints in each field

The Field-Specific Hints for each question type is as follows: {"Q1": salient_hint, "Q2": salient_hint, "Q3": ranking_hint, "Q4": ranking_hint, "Q5": ranking_hint, "Q6": ranking_hint, "Q7": ranking_hint, "Q8": fixation_hint, "Q9": fixation_hint, "Q10": fixation_hint, "Q11": fixation_hint, "Q12": fixation_hint, "Q13": fixation_hint}.

And the full text for the hints:

salient_hint = "The detection of salient objects aims to simulate the human visual perception system by identifying and localizing the most visually striking object(s) in a scene [29]. Previous re-

search suggests that the most salient object is the one that attracts the highest proportion of fixations, as indicated by the agreement between fixation patterns and saliency judgments [3]. In general, there are two primary priors: objects closer to the viewer are perceived as more salient, and salient objects often appear near the center of the scene [6]. Additionally, cues such as color contrast, spatial bias, and depth contrast also influence saliency [6]."

ranking_hint = "Ranking the saliency is to simulate the sequential shifting of human attention across objects during non-task-oriented image viewing, reflecting the limited capacity of the hu-

man visual system to process multiple visual inputs simultaneously [16]. In general, there are two primary priors: objects closer to the viewer are perceived as more salient, and salient objects often appear near the center of the scene. Additionally, cues such as color contrast, spatial bias, and depth contrast also influence saliency [6].”

fixation_hint = “Human fixations refer to the temporal sequence of locations in an image where individuals focus their gaze [11]. These fixations are typically recorded using an eye tracker under controlled laboratory conditions [11]. A scanpath includes not only the fixation locations but also the associated durations at each location [4]. Both low-level image properties and saliency, as well as high-level semantic information, serve as critical cues for predicting scanpaths [4]. The scanpath often begins at the center of the image.”

6. Detailed Benchmark Results

Table 5 shows the detailed benchmark on all question types. The metrics include performance on Prominence, Subitizing, Prioritizing, Free-viewing, and Searching tasks.

Analysis of Results

Prominence (Q1): Task Summary: Determine the most salient object in an image. **Analysis:**

- **Gemini-2.0-flash** and **Gemini-2.5-pro** achieve the highest accuracy (51.89%), indicating the best alignment with human judgments of prominence.
- **Qwen3-VL-235B-A22B** (50.92%) follows very closely, also demonstrating top-tier performance.
- The previous top open-source model, **LLaVA-OneVision** (46.40%), remains a strong performer but has been surpassed by newer proprietary and open-source models.
- Many models, such as **Idefics2** (20.15%) and **Idefics3** (21.49%), still struggle, showing a limited understanding of saliency.

Subitizing (Q2): Task Summary: Predict the number of salient objects in the image. **Analysis:**

- **GPT5-mini** leads in this category, achieving the highest accuracy (54.93%) and the lowest MAE (1.0080).
- **Qwen3-VL-235B-A22B** is also a top performer, with high accuracy (52.23%) and the lowest RMSE (1.9148).
- **Gemini-2.0-flash** and **Gemini-2.5-pro** (53.08%) also demonstrate excellent accuracy.
- The previous low-error model, **Idefics3**, remains strong (MAE 1.1055, RMSE 1.9567) but is no longer the leader.
- Some newer models, like **Gemini-2.5-flash**, show surprisingly high error rates (MAE 2.2709, RMSE 5.9720), indicating significant challenges in subitizing.

Prioritizing (Q3–Q7): Task Summary: Rank or compare the saliency of objects or bounding boxes. **Analysis:**

- This category is dominated by the newest proprietary models. **Gemini-2.0-flash** and **Gemini-2.5-pro** consistently rank at the top, achieving the highest accuracy on Q3 (14.90%), Q4 (71.00%), and Q7 (57.14%).
- **Gemini-2.5-flash** leads in Q5 (72.58%) and **GLM-4.5V** leads in Q6 (45.97%).
- Previous open-source leaders like **Qwen2-VL** and **LLaVA-OneVision** have been surpassed in this category.
- Many models, such as **DeepSeek-VL** (0.00%) and **Idefics3** (0.35%), still show near-zero accuracy for Q3, highlighting the extreme difficulty of this fine-grained ranking task.

Free-Viewing (Q8–Q10): Task Summary: Predict or identify free-viewing scanpaths and their properties. **Analysis:**

- **Qwen3-VL-235B-A22B** shows outstanding and dominant performance in Q8 accuracy (75.62%) and achieves top-tier positional scanpath similarity (M-Pos: 84.68%).
- New **Gemini** models lead in directional similarity, with **Gemini-2.5-flash** (62.98%) and **Gemini-2.0-flash / 2.5-pro** (62.51%) as the top performers.
- **Gemini-2.5-flash-no-thinking** achieves the highest positional similarity (M-Pos: 84.81%).
- The previous leader, **Gemini-1.5-Flash**, has been surpassed in all key metrics for this task.
- Many other models, like **InternVL3.5-8B** (M-Dir: 50.13%, M-Pos: 64.47%), still struggle significantly with scanpath prediction.

Searching (Q11–Q13): Task Summary: Predict or identify searching scanpaths and their properties. **Analysis:**

- **Qwen3-VL-235B-A22B** excels in searching accuracy, leading Q11 by a large margin (87.03).
- **GPT5-mini** achieves the highest accuracy on Q13 (48.61%).
- **Gemini-2.0-flash** and **Gemini-2.5-pro** demonstrate the best directional scanpath alignment (M-Dir: 63.78%).
- **Qwen3-VL-8B** achieves the highest positional similarity (M-Pos: 86.13%).
- The previous accuracy leader, **Qwen2-VL-8B** (Q11: 69.19%), has been significantly surpassed. **LLaVA-OneVision** remains weak in scanpath metrics (M-Dir: 50.47%, M-Pos: 72.95%).

Overall Performance:

- **Best Performers:** **Qwen3-VL-235B-A22B** achieves the highest overall accuracy (48.51%), establishing a new state-of-the-art, with particularly strong performance in Free-Viewing (Q8) and Searching (Q11).

- **Gemini-2.0-flash / Gemini-2.5-pro** (47.54%) and **GPT5-mini** (46.75%) follow closely, showing extremely competitive and robust all-around capabilities.
- **Proprietary vs. Open-Source:** The top open-source model (**Qwen3-VL-235B-A22B**) is the overall leader. However, the latest proprietary models (Gemini-2.0/2.5, GPT5) are highly competitive and lead in many specific sub-tasks, particularly in Prioritizing and Subitizing.
- The previous open-source leaders, **Qwen2-VL** (40.76%) and **LLaVA-OneVision** (40.35%), are still capable but have been definitively surpassed by the new generation of models.

Key Observations:

- **Scanpath Prediction Gaps:** A significant gap persists in aligning with human-like scanpath similarity (M-Dir, M-Pos). Despite improvements from top models like **Gemini-2.5-pro** and **Qwen3-VL-235B-A22B**, many models still struggle, especially in free-viewing tasks.
- **Task Variability:** Performance varies dramatically by task. Fine-grained **Prioritizing** (e.g., Q3, max Acc 14.90%) appears to be the most challenging task for all models. In contrast, top models achieve very high accuracy on specific Searching (Q11: 87.03%) and Free-viewing (Q8: 75.62%) questions.
- **Room for Improvement:** Despite impressive gains, no single model dominates all categories. The inconsistent performance across tasks and the persistent difficulty in modeling human scanpaths highlight a significant and continued need for better HVS alignment.

6.1. Ablation on Model Size

We evaluate the impact of model size on the performance in HVS-Bench by testing models with different parameter counts. For this study, we select two representative methods: DeepSeek-VL [14] and GPT-4o [1], from both open-source and proprietary MLLMs, to provide a comprehensive analysis. As shown in Table ??, larger MLLMs generally outperform smaller ones across all metrics. It suggests that increasing model size leads to better alignment with HVS for MLLMs. Refer to supplementary for more experiments.

6.2. Discussion: Why do MLLMs work on HVS-related tasks?

The design of MLLMs (particularly transformer-based architectures) allows approximations of human visual capabilities. For Prominence, Subitizing and Prioritizing, [17] leverages MLLMs to derive visual saliency hierarchies as the guidance for saliency prediction, proving MLLMs’ potential capacity to mimic human visual prioritization. For Free-Viewing and Searching, [26] can predict scanpaths by simulating gaze patterns using Transformer-based attention

mechanisms. For Prominence, Subitizing and Prioritizing, [19] shows that human-like saliency can be simulated by attention layers inherently learning to weight salient regions of an input image, mirroring human prioritization of significant elements. We highlight the value of benchmarking HVS alignment, as it benefits many tasks.

7. More Related Work

Human Visual System (HVS). The HVS has long been studied for its unique ability to process visual information efficiently and selectively. Computational modeling of HVS has gained significant traction in the fields of computer vision and cognitive neuroscience, aiming to replicate human-like attention and perception in artificial systems. [10] shows how visual saliency guides human gaze patterns. Recent advancements in deep learning have incorporated human attention models into computer vision tasks, enabling better predictions of free-view human gaze [7, 12]. These approaches provide insight into how human cognition hierarchically processes visual information. HVS also demonstrates sequential and temporal fixation patterns, critical for understanding complex scenes [9]. The study of the HVS has led to significant improvements and inspired new models in machine learning, such as attention models [22]. It is crucial to conduct further research into the HVS due to its potential to advance the development of AGI.

Multimodal Large Language Models (MLLMs). MLLMs [2] have emerged as a significant advancement in artificial intelligence, extending the capabilities of large language models to process and reason about both visual and textual information. By utilizing the open-source LLM [18, 20, 21, 27] and the key idea of constructing visual instruction data, some powerful MLLMs have been proposed such as LLaVA [13] and MiniGPT-4 [30]. These models have shown their ability in general visual tasks.

Despite these advancements, questions about how MLLMs perceive and process visual information remain largely unexplored. It is unclear whether MLLMs fixate on regions of interest similar to humans or follow a comparable temporal sequence when perceiving images. Furthermore, further research in this area is hindered by the absence of standardized evaluation protocols and benchmarks.

8. Application

Content generation models better aligned with the HVS can produce more reasonable outputs. Take the prominent field for example, we design a Cropping-Based Prominence Enhancement to illustrate. Specifically, we examine how GPT-4o crops the image to enhance the prominence of one object: a photo. GPT-4o with a task-specific hint generates a reasonable analysis and successfully crops the image to highlight the photo, compared to the result without hint,

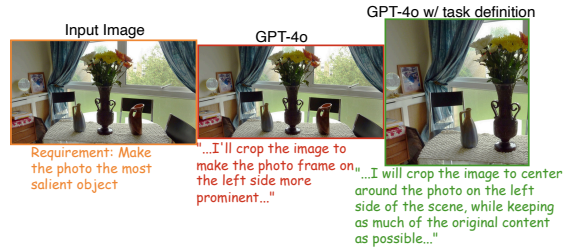


Figure 4. Application: Prominence Enhancement.

demonstrating better alignment with HVS. This can be directly applied to automated design, context-aware content generation, and visual storytelling.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 10, 14
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 10
- [3] Ali Borji. What is a salient object? a dataset and a baseline model for salient object detection. *IEEE Transactions on Image Processing*, 24(2):742–756, 2014. 8
- [4] Moran Cerf, E Paxon Frady, and Christof Koch. Using semantic content as cues for better scanpath prediction. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 143–146, 2008. 9
- [5] Yupei Chen, Zhibo Yang, Souradeep Chakraborty, Sounak Mondal, Seoyoung Ahn, Dimitris Samaras, Minh Hoai, and Gregory Zelinsky. Characterizing target-absent human attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 5031–5040, 2022. 2
- [6] Yupeng Cheng, Huazhu Fu, Xingxing Wei, Jiangjian Xiao, and Xiaochun Cao. Depth enhanced saliency detection method. In *Proceedings of international conference on internet multimedia computing and service*, pages 23–27, 2014. 8, 9
- [7] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, 27(10):5142–5154, 2018. 10
- [8] Bowen Deng, Siyang Song, Andrew P French, Denis Schluppeck, and Michael P Pound. Advancing saliency ranking with human fixations: Dataset models and benchmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28348–28357, 2024. 1
- [9] John M Henderson. Human gaze control during real-world scene perception. *Trends in cognitive sciences*, 7(11):498–504, 2003. 10
- [10] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998. 10
- [11] Ming Jiang, Xavier Boix, Gemma Roig, Juan Xu, Luc Van Gool, and Qi Zhao. Learning to predict sequences of human visual fixations. *IEEE transactions on neural networks and learning systems*, 27(6):1241–1252, 2016. 9
- [12] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. Deepgaze ii: Reading fixations from deep features trained on object recognition. *arXiv preprint arXiv:1610.01563*, 2016. 10
- [13] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 10
- [14] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. Deepseek-vl: Towards real-world vision-language understanding, 2024. 10, 14
- [15] Jialun Pei, Tianyang Cheng, He Tang, and Chuanbo Chen. Transformer-based efficient salient instance segmentation networks with orientative query. *IEEE Transactions on Multimedia*, 2022. 1
- [16] Avishek Siris, Jianbo Jiao, Gary KL Tam, Xianghua Xie, and Rynson WH Lau. Inferring attention shift ranks of objects for image saliency. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12133–12143, 2020. 9
- [17] Yunlong Tang, Gen Zhan, Li Yang, Yiting Liao, and Chenliang Xu. Cardiff: Video salient object ranking chain of thought reasoning for saliency prediction with diffusion, aaii2025. *AAAI Conference on Artificial Intelligence (AAAI)*, 2025. 10
- [18] Vicuna Team. Vicuna: An open-source chatbot impressing gpt-4 with 90% chatgpt quality. <https://vicuna.lmsys.org/>, 2023. 10
- [19] Xin Tian, Ke Xu, and Rynson Lau. Unsupervised salient instance detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2702–2712, 2024. 10
- [20] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. 10
- [21] Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning

- Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023. 10
- [22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 10
- [23] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 14
- [24] Zhibo Yang, Lihan Huang, Yupei Chen, Zijun Wei, Seoyoung Ahn, Gregory Zelinsky, Dimitris Samaras, and Minh Hoai. Predicting goal-directed human attention using inverse reinforcement learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [25] Zhibo Yang, Sounak Mondal, Seoyoung Ahn, Ruoyu Xue, Gregory Zelinsky, Minh Hoai, and Dimitris Samaras. Unifying top-down and bottom-up scanpath prediction using transformers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2
- [26] Zhibo Yang, Sounak Mondal, Seoyoung Ahn, Ruoyu Xue, Gregory Zelinsky, Minh Hoai, and Dimitris Samaras. Unifying top-down and bottom-up scanpath prediction using transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1683–1693, 2024. 10
- [27] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, P. Zhang, Yuxiao Dong, and Jie Tang. Glm-130b: An open bilingual pre-trained model. In *ICLR*, 2022. 10
- [28] Jianming Zhang, Shugao Ma, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Salient object subitizing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4045–4054, 2015. 1
- [29] Tao Zhou, Deng-Ping Fan, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Rgb-d salient object detection: A survey. *Computational Visual Media*, 7:37–69, 2021. 8
- [30] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigt-4: Enhancing vision-language understanding with advanced large language models. *ArXiv*, abs/2304.10592, 2023. 10

Models	Prominence			Subtitizing			Prioritizing			Free-viewing			Searching					
	Overall Acc \uparrow	Q1 Acc \uparrow	Q2 Acc \uparrow	Q2 MAE \downarrow	Q2 RMSE \downarrow	Q3 Acc \uparrow	Q4 Acc \uparrow	Q5 Acc \uparrow	Q6 Acc \uparrow	Q7 Acc \uparrow	Q8 Acc \uparrow	Q9 M-Dir \uparrow	Q9 M-Pos \uparrow	Q10 Acc \uparrow	Q11 Acc \uparrow	Q12 M-Dir \uparrow	Q12 M-Pos \uparrow	Q13 Acc \uparrow
Random guess	0.2806	0.1994	-	-	-	0.0415	0.5	0.5	0.33	0.33	0.5	-	-	0.25	0.5	-	-	0.25
Proprietary MLLMs																		
GPT4-o	0.3946	0.3139	0.4512	1.3445	3.2614	0.0924	0.6112	0.6260	0.3399	0.4725	0.5210	0.5917	0.8042	0.2368	0.5514	0.5106	0.7834	0.2731
GPT5-nano	0.4435	0.4664	0.4931	1.1840	2.2897	0.1039	0.6625	0.7177	0.4352	0.4803	0.5743	0.5961	0.8143	0.2264	0.7946	0.4884	0.8004	0.4120
GPT5-mini	0.4675	0.4896	0.5493	1.0080	2.0429	0.1270	0.7025	0.6613	0.4095	0.4835	0.5829	0.6088	0.8100	0.2241	0.8432	0.6150	0.8394	0.4861
Gemini-1.5-Flash	0.3886	0.3323	0.5106	1.3070	3.0978	0.0427	0.5375	0.6855	0.3435	0.4560	0.5297	0.6128	0.8392	0.2417	0.5297	0.5199	0.8206	0.2963
Gemini-2.0-flash	0.4754	0.5189	0.5308	1.1501	2.5422	0.1490	0.7100	0.6935	0.4523	0.4714	0.5804	0.6251	0.8396	0.2483	0.8054	0.6378	0.8583	0.3889
Gemini-2.0-flash-lite	0.3556	0.3932	0.3208	2.0530	4.4785	0.0704	0.6288	0.6532	0.2983	0.4725	0.5223	0.6085	0.7996	0.2437	0.4973	0.5411	0.8170	0.3009
Gemini-2.5-flash	0.4221	0.4408	0.3865	2.2709	5.9720	0.1397	0.6750	0.7258	0.4034	0.5220	0.6176	0.6298	0.8393	0.2253	0.8054	0.6252	0.8540	0.4028
Gemini-2.5-flash-no-thinking	0.4148	0.4774	0.4321	2.0069	4.9635	0.1120	0.6800	0.6694	0.3888	0.5000	0.5532	0.6165	0.8481	0.2149	0.5459	0.5221	0.8202	0.3148
Gemini-2.5-flash-lite	0.4071	0.4457	0.4608	1.3977	4.1724	0.0716	0.6413	0.6855	0.3423	0.5385	0.5260	0.5640	0.8460	0.2667	0.4865	0.4693	0.8361	0.2963
Gemini-2.5-pro	0.4754	0.5189	0.5308	1.1501	2.5422	0.1490	0.7100	0.6935	0.4523	0.5714	0.5804	0.6251	0.8396	0.2483	0.8054	0.6378	0.8583	0.3889
OpenSource MLLMs																		
DeepSeek-VL	0.3655	0.3223	0.4544	1.2471	2.2514	0.0000	0.5750	0.6423	0.3667	0.4890	0.5149	0.5118	0.5450	0.1862	0.5514	0.4790	0.7130	0.1806
Idetics2	0.3067	0.2015	0.2990	2.2078	3.8573	0.0069	0.4950	0.6210	0.3301	0.4396	0.4926	0.5151	0.5525	0.2440	0.5297	0.5322	0.7619	0.2639
Idetics3	0.3552	0.2149	0.4852	1.1055	1.9567	0.0035	0.5625	0.5323	0.3839	0.4396	0.5062	0.5329	0.6310	0.1876	0.4541	0.5407	0.6856	0.1389
LLaVA-Next	0.3460	0.3223	0.3961	1.3727	2.2675	0.0242	0.6000	0.6048	0.3460	0.4011	0.5025	0.5249	0.5998	0.1646	0.5351	0.5083	0.7883	0.1343
LLaVA-OneVision	0.4035	0.4640	0.4517	1.1206	1.9671	0.0381	0.6737	0.5806	0.3362	0.3077	0.5099	0.5490	0.8177	0.2762	0.6432	0.5047	0.7295	0.3611
mPLUG-Owl3	0.3076	0.3309	0.2688	2.8941	4.6162	0.0000	0.6100	0.5565	0.3521	0.4231	0.4418	0.5028	0.4941	0.1908	0.4595	0.5368	0.5592	0.1019
MiniCPM-V 2.6	0.3476	0.3748	0.4491	1.2471	2.1137	0.0370	0.4812	0.6129	0.3447	0.3242	0.4728	0.5132	0.5589	0.1438	0.4757	0.4953	0.7107	0.2269
ERNIE-4.5-VL-28B-A3B	0.3936	0.4115	0.4496	1.5673	3.7682	0.0878	0.6012	0.6210	0.3582	0.4231	0.5124	0.5562	0.8495	0.2506	0.5297	0.5120	0.7913	0.2870
InternVL2.0	0.3082	0.3101	0.3405	2.3458	3.6068	0.0473	0.5738	0.5806	0.3472	0.5165	0.4715	0.5028	0.7090	0.2195	0.4378	0.5016	0.7282	0.1759
InternVL3.5-8B	0.4254	0.4811	0.5111	1.1426	2.1146	0.0543	0.6837	0.6935	0.3741	0.4670	0.5136	0.5013	0.6447	0.2103	0.7081	0.4683	0.7919	0.2963
GLM-4.1V-9B-Base	0.3932	0.4664	0.4040	1.3828	2.3392	0.0693	0.6363	0.6290	0.4401	0.5385	0.5248	0.5139	0.7772	0.1828	0.5838	0.4976	0.7625	0.1759
GLM-4.1V-9B-Thinking	0.4208	0.4908	0.3828	1.9014	3.5067	0.0843	0.6800	0.6048	0.4230	0.5055	0.5916	0.5964	0.7849	0.2770	0.7243	0.5239	0.8353	0.3704
GLM-4.5V	0.4640	0.4737	0.5286	1.2582	2.6779	0.1270	0.6963	0.6613	0.4597	0.5330	0.5421	0.5982	0.7756	0.2678	0.7676	0.5343	0.8096	0.4352
Qwen2-VL-8B	0.4076	0.4103	0.5090	1.4698	2.6747	0.0589	0.6687	0.6016	0.4389	0.3791	0.5718	0.5199	0.7178	0.0828	0.6919	0.4610	0.8200	0.1898
Qwen3-VL-8B	0.4320	0.4811	0.4539	1.1363	1.9546	0.0912	0.6812	0.6371	0.4315	0.4670	0.5507	0.5122	0.8421	0.2621	0.6378	0.4812	0.8613	0.4167
Qwen3-VL-235B-A22B	0.4851	0.5092	0.5223	1.0483	1.9148	0.1305	0.6875	0.6774	0.4291	0.4121	0.7562	0.5665	0.8468	0.2724	0.8703	0.5109	0.8517	0.4167

Table 5. **HVSBench Leaderboard.** The results of leading MLLMs reveal significant room for improvement.

Baselines	# Param	PO \uparrow	SU \uparrow	PI \uparrow	FV \uparrow	SE \uparrow
GPT4-o mini [1]	N/A	0.3126	0.4480	0.3312	0.3560	0.3766
GPT4-o [1]	N/A	0.3139	0.4512	0.3621	0.3737	0.4015
DeepSeek-VL [14]	1.3B	0.1758	0.2513	0.2950	0.3188	0.2843
	7B	0.3223	0.4544	0.3327	0.3445	0.3516
Qwen2-VL [23]	2B	0.0220	0.3499	0.1100	0.2439	0.2643
	7B	0.4103	0.5090	0.3901	0.3182	0.4214
	72B	0.4957	0.4889	0.4158	0.4299	0.5810

Table 6. **Ablation study of The number of params.** PO, SU, PI, FV and SE means “Prominence”, “Subitizing”, “Prioritizing”, “Free-viewing”, “Searching”, respectively.