

# OODBENCH: OUT-OF-DISTRIBUTION BENCHMARK FOR LARGE VISION-LANGUAGE MODELS

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Existing Visual-Language Models (VLMs) have achieved significant progress by being trained on massive amounts of data and with the assumption that the data are Independent and Identically Distributed, IID. However, we must embrace the fact that, in real-world scenarios, it is not only difficult but also impractical to expect that all data processed by an AI system would satisfy this assumption. Furthermore, if an AI system chooses to ignore out-of-distribution, OOD, objects during processing, it may cause safety hazards or even lead to catastrophic consequences in real-world applications (*e.g.*, autonomous driving, medical assistance scenarios, *etc.*). Unfortunately, current research has not yet provided valid benchmarks that can comprehensively assess the performance of VLMs in response to OOD data. Therefore, we propose **OODBench**, a predominantly automated method with minimal human verification, for constructing new benchmarks and evaluating the ability of VLMs to process OOD data. OODBench contains 40K instance-level OOD instance category pairs, and we also show that VLMs still struggle to process natural image categories in OODBench, despite those categories are common. In addition, we propose a reliable automated assessment metric that employs a Basic-to-Advanced Progression of prompted questions to assess the impact of OOD data on questions of varying difficulty more fully. Lastly, we summarize substantial findings and insights to facilitate future research in the acquisition and evaluation of OOD data <sup>1</sup>.

## 1 INTRODUCTION

Recently, Visual-Language Models, VLMs (*e.g.*, GPT-4o Achiam et al. (2023); Hurst et al. (2024), Gemini Team et al. (2024), LLaVA family Li et al. (2024b), InternVL family Chen et al. (2024c;a), Qwen2-VL family Wang et al. (2024a), Llama-3.2-Vision Dubey et al. (2024), DeepSeek-VL Lu et al. (2024), *etc.*) have demonstrated impressive excellence in several domains. Benefiting from the fact that these models are trained on enormous amounts of multimodal data, they show significant potential for applications in various industries, driving technological innovations and improving real-world operational efficiencies. These results have led to the widespread belief that these large-scale models already have strong generalization capabilities based on existing knowledge. However, little attention has been paid to the performance of these VLMs in the face of out-of-distribution (OOD) data. The neglect of this issue is partly attributed to the lack of OOD evaluation benchmarks. Therefore, we are eager to construct a benchmark to evaluate the performance of VLMs on OOD data, thereby advancing the progress of VLMs in OOD research and taking a critical step towards realizing intelligent systems that can operate safely in the real world.

In this paper, we aim to obtain OOD data, *i.e.*, those samples that do not belong to the training data distribution. As shown in Fig. 1, OOD data can be classified into two categories according to the different forms of distributional changes: a data distribution shift where labels alter, known as *semantic shift*, and a shift where labels remain the same but the data distribution changes, also known as *covariate shift*. The OOD data are essentially associated with the model, as it relies on the models' assumptions and learned distributions made during the training process. However, most of the mainstream VLMs are closed-source or only open to model parameters, which prevents us from

<sup>1</sup>The dataset is open to the public for research. <https://anonymous.4open.science/r/ood-21C4>

054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107

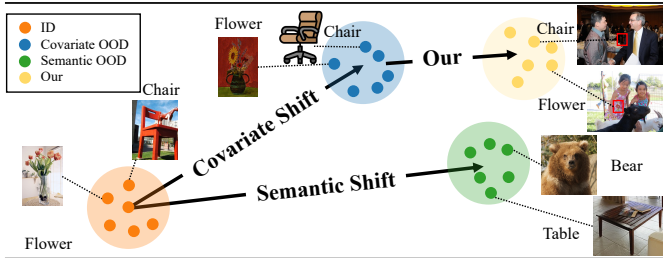


Figure 1: Comparison of differences in ID data, covariate shift OOD data, and semantic shift data.

directly accessing the training distributions of these models. Therefore, how to collect OOD data outside of these model training data distributions seems to be an insurmountable obstacle.

In the traditional OOD literature Yang et al. (2024a); Hendrycks et al. (2022); Wang et al. (2022), these studies usually train models on datasets with known categories and obtain OOD test benchmarks by constructing new category data based on the categories in the training data, thus OOD data are different from the original categories, *i.e.*, semantic shift OOD data. However, considering the training category agnostic of existing VLMs, this experimental setting is obviously no longer applicable. Moreover, the ill-pursuit to collecting rarely unseen data ensures neither the rigor of semantic shift OOD data nor real-world significance. For this reason, our work focuses on collecting covariate shift OOD data to construct OOD benchmarks. The approach is based on the assumption that existing VLMs are likely to have been trained on the most common classes of data.

Recent OOD research Averly & Chao (2023); Yang et al. (2024b); Jaeger et al. (2023); Guérin et al. (2023); Zhu et al. (2024) has shifted the focus from new category detection to failure detection scenarios, in which the OOD detector is expected to recognize misclassified samples. In this approach, candidate samples that fail to be classified are labeled as OOD data, and samples that are correctly classified are labeled as ID data to enhance the safety and reliability of deep learning models in real-world applications. Building upon this approach, we further define two categories of OOD data for VLM scenarios from a human perception perspective: **1) Objects in images that are neither main objects nor semantically related to the main semantic object;** **2) Variants or anomalous forms of target objects.** The underlying intuition is that VLM training heavily relies on the main semantic objects in images, paying limited attention to non-main semantic objects and their variants/anomalies. This constitutes the OOD in VLM semantic learning. Based on this definition, we first divide the data using an OOD detector, followed by manual spot-checking to ensure the final classification aligns with the proposed OOD definition. It is important to note that this model-based classification method is closely tied to the specific OOD detector employed. VLMs themselves should serve as OOD detectors for classifying OOD data. However, testing each VLM on distinct datasets to identify its specific OOD data is computationally and temporally intensive, making it impractical. Therefore, we propose an automated OOD data classification method supplemented by human verification, which maximizes quality while minimizing manual effort.

In the design, we adopt a generalized OOD detector instead of a model-specific VLM detector to reduce computational and time costs. Meanwhile, to alleviate the deviation of OOD data division caused by generalized detectors, we introduce a cross-validation scheme with multiple generalized OOD detectors. For implementation, we have chosen off-the-shelf VLMs such as CLIP Radford et al. (2021) and BLIP2 Li et al. (2023b) as generalized OOD detectors and defined the intersection of OOD data detected by both as OOD-Hard data; the data obtained from symmetric difference are defined as OOD-Simple data, denoted as OOD-H and OOD-S, respectively. It is worth noting that CLIP and BLIP2 are essentially replaceable in our OOD division pipeline: any sufficiently robust classifier can serve as a generalized OOD detector and yield comparable results. In the Appendix, we discuss the implications of using different classifiers as generalized OOD detectors in detail. Based on our assumptions, the VLMs have been trained in most of the common data categories. In other words, the OOD data and the training data share the same labeling space, and thus, the divided OOD data can be summarized as covariate shift OOD. We collected the initial benchmark<sup>2</sup> from a

<sup>2</sup>In Appendix A, we demonstrate the data distribution of the initial benchmark and the detailed performance of different models tested on this benchmark.

publicly available dataset, which contains about 77k yes-or-no samples. To further reduce the testing time, we randomly downsampled the initial benchmark to obtain the main benchmark OODBench.

Fig. 2 illustrates the sample distribution of OODBench, which contains about 40k yes-or-no samples, where OOD-S is 22k and OOD-H is 18k. To evaluate VLMs’ performance in processing OOD data from multiple dimensions, we propose the ‘Basic-Advanced Process Metric’, which evaluates their ability in image understanding, quantity perception, and logical reasoning, respectively.

Our main contributions can be summarized as the following three: 1) We propose a succinct and efficient OOD data division process that is primarily automated and supplemented with lightweight human validation, thereby minimizing labor cost while reducing computational and time overheads. 2) We propose the OOD benchmark for VLMs, which consists of an OOD dataset, an OOD data division pipeline for VLMs, a ‘Basic-to-Advanced Procession Metric’, and a complete set of experimental procedures. 3) We evaluate eight SOTA VLMs on OODBench, and demonstrate through a large number of experiments that current VLMs still perform poorly when confronted with OOD data, even though the classes of these OOD data are pretty common in natural scenarios.

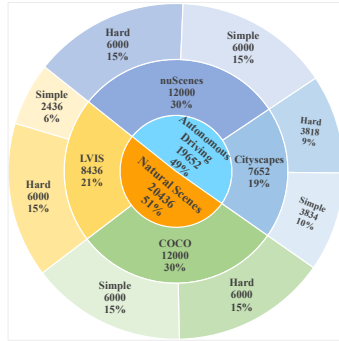


Figure 2: Distribution of categories and fields in **OODBench**.

## 2 RELATED WORK

**Out-of-Distribution.** Out-of-distribution (OOD) detection aims to recognize inputs sampled from a distribution different from the model’s training data Averly & Chao (2023); Yang et al. (2024b). Traditional OOD research mainly focuses on semantic shift (S-OOD), where novel classes arise, yet covariate shift (C-OOD)—where labels remain the same but inputs deviate—can be equally disruptive Yang et al. (2024a); Salehi et al. (2021). In safety-critical fields like autonomous driving, even minor shifts may cause severe misclassifications and catastrophic results.

While robust models may correctly handle such shifts, weaker models often benefit from rejecting uncertain samples to mitigate high-risk errors Liu et al. (2021). Reflecting this, OOD research has gradually shifted from purely detecting unseen classes to failure detection, encompassing both semantic and covariate shifts that can cause misclassification Wiles et al. (2021); Liu et al. (2021). Nevertheless, there remains limited exploration of how modern Vision-Language Models (VLMs) contend with these conditions. Our work addresses this gap by constructing an OOD benchmark for VLMs, emphasizing common categories that remain challenging under real-world variation.

**Multimodal Large Language Models & Benchmark.** Recent Vision-Language Models (VLMs) trained on massive image-text corpora have achieved impressive gains in tasks like visual reasoning, captioning, and Q&A Li et al. (2023b); Team et al. (2024); Achiam et al. (2023). Notable examples include GPT-4 Achiam et al. (2023); Hurst et al. (2024), Gemini Team et al. (2024), LLaVA Li et al. (2024b), and InternVL Chen et al. (2024a;c), often leveraging large-scale datasets such as LAION Schuhmann et al. (2022) and MS COCO Lin et al. (2014).

Although benchmarks like MMBench Liu et al. (2025), SeedBench Li et al. (2023a), and GQA Hudson & Manning (2019) assess general vision-language performance, they mainly cover in-distribution cases and rarely address out-of-distribution (OOD) data. For example, SeedBench offers 19K multiple-choice questions across images and videos, highlighting model limits but rarely exploring out-of-distribution scenarios. To fill the gap, we introduce OODBench, testing VLM robustness under covariate shift. By incorporating simple and hard OOD categories and employing a ‘Basic-to-Advanced’ protocol, OODBench reveals key weaknesses in current models and encourages future work on safer, more reliable multimodal systems.

While OOD data can trigger hallucinations, not all hallucinations arise from distribution shifts. Benchmarks like HallusionBench Guan et al. (2024) study language and visual hallucinations, whereas OODBench systematically applies covariate shifts to measure genuine OOD resilience.

### 3 EXISTING DEFINITIONS AND FORMULATIONS

Out-of-distribution (OOD) refers to samples that originate from a data distribution distinct from that of the training set. Given a VLM  $f$  trained on data sampled from the distribution  $P_{tr}(\mathcal{X}, \mathcal{Y} \in \mathcal{S})$ , where  $\mathcal{X}$  denotes training samples,  $\mathcal{Y}$  the corresponding labels, and  $\mathcal{S}$  the label space. This training distribution is referred to as the in-distribution (ID). In contrast, common OOD data can be divided into two types: covariate shifts (*i.e.*, labels remain unchanged, involving only samples from the training class) and semantic shifts (*i.e.*, labels change, involving only samples from the novel category).

**Semantic Shift.** Lipton et al. (2018) In a traditional experimental setup, the semantic shift data consists of novel category samples outside of the model labeling space  $\mathcal{S}$ . The semantic shift data consists of a novel category of samples. Given semantic labels  $Y_{tr} \in \mathcal{Y}$  from the training distribution and semantic labels  $Y_{te} \in \mathcal{Y}$  from the testing distribution, where  $Y_{tr} \cap Y_{te} = \emptyset$  and satisfy  $P_{tr}(y) = 0$  for all  $y \in Y_{te}$ .

**Covariate Shift.** Shimodaira (2000); Lipton et al. (2018) The covariate shift data consists of samples from the model label space  $\mathcal{S}$  but from different domains. In other words, the image distributions of the training and test data are different:  $P_{tr}(x) \neq P_{te}(x)$ , but the label distributions remain the same:  $P_{tr}(y|x) = P_{te}(y|x)$ .

### 4 THE OODBENCH

VLMs have demonstrated outstanding performance across multiple tasks and domains, profoundly reshaping the landscape of AI development. However, their performance when encountering OOD data remains under-evaluated. Existing VLMs typically use agnostic training categories and collect rare categories as semantic shift OOD data, which do not accurately assess model performance on OOD categories encountered in real-world applications, as these rare categories may not represent common OOD types. To remedy this shortcoming, we focus on collecting covariate shift OOD data for constructing OODBench. Specifically, the OOD data categories need to be broadly representative to ensure their real-world relevance. Meanwhile, OOD data in certain scenarios can severely degrade model performance, potentially causing catastrophic outcomes (*e.g.*, autonomous driving). Thus, we collected data for two key scenarios: natural scenarios (COCO Lin et al. (2014), LVIS Gupta et al. (2019)) and autonomous driving (nuScenes Caesar et al. (2020), Cityscapes Cordts et al. (2016)).

Our goal is to build an instance-level OOD benchmark for VLMs. To this end, we define OOD data from the perspective of VLMs’ semantic learning: 1) Objects in images that are neither main objects nor semantically related to the main semantic object; 2) Variants or anomalous forms of target objects. However, manually screening each instance object that meets the above definition is practically infeasible. To address this issue, we designed a concise and effective OOD data division process, whose overall framework is illustrated in Fig. 3. Specifically, We strictly follow the definition of OOD data division as defined in previous approaches Averly & Chao (2023); Yang et al. (2024b) and label the model-specified misclassified examples as OOD data to improve the safety and reliability of the deep learning model in real-world applications. The underlying intuition is that, taking CLIP Radford et al. (2021) as an example, its large-scale contrastive learning mechanism naturally forms a clustering structure in the image-text embedding space during training, grouping “main semantic objects” and “typical forms.” In contrast, non-main semantic objects, semantic variants, or anomalous forms typically exhibit significant deviations within this space. Therefore, when an image is paired with text describing non-primary semantic content, the unstable or erroneous predictions generated by CLIP fundamentally reflect that “the current semantic distribution of the image-text pair has deviated from its training distribution.” We leverage this anomaly in semantic distance manifested within the embedding space as an auxiliary signal to identify potential OOD candidate samples. Given an image  $x$  and all candidate category labels  $y = \{y_1, \dots, y_i, \dots, y_n\}$ , where  $i$  denotes the  $i$ -th label and  $n$  is the total number of category labels (*e.g.*, the number of cat-

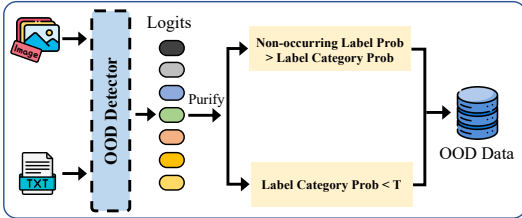


Figure 3: Pipeline of OOD data collection.

egory labels in the COCO dataset  $n = 80$ ). We input both images and labels into the off-the-shelf pretrained OOD detector to obtain logits of  $(x, y)$  pairs.

**Purify.** Since an image may correspond to multiple labels, we note that the corresponding label of the image is  $y^x \in y$ . Direct softmax operation on logits may lead to interference between labels because softmax assumes that each sample can only belong to one category, which makes the final image-category matching probability not accurately reflect the actual probability. For example, when an image contains both **cat** and **dog** categories, softmax calculates the probability by comparing the scores of each category with the scores of the other categories. Therefore, the logits of cat and dog will interact with each other after softmax, which may lead to inaccurate classification results as the cat’s score decreases due to the dog’s score and vice versa. To address this issue, we introduce the purify operation to eliminate the interaction between different  $y^x$ . In the implementation, we select one of the  $y^x$  as the current category and set the logits of the other  $y^x$  to negative infinity, thus eliminating the interference of other labels on the selected label.

After the *purify* operation, we obtained the  $(x, y)$  match probability of the selected labels that are not interfered with by other labeling categories. Subsequently, we perform OOD data collection based on this match probability. In our setting, failure detection is divided into two cases. The first case is that the probability of the category  $y'$  that does not exist on the image is higher than the probability of all label categories  $y^x$ , i.e.,  $P(x, y' | y' \in y, y' \notin y^x) > P(x, y^x)$ . The second case is that the probability of the labeling category  $y^x$  of the current image  $x$  is lower than a hyperparameter threshold  $T$ , i.e.,  $P(x, y^x) < T$ . By filtering the samples that satisfy these two cases, we have successfully acquired the instance-level OOD data. However, the OOD data collected by the model-specified approach is closely related to the OOD detectors. To mitigate the OOD division bias introduced by these detectors and their potential classification errors, while collecting OOD data with greater real-world adaptability and universality, thereby enhancing model reliability in practical applications, we introduce a cross-validation approach with multiple generalized OOD detectors. In implementation, we chose CLIP Radford et al. (2021) and BLIP2 Li et al. (2023b) as OOD detectors and defined the intersection of OOD data divided by the two models as OOD-Hard data, while the symmetric difference of data divided by the two models is defined as OOD-Simple data. After completing the OOD data division, we further conducted manual sampling inspections to ensure that the division strictly conformed to our definition of OOD. The experimental results show that the OOD data classified by our proposed OODBench can effectively approximate the actual OOD distribution faced by large VLMs. Users can easily collect and extend OOD data through this process for applications in different fields. Our experiments demonstrate that, despite the prevalence of these OOD data classes in real-world scenarios, current state-of-the-art VLMs still struggle with out-of-distribution inputs. Even the GPT-4o achieves only approximately 65% accuracy on OOD-H data, compared to over 90% on in-distribution samples.

**Basic-to-Advanced Progression (BAP) Metric.** Through the above process, we divide the OOD data at the instance level. To comprehensively evaluate the impact of OOD data on VLMs from different dimensions, we meticulously devised the Basic-to-Advanced Progression Metric. Specifically, as shown in Fig. 4, we examine the recognition ability, counting perception ability, and reasoning ability of VLMs by constructing existential, counting, and logical reasoning problems. The existential question is the essential reasoning task. It is usually the precursor to other questions, which require the model to make a judgment about whether a particular category of objects is present in the image. The counting problem is a further judgment about the number of objects in the image after the model has correctly answered the existential question. This stage requires the model to return a specific numerical value rather than a simple ‘yes’ or ‘no’. In the third stage, we evaluate the reasoning ability of the model by asking questions about different objects in the same image (e.g., *Is the number of [class 1] in the image greater than the number of [class 2]? Answer with ‘yes’ or ‘no’*). Logical reasoning problems build on counting problems, which require the model to be aware not only of the number of objects in each category but also of making comparisons between the number of categories. Progressive questioning helps to improve the interpretability of the model’s reasoning process. Each step of the answer is based on clear logic, which helps the user understand the thinking process of the model. Finally, we calculated  $C/N \times 100\%$  as the BAP Metric, where  $C$  is the number of questions answered correctly, and  $N$  is the total number of questions.

270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323

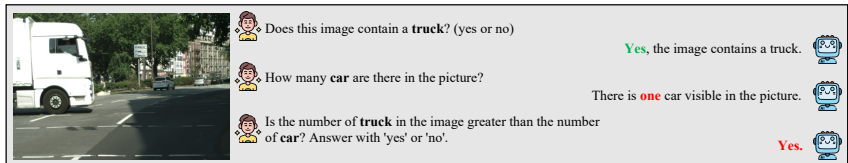


Figure 4: Basic-to-Advanced Progression Metric Example. Basic-to-Advanced Progression Metric covers three problems: existential, counting, and logical reasoning problems from top to bottom.

## 5 EXPERIMENT

### 5.1 SELECTED DATASETS

**COCO. Lin et al. (2014).** The COCO dataset is a widely used natural image dataset in computer vision. It contains over 330,000 images across 80 categories (*e.g.*, people, animals, vehicles, *etc.*). We chose its 2017 version validation set as the source of OODBench metadata. **LVIS Gupta et al. (2019).** The LVIS dataset is a large-scale natural scene dataset that focuses on instance segmentation. Especially, it focuses on the recognition of fine-grained categories and provides rich instance-level annotations. **nuScenes Caesar et al. (2020).** nuScenes is a large-scale autonomous driving dataset that provides 360-degree views of city streets, covering a wide range of urban environments and road conditions. The dataset provides camera, LiDAR, and radar sensor data. To control the benchmark’s sample number, we selected 6,019 keyframes from the front camera in the validation set as metadata for nuScenes. **Cityscapes Cordts et al. (2016).** Cityscapes is an autonomous driving dataset focused on urban street scenes for scene understanding. We chose images from the camera on the left side of the validation set as metadata for Cityscapes. In addition, we adjusted labels to ensure that their expressions conform to natural language conventions and excluded potentially ambiguous labels. For more details, please refer to Appendix A.

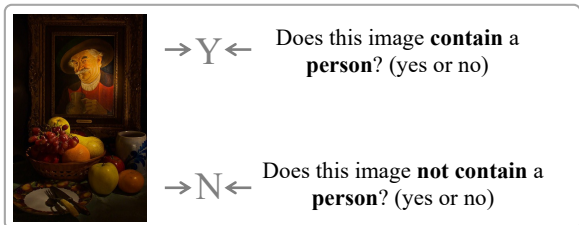


Figure 5: The **OODBench** example has two questions and one instance, with the answers alternated to avoid the model being over-scored due to biased output distribution.

### 5.2 EVALUATION SETUP

In our main results, in order to comprehensively evaluate the performance of the model, we selected several commonly used evaluation metrics, including Accuracy, F1 Score, Precision, Recall, and Matthews Correlation Coefficient (MCC). In addition, in the experiments of BAP Metric, we introduced three additional metrics to refine the evaluation further. Specifically, we define the ‘Existential Accuracy’ (E-Acc) metric, where a point is awarded only if the model correctly answers the existential question. The ‘Counting Accuracy’ (C-Acc) metric, on the other hand, requires the model to obtain a point only when it correctly answers the number of specified categories in the image. Finally, we award a point for ‘Logical Accuracy’ (L-Acc) when the model correctly answers both counting and logical questions in the test sample. Regarding the OOD detectors, we conducted systematic ablation experiments to select the type and number of generalized OOD detectors, with detailed analysis of their impact on performance in Appendix E.

### 5.3 MAIN RESULTS ANALYSIS

Through our meticulously designed OOD data collection process, we effectively divide the instance-level OOD data. We refer to the intersection of OOD data distributions obtained by dividing through multiple OOD detectors as OOD-H and the symmetric difference is denoted as OOD-S. Based on this, we obtain ID data by randomly pairing images and categories and rejecting image-category pairs belonging to OOD data. As shown in Fig. 5, we present two questions for each instance respectively (*Does this image contain a [class]? (yes or no)* and *Does this image not contain a [class]? (yes or no)*) and assign the opposite label to each question. This strategy effectively balances the label distributions of different instance data. It avoids the problem that VLMs tend to answer ei-

Model	Image Encoder	Language Model	Data Type	OODBench Performance				
				Accuracy(%)	F1(%)	Precision(%)	Recall(%)	MCC(%)
Random Chance	-	-	ID/OOD-S/OOD-H	50.00	50.00	50.00	50.00	0.00
<b>Open-source Models</b>								
LLaVA-NeXT-8B	CLIP-L-14	Llama-3-8B-Instruct	ID	84.79	84.72	85.15	84.28	69.59
			OOD-S	64.54	56.16	73.52	45.44	31.46
			OOD-H	63.22	52.99	73.40	41.46	29.36
			ID <sup>CoT</sup>	64.96	70.74	60.72	84.72	32.57
			OOD-S <sup>CoT</sup>	59.66	55.49	61.88	50.29	19.66
			OOD-H <sup>CoT</sup>	64.40	57.41	71.44	47.99	30.49
DeepSeek-VL-7B-Chat	SigLIP-L & SAM-B	DeepSeek-LLM-7B	ID	89.14	89.02	90.07	87.98	78.31
			OOD-S	64.36	58.19	70.38	49.60	30.07
			OOD-H	61.14	52.82	67.21	43.50	23.81
			ID <sup>CoT</sup>	85.14	84.59	87.82	81.60	70.46
			OOD-S <sup>CoT</sup>	63.15	56.40	69.05	47.67	27.66
			OOD-H <sup>CoT</sup>	57.44	48.27	61.53	39.72	15.92
DeepSeek-VL2-Small	SigLIP-SO400M	DeepSeekMoE LLM	ID	84.81	83.07	74.51	93.84	71.15
			OOD-S	63.03	49.96	36.91	77.28	30.56
			OOD-H	60.68	43.58	30.38	77.10	26.85
			ID <sup>CoT</sup>	82.95	79.50	66.13	99.64	69.96
			OOD-S <sup>CoT</sup>	59.81	40.17	26.98	78.58	26.02
			OOD-H <sup>CoT</sup>	56.04	29.82	18.68	73.92	18.19
InternVL2-8B	InternViT-300M-448px	InternLM2.5-7b-chat	ID	90.93	90.75	92.51	89.06	81.91
			OOD-S	69.56	67.58	72.27	63.46	39.40
			OOD-H	59.05	55.87	60.58	51.85	18.30
			ID <sup>CoT</sup>	93.18	92.75	98.95	87.29	86.97
			OOD-S <sup>CoT</sup>	66.90	59.08	77.36	47.79	36.58
			OOD-H <sup>CoT</sup>	69.26	61.11	83.17	48.30	42.44
InternVL2.5-8B	InternViT-300M-448px-V2.5	InternLM2.5-7b-chat	ID	91.66	91.79	90.37	93.26	83.37
			OOD-S	72.37	71.51	73.82	69.34	44.83
			OOD-H	58.49	57.91	58.73	57.11	16.98
			ID <sup>CoT</sup>	90.49	90.56	89.89	91.24	80.99
			OOD-S <sup>CoT</sup>	69.87	65.93	75.84	58.30	40.84
			OOD-H <sup>CoT</sup>	67.09	63.57	71.19	57.42	34.84
Llama-3.2-11B-Vision	ViT-H-14	Llama3.1-8B	ID	92.73	92.87	91.16	94.65	85.53
			OOD-S	68.65	68.42	68.94	67.91	37.31
			OOD-H	66.26	67.55	65.07	70.22	32.63
			ID <sup>CoT</sup>	74.49	76.31	71.23	82.18	49.57
			OOD-S <sup>CoT</sup>	59.80	58.26	60.59	56.10	19.66
			OOD-H <sup>CoT</sup>	61.14	62.20	60.55	63.94	22.31
Qwen2-VL-7B-Instruct	CLIP-L-14	Qwen2-7B	ID	90.67	90.92	88.52	93.46	81.47
			OOD-S	73.01	70.79	77.14	65.40	46.56
			OOD-H	66.77	63.19	70.82	57.05	34.20
			ID <sup>CoT</sup>	90.65	90.92	88.32	93.69	81.45
			OOD-S <sup>CoT</sup>	71.61	69.27	75.50	63.99	43.74
			OOD-H <sup>CoT</sup>	65.23	62.10	68.24	56.97	30.87
Qwen2.5-VL-7B-Instruct	ViT(SwiGLU, RMSNorm)	Qwen2.5 LLM	ID	90.35	90.14	88.31	92.06	80.76
			OOD-S	64.65	56.67	46.25	73.17	31.50
			OOD-H	68.70	61.31	49.61	80.24	40.46
			ID <sup>CoT</sup>	93.29	92.81	86.60	99.97	87.36
			OOD-S <sup>CoT</sup>	68.27	58.75	45.18	83.94	41.19
			OOD-H <sup>CoT</sup>	68.93	59.31	45.29	85.92	42.97
<b>Closed-source Models</b>								
Gemini	-	-	ID	91.12	91.29	89.52	93.14	82.30
			OOD-S	71.78	71.87	71.64	72.10	43.56
			OOD-H	63.54	64.38	62.93	65.89	27.11
			ID <sup>CoT</sup>	94.37	94.30	95.56	93.07	88.77
			OOD-S <sup>CoT</sup>	69.36	65.34	75.22	57.74	39.82
			OOD-H <sup>CoT</sup>	73.45	71.02	78.16	65.07	47.56
GPT-4o	-	GPT-4	ID	91.95	92.24	89.04	95.68	84.14
			OOD-S	73.37	74.13	72.05	76.34	46.81
			OOD-H	65.13	67.06	63.54	71.00	30.47
			ID <sup>CoT</sup>	78.13	80.94	71.72	92.87	58.87
			OOD-S <sup>CoT</sup>	62.44	61.49	63.08	59.98	24.91
			OOD-H <sup>CoT</sup>	63.62	64.65	62.88	66.52	27.29

Table 1: **Performance on OODBench.** We report the performance of the **10** leading VLMs on OODBench. All models perform significantly lower on OOD-H than on ID. The performance of the latest models, such as LLaVA-NeXT, Llama3.2-Vision, InternVL2.5, and Qwen2-VL, lags behind the ID data by about 20% to 30% on OOD-H data. Even the best closed-source model, GPT-4o, still performs 26% lower on OOD-H than ID.

ther ‘yes’ or ‘no’ under training data bias, leading to over- or under-performance. Given that the problem format of this task superficially resembles the prompt templates used in some hallucination detection literature, we systematically compare hallucination data with OODBench in Appendix D across three dimensions—definition, generation mechanism, and experimental results—to clarify their fundamental differences.

Tab. 1 shows the performance of the current state-of-the-art LLaVA-NeXT Li et al. (2024b), DeepSeek-VL Lu et al. (2024), InternVL2 Chen et al. (2024c), InternVL2.5 Chen et al. (2024a), Llama-3.2-Vision Dubey et al. (2024), Qwen2-VL Wang et al. (2024a), Gemini Team et al. (2024) and GPT-4o Achiam et al. (2023); Hurst et al. (2024) models on **OODBench**. The results show that the accuracy of most VLMs on OOD-H data is about 20% to 30% lower than their performance on ID data. In addition, all models show a decreasing trend in accuracy on ID, OOD-S, and OOD-H data. Essentially, OOD-S data is data that is only identified as OOD data by one OOD detector and not identified as OOD data by another OOD detector. In contrast, OOD-H data is identified as OOD data by both OOD detectors. These results show decreasing accuracy on ID, OOD-S, and OOD-H data, suggesting that using multiple well-pretrained VLMs as OOD detectors is more effective for

Threshold $T$	Data Type	nuScenes Performance					
		Num	Accuracy	F1	Precision	Recall	MCC
0.05	OOD-S	25798	66.30	63.34	69.43	58.24	33.02
	OOD-H	9882	58.86	60.28	58.28	62.42	17.77
0.01	OOD-S	23546	62.65	55.69	68.44	46.95	26.64
	OOD-H	5934	60.79	59.08	61.76	56.62	21.65
0.005	OOD-S	23644	60.93	53.31	66.24	44.60	23.13
	OOD-H	4274	59.20	57.42	60.03	55.03	18.45

Table 2: An ablation study of hyperparameter  $T$  using the LLama-3.2-Vision Dubey et al. (2024).

collecting OOD data aligned with real-world scenarios than a single detector. Moreover, this process requires no extra human effort, minimizing labor costs. Notably, VLMs (e.g., LLaVA-NeXT, DeepSeek-VL, InternVL2/2.5, Qwen2-VL) show poor recall on OOD-H data, with LLaVA-NeXT and DeepSeek-VL even falling below 50% of random chance. By contrast, their recall on ID data remains around 90%, yielding a 26–43% drop on OOD-H.

This experimental result suggests that instances in OOD data are more likely to be missed compared to instances in ID data. In safety-critical contexts, low recall is unacceptable. For example, an autonomous driving system must detect all hazards—pedestrians, vehicles, or traffic signs, since missing any can greatly increase accident risk. Appendix F details VLM performance on natural and driving datasets, illustrating how OOD data affect model behavior across scenarios.

### Does OODBench Accurately Reflect the OOD Distributions of Large VLMs?

To assess whether OODBench accurately captures the out-of-distribution (OOD) characteristics encountered by VLMs, we conducted an overlap-rate analysis. Specifically, we strictly follow the definition of OOD data division in the literature (Averly & Chao, 2023; Yang et al., 2024b) by labelling samples that the model misclassifies as OOD data. In an ideal setting, using each VLM itself as an OOD detector should most accurately reveal its OOD distribution. Accordingly, we employed Qwen2-VL, Gemini, and GPT-4o as model-specific detectors to flag their OOD samples on COCO as “ground truth.” We then computed the overlap between these model-specific OOD sets and the OOD samples identified by the generalized detector in OODBench (Fig. 6).

We find that 80.57% of OODBench’s samples are also marked by Qwen2-VL; of the remaining samples, an additional 15.06% are recognized by Gemini. This indicates that samples missed by one VLM may still be detected as OOD by another. Crucially, as we aggregate detections across models, coverage rises sharply: with three or more VLMs, over 97.97% of OODBench samples are deemed OOD by at least one model. These results show that OODBench’s partitions approximate the actual OOD distributions faced by diverse VLMs, offering a representative and consistent benchmark for evaluating their generalization performance.

**The selection of the threshold  $T$ .** In the process of collecting OOD data, we define that when the probability of an instance category is lower than the threshold  $T$ , it is considered a detection failure, and thus, the instance category is considered OOD data. In order to investigate the effect of different thresholds  $T$  on the division of OOD data, we conducted  $T$  ablation experiments. Considering that Llama Touvron et al. (2023) is currently the most popular Large Language Model, we chose Llama-3.2-Vision Dubey et al. (2024) as the test model. When  $T$  is larger than 0.05, there is too much OOD data division, so we set  $T$  to 0.05, 0.01, and 0.005 to control the amount of OOD data. The experimental results are shown in Tab. 2: when  $T = 0.05$ , the accuracy difference between OOD-S and OOD-H is 7.44; when  $T = 0.01$ , the difference decreases to 1.86; and when  $T = 0.005$ , the difference further decreases to 1.73. As the value of  $T$  decreases, the boundary of the distributions between OOD-S and OOD-H becomes more blurred. In terms of the number of OOD data, the decrease in the value of  $T$  leads to a decrease in the number of OOD-S from 25,798 to about 23,500 and a significant decrease in the number of OOD-H from 9,882 to 4,274. To ensure that the model divides the OOD data with a high degree of differentiation, we finally set  $T$  to 0.05.

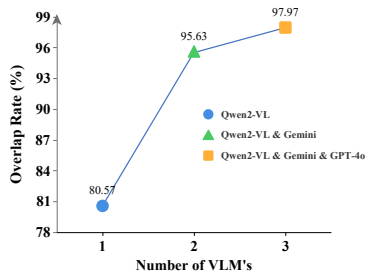


Figure 6: One-shot overlap rate between data of OODBench and model-specific VLMs.

#### 5.4 DOES CHAIN-OF-THOUGHT REASONING WORK ON OOD DATA?

Chain-of-Thought (CoT) prompting aims to enhance a model’s reasoning ability by guiding it to generate intermediate steps rather than directly outputting an answer, thereby improving reasoning accuracy. However, when confronted with OOD data, models often fail to reason accurately using their existing knowledge, as such data fall outside the training distribution. Can CoT prompting still help in these cases? Specifically, does explicitly expanding the inference process enable models to handle unfamiliar data and uncover latent patterns? To investigate this, we designed a CoT prompt to incorporate step-by-step reasoning into the OOD processing flow, evaluating whether CoT can improve VLMs’ reasoning on OOD data. Our prompt setup mirrors that of the main experiment, with an added requirement for stepwise analysis: *e.g.*, *Does this image contain [class]? Yes or No? Let’s break down the information step by step.* and *Does this image not contain [class]? Yes or No? Let’s break down the information step by step.*

Table 1 demonstrates the performance of different VLMs after adding CoT prompts. Gemini Team et al. (2024), InternVL2 Chen et al. (2024c), and InternVL2.5 Chen et al. (2024a) exhibit similar notable improvements: on challenging OOD-H samples, accuracy increases by about 10% over the baseline; ID accuracy remains stable or rises slightly; OOD-S accuracy declines by 2–3% on average. Precision gains 15.23%, 22.59%, and 12.46% for Gemini, InternVL2, and InternVL2.5, respectively, while recall is unchanged or marginally lower. In contrast, LLaVA-NeXT Li et al. (2024b), Qwen2-VL Wang et al. (2024a), and Qwen2.5-VL Bai et al. (2025a) show negligible OOD-H gains: Qwen2-VL remains stable on ID and OOD-S, whereas LLaVA-NeXT’s ID accuracy drops sharply from 84.79% to 64.96%. Moreover, GPT-4o Achiam et al. (2023); Hurst et al. (2024), DeepSeek-VL Lu et al. (2024), DeepSeek-VL2 Wu et al. (2024a), and Llama-3.2-Vision Dubey et al. (2024) all suffer overall performance degradation after CoT prompting: GPT-4o and Llama-3.2-Vision ID accuracy falls by 13.82% and 18.27%, respectively, and OOD-H accuracy declines by 1.5–5%. Hence, CoT yields negative gains for these four models.

#### 5.5 BAP EVALUATION OF ID AND OOD DATA DIFFERENCES

To comprehensively assess VLM performance on OOD data across multiple dimensions, we introduce the Basic-to-Advanced Progression (BAP) metric, designed to evaluate a model’s capabilities in recognition, counting, and reasoning. Specifically, we construct three metrics: Existential Accuracy (E-Acc), Counting Accuracy (C-Acc), and Logical Accuracy (L-Acc), employing a progressive questioning framework to assess existential, counting, and logical challenges, respectively. For BAP evaluation, we select images containing at least two distinct labels from the ID, OOD-S, and OOD-H datasets. Each image is paired with two different category labels (without duplication), resulting in test samples with paired labels from different categories. The quantity labels for counting tasks are obtained by counting the number of target object instances in the source annotations.

As task difficulty increases from E-Acc to C-Acc to L-Acc, all models exhibit a consistent decline in performance. This trend is further exacerbated as the dataset distribution shifts from ID to OOD-S and OOD-H, reflecting greater divergence from the VLMs’ training distribution and leading to varying degrees of performance degradation across the three metrics. Most models achieve robust results on ID data, particularly on E-Acc and C-Acc. However, performance declines as the data transitions to OOD-S and especially to OOD-H, with the most pronounced drop observed on L-Acc. Complex OOD data (*e.g.*, OOD-H) results in a significant decrease in logical reasoning

Model	Data Type	OOdBench BAP Performance		
		E-Acc(%)	C-Acc(%)	L-Acc(%)
LLaVA-NeXT-8B	ID	81.39	38.25	21.70
	OOD-S	66.61	33.75	21.48
	OOD-H	44.22	21.71	13.45
DeepSeek-VL-7B-Chat	ID	84.66	38.67	24.40
	OOD-S	69.73	26.72	17.50
	OOD-H	46.25	22.34	14.46
DeepSeek-VL2-Small	ID	70.77	24.32	15.04
	OOD-S	63.54	30.75	17.96
	OOD-H	37.74	20.32	12.09
InternVL2-8B	ID	87.57	51.07	33.35
	OOD-S	73.20	40.43	26.62
	OOD-H	56.72	40.20	28.43
InternVL2.5-8B	ID	91.34	45.63	32.42
	OOD-S	76.89	38.68	29.03
	OOD-H	61.62	35.40	26.33
Llama-3.2-11B-Vision-Instruct	ID	94.57	47.27	28.44
	OOD-S	77.99	38.95	22.49
	OOD-H	74.33	31.58	18.56
Qwen2-VL-7B-Instruct	ID	91.73	46.13	30.01
	OOD-S	75.31	42.36	30.72
	OOD-H	59.77	33.44	23.88
Qwen2.5-VL-7B-Instruct	ID	86.34	37.38	32.75
	OOD-S	73.39	39.25	28.13
	OOD-H	56.52	33.67	24.39
Gemini	ID	91.43	58.36	35.56
	OOD-S	77.84	52.20	34.12
	OOD-H	72.28	42.82	27.86
GPT-4o	ID	94.84	48.26	30.83
	OOD-S	81.26	41.43	27.93
	OOD-H	76.38	36.00	24.36

Table 3: Performance of 10 leading VLMs on **OOdBench** for Basic-to-Advanced Progress.

486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539



Figure 7: Typical failure cases of GPT-4o on OODBench. Errors primarily fall into two categories: non-main semantic objects and semantic variants.

accuracy across models. E-Acc mainly tests recognition without complex reasoning. On OOD-H data, GPT-4o Achiam et al. (2023); Hurst et al. (2024) shows the best recognition, correctly identifying category existence in images. In contrast, C-Acc evaluates whether models can count distinct categories, requiring both recognition and precise counting. Gemini Team et al. (2024) leads with 42.82%, underscoring its relative strength in counting. L-Acc evaluates logical reasoning by comparing category counts (e.g., *Is A greater than B?*). This task couples counting with higher-level reasoning, and all models show a sharp drop in performance, highlighting the challenge of logical reasoning and quantitative relations in OOD settings.

## 6 ANALYSIS OF ERROR CASES

As shown in Figure 7, we conducted a qualitative analysis of the failure patterns of the representative GPT-4o model on OODBench. Errors primarily cluster around two semantic OOD scenarios: non-main objects and semantic variants.

**(1) Failures caused by non-primary semantic objects.** When a clear main semantic object exists in an image (e.g., "people" in the first and fourth examples of the first row in Figure 7), the model's image-caption autoregressive learning during pretraining tends to reinforce the alignment between the image and the main semantic object, while learning about non-main semantic objects is insufficient or even absent. Therefore, when we construct scenarios in OODBench where the main semantic object is present, but the question involves a non-main semantic object, the model has established almost no robust image-text associations for these semantics during training, leading to systematic misjudgments.

**(2) Failures due to semantic variants.** Another common error category stems from semantic variants. For example, the "skateboard made of cake" in the first sample of the second row in Figure 7. During training, the model encounters numerous "skateboards" typically possessing the standard appearance found in natural distributions. The joint distribution of images and text forms the model's default semantic template for the concept of "skateboard." When we replace the target object with variants featuring different materials, shapes, or construction methods (such as a "cake-shaped skateboard"), this sample deviates from the training distribution within the model's image-text semantic space. Consequently, the model misclassifies or ignores it.

These qualitative analyses indicate that model failures under semantic shifts are not random errors but stem from a structural mismatch between the joint distribution of text and images and pre-trained knowledge, consistent with our quantitative findings.

## 7 CONCLUSION

In this paper, we introduce **OODBench** to evaluate the performance of Vision Language Models (VLMs) in the face of out-of-distribution data. OODBench is able to more accurately reflect the coping ability of VLMs when they encounter data distribution shifts in real-world scenarios. In addition, we present the Basic-to-Advanced Progression Metric for evaluating the recognition, count-awareness, and reasoning abilities of VLMs. Our automated OOD division benchmarking process adapts efficiently to new data sources and does not require significant human effort, enabling future evaluations of the ability of vision language models to be out-of-distribution of different domains.

## REFERENCES

- 540  
541  
542 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Ale-  
543 man, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical  
544 report. *arXiv preprint arXiv:2303.08774*, 2023.
- 545 Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel  
546 Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language  
547 model for few-shot learning. *NeurIPS*, 35:23716–23736, 2022.
- 548 Reza Averly and Wei-Lun Chao. Unified out-of-distribution detection: A model-specific perspective.  
549 In *ICCV*, pp. 1453–1463, 2023.
- 550 Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani  
551 Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-  
552 source framework for training large autoregressive vision-language models. *arXiv preprint*  
553 *arXiv:2308.01390*, 2023.
- 554 Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang  
555 Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, local-  
556 ization, text reading, and beyond, 2023. URL <https://arxiv.org/abs/2308.12966>.
- 557 Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang,  
558 Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*,  
559 2025a.
- 560 Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang,  
561 Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*,  
562 2025b.
- 563 Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng  
564 Shou. Hallucination of multimodal large language models: A survey. *arXiv preprint*  
565 *arXiv:2404.18930*, 2024.
- 566 Baidu-ERNIE-Team. Ernie 4.5 technical report. [https://ernie.baidu.com/blog/  
567 publication/ERNIE\\_Technical\\_Report.pdf](https://ernie.baidu.com/blog/publication/ERNIE_Technical_Report.pdf), 2025.
- 568 Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush  
569 Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for  
570 autonomous driving. In *CVPR*, pp. 11621–11631, 2020.
- 571 Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman  
572 Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large  
573 language model as a unified interface for vision-language multi-task learning. *arXiv preprint*  
574 *arXiv:2310.09478*, 2023a.
- 575 Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Car-  
576 los Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a  
577 multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023b.
- 578 Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shen-  
579 glong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source  
580 multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*,  
581 2024a.
- 582 Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shen-  
583 glong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source  
584 multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*,  
585 2024b.
- 586 Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong,  
587 Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to com-  
588 mercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024c.
- 589  
590  
591  
592  
593

- 594 Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong,  
595 Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to  
596 commercial multimodal models with open-source suites. *SCIS*, 67(12):220101, 2024d.  
597
- 598 Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong  
599 Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning  
600 for generic visual-linguistic tasks. In *CVPR*, pp. 24185–24198, 2024e.
- 601 Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo  
602 Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic  
603 urban scene understanding. In *CVPR*, 2016.  
604
- 605 Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li,  
606 Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models  
607 with instruction tuning. *NeurIPS*, 36:49250–49267, 2023.
- 608 Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao  
609 Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv  
610 preprint arXiv:2505.14683*, 2025.  
611
- 612 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha  
613 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.  
614 *arXiv preprint arXiv:2407.21783*, 2024.
- 615 Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang  
616 Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for  
617 entangled language hallucination and visual illusion in large vision-language models. In *Pro-  
618 ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–  
619 14385, 2024.  
620
- 621 Joris Guérin, Kevin Delmas, Raul Ferreira, and Jérémie Guiochet. Out-of-distribution detection is  
622 not all you need. In *AAAI*, volume 37, pp. 14829–14837, 2023.
- 623 Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance seg-  
624 mentation. In *CVPR*, pp. 5356–5364, 2019.  
625
- 626 Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution  
627 examples in neural networks. *ICLR*, 2017.
- 628 Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi,  
629 Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings.  
630 *ICML*, 2022.  
631
- 632 Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning  
633 and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer  
634 vision and pattern recognition*, pp. 6700–6709, 2019.
- 635 Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Os-  
636 trow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint  
637 arXiv:2410.21276*, 2024.  
638
- 639 Paul F Jaeger, Carsten T Lüth, Lukas Klein, and Till J Bungert. A call to reflect on evaluation  
640 practices for failure detection in image classification. *ICLR*, 2023.
- 641 Bo Li, Kaichen Zhang, Hao Zhang, Dong Guo, Renrui Zhang, Feng Li, Yuanhan Zhang,  
642 Ziwei Liu, and Chunyuan Li. Llava-next: Stronger llms supercharge multimodal ca-  
643 pabilities in the wild, May 2024a. URL [https://llava-vl.github.io/blog/  
644 2024-05-10-llava-next-stronger-llms/](https://llava-vl.github.io/blog/2024-05-10-llava-next-stronger-llms/).  
645
- 646 Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Bench-  
647 marking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*,  
2023a.

- 648 Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li.  
649 Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv*  
650 *preprint arXiv:2407.07895*, 2024b.
- 651 Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image  
652 pre-training with frozen image encoders and large language models. In *ICML*, pp. 19730–19742.  
653 PMLR, 2023b.
- 654 Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and  
655 Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal  
656 models. In *CVPR*, pp. 26763–26773, 2024c.
- 657 Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr  
658 Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pp.  
659 740–755. Springer, 2014.
- 660 Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object  
661 detection. In *ICCV*, pp. 2980–2988, 2017.
- 662 Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift  
663 with black box predictors. In *ICML*, pp. 3122–3130. PMLR, 2018.
- 664 Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou,  
665 Rongjun Li, and Wei Peng. A survey on hallucination in large vision-language models. *arXiv*  
666 *preprint arXiv:2402.00253*, 2024a.
- 667 Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*,  
668 36:34892–34916, 2023.
- 669 Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction  
670 tuning. In *CVPR*, pp. 26296–26306, 2024b.
- 671 Jiashuo Liu, Zheyang Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards  
672 out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.
- 673 Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan,  
674 Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around  
675 player? In *European Conference on Computer Vision*, pp. 216–233. Springer, 2025.
- 676 Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren,  
677 Zhuoshu Li, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. Deepseek-  
678 vl: Towards real-world vision-language understanding, 2024.
- 679 Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov,  
680 Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning  
681 robust visual features without supervision. *Transactions on Machine Learning Research Journal*,  
682 pp. 1–31, 2024.
- 683 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,  
684 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual  
685 models from natural language supervision. In *ICML*, pp. 8748–8763. PMLR, 2021.
- 686 Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Ro-  
687 hban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out-of-  
688 distribution detection: Solutions and future challenges. *arXiv preprint arXiv:2110.14051*, 2021.
- 689 Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi  
690 Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An  
691 open large-scale dataset for training next generation image-text models. *Advances in Neural*  
692 *Information Processing Systems*, 35:25278–25294, 2022.
- 693 Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-  
694 likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

- 702 Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors  
703 with online hard example mining. In *CVPR*, pp. 761–769, 2016.  
704
- 705 Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer,  
706 Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal under-  
707 standing across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.
- 708 Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej,  
709 Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical  
710 report. *arXiv preprint arXiv:2503.19786*, 2025.  
711
- 712 Shengbang Tong, Erik Jones, and Jacob Steinhardt. Mass-producing failures of multimodal systems  
713 with language models. *NeurIPS*, 36:29292–29322, 2023.
- 714 Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide  
715 shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, pp. 9568–9578, 2024.  
716
- 717 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée  
718 Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and  
719 efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.  
720
- 721 Ultralytics. Yolov8, 2023. URL <https://github.com/ultralytics/ultralytics>.
- 722 Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-  
723 logit matching. In *CVPR*, pp. 4921–4930, 2022.  
724
- 725 Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu,  
726 Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng  
727 Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model’s  
728 perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.
- 729 Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu,  
730 Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the  
731 world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024b.  
732
- 733 Weihang Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang,  
734 Lei Zhao, Song XiXuan, et al. Cogvlm: Visual expert for pretrained language models. *NeurIPS*,  
735 37:121475–121499, 2024c.
- 736 Weizhi Wang, Yu Tian, Linjie Yang, Heng Wang, and Xifeng Yan. Open-qwen2vl: Compute-  
737 efficient pre-training of fully-open multimodal llms on academic resources. *arXiv preprint*  
738 *arXiv:2504.00595*, 2025.  
739
- 740 Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan  
741 Zhang, Yueze Wang, Zhen Li, Qiyang Yu, et al. Emu3: Next-token prediction is all you need.  
742 *arXiv preprint arXiv:2409.18869*, 2024d.
- 743 Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvisé-Rebuffi, Ira Ktena, Krishnamurthy  
744 Dvijotham, and Taylan Cemgil. A fine-grained analysis on distribution shift. *arXiv preprint*  
745 *arXiv:2110.11328*, 2021.  
746
- 747 Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang  
748 Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language  
749 models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024a.
- 750 Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang  
751 Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language  
752 models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024b.  
753
- 754 Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong  
755 Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, pp. 18134–  
18144, 2022.

- 756 Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj  
757 Prabhu, Yutong Dai, Michael S Ryoo, et al. Blip-3: A family of open large multimodal models.  
758 In *ICCV*, pp. 6124–6135, 2025.
- 759  
760 Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection:  
761 A survey. *IJCV*, 132(12):5635–5662, 2024a.
- 762 William Yang, Byron Zhang, and Olga Russakovsky. Imagenet-ood: Deciphering modern out-of-  
763 distribution detection algorithms. *ICLR*, 2024b.
- 764  
765 Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li,  
766 Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint*  
767 *arXiv:2408.01800*, 2024.
- 768 Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov,  
769 and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *CVPR*, pp. 18123–  
770 18133, 2022.
- 771  
772 Yuhui Zhang, Alyssa Unell, Xiaohan Wang, Dhruva Ghosh, Yuchang Su, Ludwig Schmidt, and  
773 Serena Yeung-Levy. Why are visually-grounded language models bad at image classification? In  
774 *NeurIPS*, 2024.
- 775 Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal  
776 chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.
- 777  
778 Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigt-4: En-  
779 hancing vision-language understanding with advanced large language models. *arXiv preprint*  
780 *arXiv:2304.10592*, 2023.
- 781 Yao Zhu, Yuefeng Chen, Xiaodan Li, Rong Zhang, Hui Xue, Xiang Tian, Rongxin Jiang, Bolun  
782 Zheng, and Yaowu Chen. Rethinking out-of-distribution detection from a human-centric perspec-  
783 tive. *IJCV*, pp. 1–18, 2024.

## 784 785 8 ETHICS STATEMENT

786  
787 This work does not involve human subjects, personally identifiable information, or sensitive data.  
788 All datasets used in our experiments are publicly available and widely used in the research com-  
789 munity. We have carefully followed the licenses and terms of use of these datasets. Our proposed  
790 methodology aims to automatically construct a new benchmark and evaluate the ability of existing  
791 vision-language models (VLMs) to handle out-of-distribution (OOD) data on this benchmark. The  
792 contributions of this work are intended solely for academic research, and we strongly encourage  
793 responsible use in alignment with ethical guidelines. This work complies with all applicable ethical  
794 standards and research integrity guidelines.

## 795 796 9 REPRODUCIBILITY STATEMENT

797  
798 We have taken several measures to ensure the reproducibility of our benchmark construction and  
799 experimental results. The process of automatically collecting and constructing the benchmark is  
800 described in detail in the main paper and supplementary materials. In addition, we provide an  
801 anonymous link containing the OODBench dataset, the benchmark collection source code, and the  
802 model evaluation code to facilitate full reproduction of our results.

803  
804  
805  
806  
807  
808  
809

## Outline

This document supplements the main paper with detailed results. Below is the outline.

- Section A details the collection process of OODBench.
- Section B demonstrates through a series of statistical analyses that OODBench exhibits distinct OOD properties relative to contemporary MLLMs.
- Section C describes the differences between the OODBench and the hard samples.
- Section D describes the differences between the OODBench and the hallucinations.
- Section E provides ablation studies of the number and type of generalized OOD Detector.
- Section F reports the performance of classification metrics and BAP metrics for the OOD-Bench sub-dataset.
- Section G describes the performance various of models of different scales on OODBench.
- Section H provides a visualisation of the distribution of category logits for CLIP and BLIP2.
- Section I provides partial data and case demonstration of OODBench.
- Section J discussed the limitations of OODBench.
- Section K discussed the LLMs usage.

## A COLLECTION DETAILS

**Initial Benchmark.** Fig. 8 illustrates that we collected the initial benchmark from a publicly available dataset, which contains about 77k yes-or-no samples, where OOD-S is 51k and OOD-H is 26k. Due to the large sample size of the benchmarks, to save testing time, we randomly downsampled the OOD-Simple and OOD-Hard category data in the benchmarks with a sample size of more than 6,000 to the **OODBench** to ensure that the sample size of each category did not exceed 6,000.

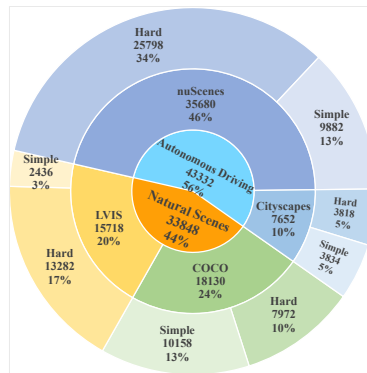


Figure 8: Distribution of categories and fields in initial benchmark.

**VLMs for Classification.** In terms of implementation details, we input image-question pairs corresponding to OOD samples into multimodal large models to generate textual responses. Subsequently, we convert the model output into binary labels by identifying the keywords ‘yes’ or ‘no’ in the generated text. This method has also been applied in the literature Zhang et al. (2024). In Tab. 1, an average accuracy of 90% is achieved on ID data, which further validates the effectiveness of the method in classification tasks. In the BAP task, we also used this method to obtain the model’s counting prediction results and compare them with the real labels to determine whether the model’s answers to counting questions were accurate.

**nuScenes and Cityscapes dataset label modifications.** Since the labels of nuScenes do not conform to natural languages, such as ‘human.pedestrian.adult’, using the original labels as text directly will cause the OOD detector to fail to understand their meaning, which will fail OOD data division. Therefore, as shown in Table 4, we manually convert these labels into a form that conforms to natural language conventions. Meanwhile, since the original label ‘movable object.debris’ of nuScenes

is not clear, we exclude it to avoid ambiguity. Similarly, we also eliminated the background, ambiguous, and semantic duplicates from the Cityscapes labels, and Table 5 shows the final labels we retained.

Original Labels	Modified Labels	Remark
human.pedestrian.adult	Adult Pedestrian	
human.pedestrian.child	Child Pedestrian	
human.pedestrian.wheelchair	Pedestrian in Wheelchair	
human.pedestrian.stroller	Pedestrian with Stroller	
human.pedestrian.personal.mobility	Pedestrian using Personal Mobility Device	
human.pedestrian.police.officer	Police Officer	
human.pedestrian.construction.worker	Construction Worker	
animal	Animal	
vehicle.car	Car	
vehicle.motorcycle	Motorcycle	
vehicle.bicycle	Bicycle	
vehicle.bus.bendy	Bendy Bus	
vehicle.bus.rigid	Rigid Bus	
vehicle.truck	Truck	
vehicle.construction	Construction Vehicle	
vehicle.emergency.ambulance	Ambulance	
vehicle.emergency.police	Police Vehicle	
vehicle.trailer	Trailer	
movable.object.barrier	Barrier	
movable.object.trafficcone	Traffic Cone	
movable.object.pushable.pullable	Pushable/Pullable Object	
movable.object.debris	Debris	The class reference is unclear and has been deleted.
static.object.bicycle.rack	Bicycle Rack	

Table 4: Original and modified labels for nuScenes.

Original Labels	Modified Labels	Remark
road	road	
sidewalk	sidewalk	
parking	parking	
rail track	rail track	
person	person	
rider	rider	
car	car	
truck	truck	
bus	bus	
on rails	-	The class reference is unclear and has been deleted.
motorcycle	motorcycle	
bicycle	bicycle	
caravan	caravan	
trailer	trailer	
building	building	
wall	wall	
fence	fence	
guard rail	guard rail	
bridge	bridge	
tunnel	tunnel	
pole	pole	
pole group	-	Pole group and pole are semantically duplicated, we choose pole for retention.
traffic sign	traffic sign	
traffic light	traffic light	
vegetation	vegetation	
terrain	-	The class reference is unclear and has been deleted.
sky	-	Background classes we exclude.
ground	-	The class reference is unclear and has been deleted.
dynamic	-	The class reference is unclear and has been deleted.
static	-	The class reference is unclear and has been deleted.

Table 5: Original and modified labels for Cityscapes.

## B STATISTICAL EVIDENCE FOR THE OOD NATURE OF OODBENCH IN MODERN MLLMs

To verify whether OODBench constitutes OOD data within existing MLLMs, we conducted systematic statistical analysis across two complementary dimensions: open-source models and closed-source models. In the open-source model section, we calculated the semantic overlap rate between the model training corpus and the COCO-train-OOD partition, which was constructed according to the same rules as OODBench. We then evaluated the consistency and robustness of this partition using methods such as exact binomial intervals, Bayesian priors, and Two One-Sided Tests (TOST). In the closed-source model section, we further establish independent chains of evidence based on both the “significance” and “magnitude consistency” of performance results. This is achieved through significance testing for performance degradation (ID  $\rightarrow$  OOD) and equivalence assessments against open-source OOD behavioral signatures. Through these complementary statistical inferences from both open-source and closed-source models, we jointly demonstrate that OODBench indeed exhibits verifiable OOD characteristics for current MLLMs, examining both the nature of data distributions and model response behaviors.

### B.1 OOD EVIDENCE FROM OPEN-SOURCE MODELS

Open-Source MLLMs refer to models that disclose their core components to varying degrees within the research community, thereby enabling independent auditability, inspectability, reproducibility, or reusability. It is crucial to emphasize that open-source is not a binary attribute but exists along a continuum of openness. As shown in Table 6, we categorize current open-source MLLMs into four primary levels based on the scope and granularity of disclosed information, where each level exhibits a hierarchical increase in model weights, architectural details, training code, training data transparency, and reproducibility.

Table 6: Current open-source MLLM models are categorized into four primary tiers based on the scope and granularity of publicly disclosed information. Each level exhibits progressively enhanced transparency and reproducibility across model weights, architectural details, training code, training data, and overall reproducibility.

Level	Name	Weights	Architecture	Training Code	Training Data	Reproducibility
0	Non-Open Source	✗	✗	✗	✗	✗
1A	Architecture Open Source	✗	✓	Partial	✗	✗
1B	Weights Open but Architecture Not Fully Open	✓	Partial	Partial	✗	✗
1C	Data Partially Open	Optional	Optional	Optional	Partial	✗
2	Weights Open	✓	✓	✓	✗ / Rough	✓ (Partial)
3	Strongly Open (Reproducible)	✓	✓	✓	✓ (Clean)	✓ (Partial)
4	Fully Open Source (Full Pipeline Open)	✓	✓	✓	✓ (Fully Open)	✓✓✓ (Strict)

To rigorously assess whether the proposed OODBench constitutes out-of-distribution (OOD) data for existing open-source MLLMs, we conducted a systematic statistical analysis. We first quantified the overlap ratio between OODBench and each model’s corresponding training corpus. Then we incorporated statistical inference procedures to estimate the population-level proportion of OOD samples across models. To enhance the representativeness and cross-model comparability of these evaluations, we adopted the COCO dataset—one of the most commonly used and widely standardized datasets in multimodal model training—as the unified reference source. This provides a consistent framework for estimating overlap ratios and determining the OOD characteristics of our constructed benchmark.

#### B.1.1 PRINCIPLES FOR SELECTING MODEL SAMPLES

We first conducted a systematic survey of current mainstream open-source MLLMs, covering diverse model families, parameter scales, and architectural paradigms. By comprehensively collecting publicly released versions, model repositories, and official documentation, we identified 28 model families corresponding to a candidate set of 93 models spanning parameter sizes from 0.5B to 424B.

**Table 7: Complete summary of collected models. Includes models ultimately included in the analysis as well as those excluded during the screening process.**

No.	Family	Years	Models	Institution	COCO Used	Openness Level	Included in Analysis
1	Flamingo Alayrac et al. (2022)	2022	Flamingo-3B, Flamingo-9B	DeepMind	✗	L3	✗
2	Qwen-VL Bai et al. (2023)	2023	Qwen-VL(7B), Qwen-VL-Chat(7B), Qwen-VL-Plus(7B), Qwen-VL-Max(7B)	Alibaba Group	✓	L3	✓
3	PaLI-X Chen et al. (2023b)	2023	PaLI-X(55B)	Google Research	✗	L1C	✗
4	LLaVA-v1 Liu et al. (2023)	2023	LLaVA-v1-Vicuna(13B), LLaVA-v1-Llama(13B), LLaVA-v1-Llama(7B)	UW-Madison	✓	L3	✓
5	InstructBLIP Dai et al. (2023)	2023	InstructBLIP (FlanT5XL)(3B), InstructBLIP (FlanT5XXL)(11B), InstructBLIP (Vicuna-7B), InstructBLIP (Vicuna-13B)	Salesforce Research	✓	L3	✓
6	MiniGPT-4 Zhu et al. (2023)	2023	MiniGPT-4 (Vicuna 13B), MiniGPT-4 (Vicuna 7B), MiniGPT-4 (LLaMA-2 Chat 7B),	KAUST	✗	L3	✗
7	MiniGPT-v2 Chen et al. (2023a)	2023	MiniGPT-v2-LLaMA2-7B, MiniGPT-v2-Vicuna-7B, MiniGPT-v2-Vicuna-13B	KAUST	✓	L3	✓
8	OpenFlamingo Awadalla et al. (2023)	2023	OpenFlamingo-3B, OpenFlamingo-3B (Instruct), OpenFlamingo-4B, OpenFlamingo-4B (Instruct), OpenFlamingo-9B	UW	✗	L1C	✗
9	InternVL Chen et al. (2024c)	2024	InternVL-C(7B), InternVL-G(7B), InternVL-Chat-13B, InternVL-Chat-19B	OpenGVLab	✓	L3	✓
10	DeepSeek-VL Lu et al. (2024)	2024	DeepSeek-VL-1.3B-base, DeepSeek-VL-1.3B-chat, DeepSeek-VL-7B-base, DeepSeek-VL-7B-chat	DeepSeek-AI	✗	L3	✗
11	LLaVA-1.5 Liu et al. (2024b)	2024	LLaVA-1.5-7B, LLaVA-1.5-13B	UW-Madison	✓	L3	✓
12	InternVL Chen et al. (2024d)	2024	InternVL-1.2(40B), InternVL-1.5(26B)	Shanghai AI Laboratory	✓	L3	✓
13	CogVLM Wang et al. (2024c)	2024	cogvlm-chat-v1.1-Vicuna-7B, cogvlm-chat-v1.1-Llama3-8B, cogvlm-base-224(7B), cogvlm-base-490(7B), cogvlm-grounding-generalist(7B)	Zhipu AI	✓	L3	✓
14	MiniCPM-V Yao et al. (2024)	2024	MiniCPM-V 1.0(2.8B), MiniCPM-V 2.0(2.8B), MiniCPM-Llama3-V 2.5(8.5B)	OpenBMB	✓	L3	✓
15	Emu3 Wang et al. (2024d)	2024	Emu3-Chat(8B), Emu3-Gen(8B), Emu3-DPO(8B)	BAAI	✗	L1C	✗
16	Monkey Li et al. (2024c)	2024	Monkey(9.8B)	HUST	✓	L3	✓
17	Qwen2-VL Wang et al. (2024b)	2024	Qwen2-VL-2B, Qwen2-VL-7B, Qwen2-VL-72B	Alibaba Group	unknown	L1C	✗
18	CogVLM Wang et al. (2024c)	2024	CogVLM2-LLaMA3(8B), GLM-4V-9B	Zhipu AI	✗	L1C	✗
19	DeepSeek-VL2 Wu et al. (2024b)	2024	DeepSeek-VL2-tiny(3B), DeepSeek-VL2-small(16B), DeepSeek-VL2(27B)	DeepSeek-AI	unknown	L1C	✗
20	LLaVA-NeXT Li et al. (2024a)	2024	LLaVA-NeXT-Interleave-0.5B, LLaVA-NeXT-Interleave-7B, LLaVA-NeXT-Interleave-14B, LLaVA-NeXT-Interleave-7B-DPO	ByteDance	✓	L1C	✗
21	Open-Qwen2-VL Wang et al. (2025)	2025	Open-Qwen2-VL(2B)	UC Santa Barbara	✓	L3	✓
22	BLIP-3 Xue et al. (2025)	2025	BLIP-3-4B-SI, BLIP-3-4B-MI, BLIP-3-14B-SI, BLIP-3-14B-MI	Salesforce AI Research	✓	L3	✓
23	BAGEL Deng et al. (2025)	2025	BAGEL-1.5B, BAGEL-7B	ByteDance Seed	✓	L3	✗
24	InternVL2 Chen et al. (2024b)	2025	InternVL2-2B, InternVL2-4B, InternVL2-8B, InternVL2-26B, InternVL2-40B, InternVL2-Llama3-76B	Shanghai AI Laboratory	✓	L3	✓
25	InternVL2.5 Chen et al. (2024b)	2025	InternVL2.5-1B, InternVL2.5-2B, InternVL2.5-4B, InternVL2.5-8B, InternVL2.5-26B, InternVL2.5-38B, InternVL2.5-78B	Shanghai AI Laboratory	✓	L3	✓
26	Gemma-3 Team et al. (2025)	2025	Gemma-3-1B, Gemma-3-4B, Gemma-3-12B, Gemma-3-27B	Google DeepMind	unknown	L1C	✗
27	ERNIE 4.5 Baidu-ERNIE-Team (2025)	2025	ERNIE-4.5-VL-28B-A3B-Thinking, ERNIE-4.5-VL-424B-A47B-Base, ERNIE-4.5-VL-424B-A47B, ERNIE-4.5-VL-28B-A3B-Base, ERNIE-4.5-VL-28B-A3B	Baidu	unknown	L1C	✗
28	Qwen2.5-VL Bai et al. (2025b)	2025	Qwen2.5-VL-3B, Qwen2.5-VL-7B, Qwen2.5-VL-72B	Alibaba Group	unknown	L1C	✗

To ensure transparency and auditability at the training data level, we restricted our selection to robust open-source models at Levels 3 and 4, meaning those that either publicly disclose their training data inventories or possess strict reproducibility. In this process, we excluded the following two categories of models that did not meet these criteria:

- Models using in-house or non-public data.
- Models with incomplete training data lists often lack precise entries, which prevents line-by-line verification.

After obtaining a set of candidate models that meet the training data transparency requirements, we further exclude models that did not use COCO as a training data source. Since our objective is to quantify the overlap ratio between OODBench-COCO and each MLLM’s training data, a model with no COCO inclusion in its training corpus would yield a trivially zero overlap ratio. This would render it incapable of providing any meaningful validation for our proposed overlap ratio analysis

workflow. Consequently, such models are unsuitable as statistical test samples. After these two rounds of screening, we ultimately obtained 50 qualifying MLLMs for subsequent overlap ratio calculations and statistical inference analysis. Table 7 provides a comprehensive summary of the collected models, including both those ultimately included in the analysis and those excluded during the screening process.

### B.1.2 SEMANTIC-LEVEL OOD PAIRING AND OVERLAP RATE ANALYSIS

We first select an image and perform semantic consistency matching for its image-category within the OOD division. Specifically, we collect all annotated versions of this image across various MLLM training corpora (*e.g.*, captions, OCR, conversations, detailed descriptions, complex reasoning, REC, REG, grounding, *etc.*). We then systematically examine each annotation to determine whether it contains semantic expressions matching the OOD category we assigned. If the category appears in any annotated version, it indicates the model has learned the corresponding image–category semantic relationship, and this image–category pair is considered ID data. Conversely, if the category is absent from all annotated versions, the image–category pair is classified as OOD data at the semantic level of learning. It is important to note that OODBench utilizes the COCO val dataset, while most MLLM training data is based on the COCO train version. These datasets have no overlap at the image level, making it impossible to calculate overlap rates using image matching rules directly. This version discrepancy may result in false-zero overlap. To address this issue, we regenerate OOD data using the COCO train image set, applying the same division criteria as OODBench. We then calculate the overlap rate with each MLLM’s training data based on this division. This approach verifies whether our categorized samples genuinely constitute an OOD distribution at the training semantic level.

### B.1.3 POPULATION-LEVEL OOD ASSESSMENT

**Settings and Symbols.** We selected  $N$  auditable open-source MLLMs as test subjects. Our constructed dataset is based on COCO-train and generates OOD data at the semantic level according to OODBench’s division strategy, denoted as “COCO-train-OOD”.

For each model  $m$ , its training overlap rate on “COCO-train-OOD” is defined as:

$$r_m = \frac{H_m}{C}, \quad (1)$$

where  $H_m$  denotes the number of samples in the model training set that achieve semantic overlap with the “COCO-train-OOD” image-category pairs, while  $C$  represents the total number of samples in “COCO-train-OOD”. Setting the threshold  $\varepsilon$  as the maximum acceptable overlap rate, the model’s OOD decision criterion (pre-registered criterion) is defined as:

$$Z_m = \mathbb{1}[\text{UCI}(r_m) < \varepsilon], \quad (2)$$

when the 95% upper confidence bound of the training overlap rate for model  $m$  does not exceed  $\varepsilon$ , we classify it as ‘treating this data as OOD’ ( $Z_m = 1$ ); if the upper confidence bound exceeds  $\varepsilon$ , it is classified as non-OOD ( $Z_m = 0$ ).

**Exact Clopper–Pearson Interval (Frequentist Lower Bound).** To estimate, at the population level, the proportion of models that satisfy the preregistered criterion, we treat the set of  $N$  evaluated open-source models as a finite population. Among them,  $k$  models satisfy  $Z_m = 1$ . We therefore compute the lower confidence bound for the binomial proportion based on  $k$  successes out of  $n = N$  trials.

The lower Clopper–Pearson bound is defined as:

$$\text{LCP} = \text{BetaInv}(0.025; k, n - k + 1). \quad (3)$$

The calculation results are as follows:

- When  $\varepsilon = 5\%$ ,  $k = 49$ ,  $n = 50$ ,  $LCP = \text{BetaInv}(0.025; 49, 2) \approx 0.894$ .
- When  $\varepsilon = 2\%$ ,  $k = 27$ ,  $n = 50$ ,  $LCP = \text{BetaInv}(0.025; 27, 24) \approx 0.393$ .

Therefore, under the criterion of "95% upper confidence bound  $\leq 5\%$  for training overlap rate," it can be asserted with 95% confidence that at least approximately 89.4% of comparable open-source MLLMs would classify this data as OOD. Even under the stricter criterion of " $\leq 2\%$ ," at least approximately 39.3% of models still meet this condition.

**Bayesian Inference with a Beta(1,1) Prior (Posterior Credible Interval).** To validate the above results from a probabilistic reasoning perspective, we employ a uniform prior "Beta(1,1)" to compute the lower bound of the 95% confidence interval under the posterior distribution:

$$L_{\text{Bayes}} = \text{BetaInv}(0.025; k + 1, n - k + 1). \quad (4)$$

The result is:

- When  $\varepsilon = 5\%$ ,  $k = 49$ ,  $n = 50$ ,  $L_{\text{Bayes}} = \text{BetaInv}(0.025; 50, 2) \approx 0.896$ .
- When  $\varepsilon = 2\%$ ,  $k = 27$ ,  $n = 50$ ,  $L_{\text{Bayes}} = \text{BetaInv}(0.025; 28, 24) \approx 0.403$ .

Therefore, from a Bayesian perspective, it can be asserted with 95% confidence that: when the criterion is "95% upper confidence bound  $\leq 5\%$ ," at least approximately 89.6% of similar models will classify the data as OOD; when the criterion is " $\leq 2\%$ ," at least approximately 40.3% of models will classify the data as OOD.

**Population-Level Conclusion.** From two complementary statistical perspectives, the Clopper–Pearson method ensures frequentist coverage, while the Bayesian Beta method provides posterior credibility. Both methods align in direction and magnitude, indicating that, **with 95% confidence, at least approximately 89–90% of comparable open-source MLLMs perceive COCO-train-OOD data as semantically OOD during training.** These open-source MLLMs encompass diverse families, parameter scales, architectures, and training strategies. Thus, this division can be robustly regarded as an "out-of-distribution relative to training semantics" sample set at the population level.

#### B.1.4 DISTRIBUTIONAL EQUIVALENCE BETWEEN COCO-TRAIN-OOD AND COCO-VAL-OOD

Let the random variable be:

$$(X, C) \in \mathcal{X} \times \mathcal{C}, \quad (5)$$

where  $X$  denotes images and  $C$  denotes discrete categorical labels. We consider separately:

- $P_{\text{train}}(X, C)$ : The joint distribution of OOD subsets divided according to OODBench rules on COCO-train.
- $P_{\text{val}}(X, C)$ : The joint distribution of OOD subsets divided according to the same rules on COCO-val.

Section B.1.3, based on the exact Clopper–Pearson interval and the Bayesian Beta(1,1) posterior interval, infers at the 95% confidence level that at least approximately 89–90% of open-source MLLMs will treat COCO-train-OOD data as OOD for training semantics. To extend this population-level OOD classification to COCO-val-OOD, it is necessary to verify further whether COCO-train-OOD and COCO-val-OOD belong to the same statistical distribution family.

**Definition of Distribution Equivalence.** We adopt the kernel mean embedding framework and utilize Maximum Mean Discrepancy (MMD) to measure the distance between two joint distributions.

For any image  $x$ , features are extracted using the self-supervised visual model DINOv2 Oquab et al. (2024).

1134  
 1135  
 1136  
 1137  
 1138  
 1139  
 1140  
 1141  
 1142  
 1143  
 1144  
 1145  
 1146  
 1147  
 1148  
 1149  
 1150  
 1151  
 1152  
 1153  
 1154  
 1155  
 1156  
 1157  
 1158  
 1159  
 1160  
 1161  
 1162  
 1163  
 1164  
 1165  
 1166  
 1167  
 1168  
 1169  
 1170  
 1171  
 1172  
 1173  
 1174  
 1175  
 1176  
 1177  
 1178  
 1179  
 1180  
 1181  
 1182  
 1183  
 1184  
 1185  
 1186  
 1187

$$\phi(X) = \mathbb{R}^d. \quad (6)$$

The reason for selecting DINOv2 is that its training process does not rely on COCO labels, making its feature space "semantically neutral" and thus avoiding the introduction of unnecessary supervised bias. Category labels  $c \in C$  are represented using one-hot vectors  $e_c \in \mathbb{R}^{\ell}$ .

We select a kernel function on the joint space  $\mathcal{Z} = \mathcal{X} \times \mathcal{C}$ :

$$k((x, c), (x', c')). \quad (7)$$

The nature of the kernel guarantees the existence of a Hilbert space (RKHS)  $\mathcal{H}$  and a mapping:

$$\exists \mathcal{H}, \Phi : \mathcal{Z} \rightarrow \mathcal{H} \quad s.t. \quad k(z, z') = \langle \Phi(z), \Phi(z') \rangle_{\mathcal{H}}. \quad (8)$$

where  $z = (x, c), z' = (x', c')$ .

For any distribution  $P$  over  $\mathcal{Z}$ , we can define its mean embedding in  $\mathcal{H}$ :

$$\mu_P := \mathbb{E}_{Z \sim P}[\Phi(Z)] \in \mathcal{H}. \quad (9)$$

We hereby order:

$$k((x, c), (x', c')) = k_v(\phi(x), \phi(x')) \cdot \mathbb{1}[c = c'], \quad (10)$$

where  $\phi(x) \in \mathbb{R}^d$  represents image features extracted via self-supervised visual models (e.g., DINOv2),  $k_v$  denotes the RBF kernel applied to visual features, and  $\mathbb{1}[\cdot]$  is the indicator function. This design constructs a block-diagonal joint kernel in the joint space  $(X, C)$ , where kernel values between different categories are zero, making them mutually orthogonal in the RKHS. Meanwhile, similarity within the same category is entirely determined by the RBF kernel of the visual features. In other words, the label variable  $C$  participates in joint modeling through the discrete structure of "intra-class/cross-class," while the visual subspace provides the intra-class geometric structure.

Based on the above joint kernel, the maximum mean displacement (MMD) between the joint distributions  $P_{\text{train}}$  and  $P_{\text{val}}$  can be expressed as:

$$\text{MMD}^2(P_{\text{train}}, P_{\text{val}}) = \|\mu_{\text{train}} - \mu_{\text{val}}\|_{\mathcal{H}}^2, \quad (11)$$

where  $\mu_{\text{train}}, \mu_{\text{val}} \in \mathcal{H}$  represent the kernel mean embeddings of  $P_{\text{train}}(X, C)$  and  $P_{\text{val}}(X, C)$  in the reproducing kernel Hilbert space  $\mathcal{H}$ , respectively. That is,  $\mu_P = \mathbb{E}_{(X, C) \sim P}[\Phi(X, C)]$ , where  $\Phi$  is the feature map induced by the joint kernel  $k$ .

**Statistical Test for Distribution Equivalence (Two-Sided Test).** We used the equivalence test (TOST) to verify the following hypothesis:

- $H_0$ :  $\text{MDD}^2(P_{\text{train}}, P_{\text{val}}) \geq \tau$ . (The difference is at least  $\tau$  and cannot be considered equivalent)
- $H_1$ :  $\text{MDD}^2(P_{\text{train}}, P_{\text{val}}) < \tau$ . (The difference is less than  $\tau$  and can be considered equivalent)

To determine the value of  $\tau$ , we construct a 'homogeneous baseline' within COCO-train-OOD as follows:

- Randomly partition COCO-train-OOD into two mutually exclusive subsets:

$$(D_{\text{train}}^{(1)}, D_{\text{train}}^{(2)}).$$

- Calculate the  $\text{MMD}^2$  between them:

$$\text{MMD}^2(D_{\text{train}}^{(1)}, D_{\text{train}}^{(2)}).$$

- Repeat multiple times to obtain a set of  $\text{MMD}^2$  values within the same distribution.
- Take the 95th percentile of this distribution as  $\tau$ :

$$\tau = \text{Quantile}_{0.95}\{\text{MMD}^2(D_{\text{train}}^{(1)}, D_{\text{train}}^{(2)})\}.$$

$\tau$  denotes the upper bound of the finite-sample fluctuations of  $\text{MMD}^2$  observed between two independent samples originating from the same underlying distribution. Consequently, if the  $\text{MMD}^2$  between COCO-train-OOD and COCO-val-OOD falls below  $\tau$ , the two partitions can be statistically regarded as independent realizations of a common source distribution, *i.e.*, as distributionally isomorphic.

To further quantify the uncertainty in  $\text{MMD}^2$  estimation, we employ a permutation test to construct a reference distribution for its null hypothesis. Specifically, we merge all samples from COCO-train-OOD and COCO-val-OOD into a single dataset, randomly permute their group labels (train/val), and recalculate  $\text{MMD}^2$  based on the permuted grouping structure. Repeating this process  $B$  times ( $B=500$  in this experiment) yields the empirical permutation distribution of  $\text{MMD}^2$  under the null hypothesis  $H_0$  (equivalence of the two distributions):

$$\mathcal{D}_{\text{perm}} = \{\text{MMD}_{\text{perm}}^2(b)\}_{(b=1)}^B. \quad (12)$$

Thus, we construct an upper bound for the  $\text{MMD}^2$  by taking the  $1 - \alpha$ -quantile of this distribution:

$$\text{UCI}_{(1-\alpha)} = \text{Quantile}_{(1-\alpha)}(\mathcal{D}_{\text{perm}}). \quad (13)$$

Under the Two One-Sided Tests (TOST) procedure, if the following condition is satisfied:

$$\text{UCI}_{0.95}(\text{MMD}^2(P_{\text{train}}, P_{\text{val}})) < \tau, \quad (14)$$

then it can be asserted:

$$P_{\text{train}}^{\text{OOD}} \approx P_{\text{val}}^{\text{OOD}}, \quad (15)$$

then, at significance level  $\alpha$ , the null hypothesis  $H_0$  that the difference is at least  $\tau$  can be rejected, and the equivalence hypothesis  $H_1$  can be accepted, indicating that the two joint distributions are statistically equivalent within the prescribed tolerance.

**Observation Conclusion:** Based on the homogenous baseline constructed from COCO-train-OOD internal partition, the upper bound of  $\text{MMD}^2$ 's 95% co-distribution fluctuation is determined to be:

$$\tau = 4.47 \times 10^{-4}.$$

After estimating the joint distribution of COCO-train-OOD and COCO-val-OOD, we further obtained the 95% confidence interval for  $\text{MMD}^2$  via bootstrap sampling:

$$\text{CI}_{95}(\text{MMD}^2) = [1.84 \times 10^{-4}, 4.21 \times 10^{-4}],$$

Its upper confidence bound is significantly lower than the homogeneity threshold  $\tau$ .

Simultaneously, the TOST equivalence test constructed based on the empirical null distribution from  $B=500$  permutations also indicates that the 95% upper confidence bound of  $\text{MMD}^2$  satisfies:

$$\text{UCI}_{0.95}(\text{MMD}^2(P_{\text{train}}, P_{\text{val}})) < \tau,$$

At the significance level  $\alpha = 0.05$ , we reject the null hypothesis  $H_0$ —that the difference is at least  $\tau$ —and accept the equivalence hypothesis  $H_1$ . This confirms that COCO-train-OOD and COCO-val-OOD are statistically distributionally equivalent within the prescribed tolerance. Since a model’s OOD response is fundamentally determined by the degree to which the test data deviates from its training semantic distribution, and train-OOD has already been shown in the previous section to constitute semantic OOD for open-source MLLMs, the statistical indistinguishability between val-OOD and train-OOD indicates that both deviate from the training distribution to the same order of magnitude. Consequently, COCO-val-OOD should exhibit the same OOD behavior for open-source MLLMs and can thus be regarded as semantic OOD data equivalent to train-OOD.

## B.2 OOD EVIDENCE FROM CLOSED-SOURCE MODELS

### Problem Setup and Assumptions. Let

- $D_{\text{ID}}$ : Conventional ID test set.
- $D_{\text{OOD}}$ : A semantic OOD test set constructed according to OODBench semantic rules (non-main semantic objects, semantic variants, and anomalous image–category pairings), with its image subset sourced from COCO-val.

In the open-source model section, we have established the following two key premises:

#### 1. Training Out-of-Distribution Properties.

By calculating the semantic overlap rate between the training corpora of open-source MLLMs and COCO-train-OOD, and combining precise binomial intervals with Bayesian credible intervals using uniform priors, we obtain the following: At the 95% confidence level, at least 90% of open-source models exhibit an overlap rate with COCO-train-OOD not exceeding 5%. This result indicates that, **from the perspective of training distribution, COCO-train-OOD can be robustly regarded as OOD data for the collection of open-source models.**

#### 2. Distributional Equivalence Between COCO-train-OOD and COCO-val-OOD.

Under the same semantic division rules, we construct COCO-train-OOD and COCO-val-OOD and perform an equivalence test in the joint space of image features and category labels using kernel MMD combined with the TOST procedure. The results indicate that **the two subsets are statistically indistinguishable and can be regarded as two independent samples drawn from the same underlying distribution, i.e., they exhibit distributional equivalence.**

Therefore, in subsequent reasoning, it can be directly treated as a known premise:

- **For the open-source model collection  $\mathcal{M}_{\text{open}}$ ,  $D_{\text{OOD}}$  is OOD in terms of the training distribution.**

For closed-source models (such as GPT-4o, Gemini, etc.), since their training data is inaccessible, we cannot directly determine whether  $D_{\text{OOD}}$  falls outside their training distribution. To address this, we employ behavioral consistency inference:

- First, inductively derive observable “open-source model OOD behavioral patterns”—i.e., OOD behavioral signatures—from the inference results of open-source models on  $D_{\text{OOD}}$ .
- Subsequently, verify whether the behavior of closed-source models on the same data  $D_{\text{OOD}}$  aligns with these known “open-source OOD behavioral patterns”.

Suppose its behavior aligns with the OOD behavioral signature of the open-source model. In that case, it can be inferred that the closed-source model’s response pattern on  $D_{\text{OOD}}$  corresponds to its performance on typical OOD data. This indicates that it exhibits behavioral characteristics consistent with OOD scenarios on that dataset.

### B.2.1 BEHAVIORAL STATISTICS AND PERFORMANCE DEGRADATION

For any model  $m$  (either open-source or closed-source) and for any data split  $S \in \{ID, OOD\}$ , we define the basic classification metrics as follows:

$$T_{(m,S)}^{(k)} \in \{\text{ACC}, \text{F1}, \text{Precision}, \text{Recall}, \text{MCC}\}, \quad k = 1, \dots, K, \quad (16)$$

For example,  $T_{(m,S)}^{(1)} = \text{ACC}_{m,S}$  and  $T_{(m,S)}^{(2)} = \text{F1}_{m,S}$  correspond to five performance statistics respectively.

For each metric dimension  $k$ , we define the ID  $\rightarrow$  OOD performance degradation measure:

$$\Delta_m^{(k)} := T_{m,ID}^{(k)} - T_{m,OOD}^{(k)}. \quad (17)$$

Intuitively,  $\Delta_m^{(k)} > 0$  indicates that the model’s performance degrades on semantically OOD data; the larger the degradation, the more pronounced the OOD behavior in this dimension. In practice, we perform bootstrap resampling on the model’s outputs across each data division to obtain the 95% confidence interval for  $\Delta_m^{(k)}$ :

$$\text{CI}_{95}(\Delta_m^{(k)}) = [L_m^{(k)}, U_m^{(k)}]. \quad (18)$$

As demonstrated by the aforementioned distribution homogeneity analysis, for the open-source model ensemble,  $D_{\text{OOD}}$  is indeed OOD on the training distribution. We can further abstract a set of observable OOD behavioral patterns from its performance on  $D_{\text{OOD}}$ , namely the OOD behavioral signatures of open-source models.

Let the open-source model collection be denoted as  $\mathcal{M}_{\text{open}}$ . For each metric dimension  $k$ , we define a unified degradation band from the confidence intervals  $[L_m^{(k)}, U_m^{(k)}]$  of all open-source models:

$$I_{(k)}^{\text{gap}} := \left[ \min_{m \in \mathcal{M}_{\text{open}}} L_m^{(k)}, \max_{m \in \mathcal{M}_{\text{open}}} U_m^{(k)} \right]. \quad (19)$$

This interval  $I_{(k)}^{\text{gap}}$  represents the typical performance degradation range exhibited by the open-source model on the  $k$ th performance metric dimension for  $D_{\text{OOD}}$  instances that have been confirmed as OOD.

Let the set of degeneracy bands for all metric dimensions be denoted as:

$$A := \{I_{(k)}^{\text{gap}} | k = 1, \dots, K\}. \quad (20)$$

We refer to  $A$  as the behavioral signature of the open-source model when processing semantic OOD inputs. It reflects the systematic decline pattern in the model’s performance from ID to OOD when confronted with semantic OOD. It is crucial to emphasize that Signature  $A$  is not the definition of OOD itself; rather, the definition of OOD stems from data construction and training distribution analysis. Signature  $A$  is merely an observable “behavioral fingerprint” of OOD’s effect on model behavior.

## B.2.2 BEHAVIORAL VERIFICATION OF CLOSED-SOURCE MODELS

For closed-source model  $c$ , we cannot observe its training data. Therefore, we employ behavioral-side statistical tests to address two questions:

1. Does the closed-source model exhibit significant ID  $\rightarrow$  OOD degradation on  $D_{\text{OOD}}$ ?
2. Does the degradation magnitude of the closed-source model align with the OOD behavioral signature  $A$  of the open-source model?

### (1) Significance Test of Performance Differences Between Closed-Source Models in ID vs OOD.

For each metric dimension  $k$ , we define the same degeneracy measure for closed-source models:

$$\Delta_{c,obs}^{(k)} := T_{c,ID}^{(k)} - T_{c,OOD}^{(k)}, \quad (21)$$

where  $\Delta_{c,obs}^{(k)}$  denotes the observed degradation amount calculated based on the valid ID/OOD classification.

To verify whether this represents OOD behavior rather than random fluctuations, we construct the following hypothesis test for each  $k$ :

- Null hypothesis  $H_0$ : The closed-source model exhibits no performance difference between ID and OOD:

$$H_0 : T_{c,ID}^{(k)} = T_{c,OOD}^{(k)}.$$

- Alternative hypothesis  $H_1$ : The closed-source model performs worse on OOD (degradation exists):

$$H_1 : T_{c,ID}^{(k)} > T_{c,OOD}^{(k)}.$$

We achieve this through a permutation test of ID and OOD sample labels. Under the null hypothesis  $H_0$ , ID/OOD labels should be interchangeable, yielding the null distribution of the degradation amount  $\Delta_c^{(k)}$ . If the observed  $\Delta_c^{(k)}$  falls in the extreme tail of this distribution (*e.g.*, beyond the top 5%), reject  $H_0$  and conclude that degradation is significant. Therefore, we randomly permute the ID/OOD labels of the original data and recalculate the degradation amount for each permutation:

$$\delta_b^{(k)}, \quad b = 1, \dots, B,$$

where  $B$  represents the number of samples. These constitute the empirical distribution under the null hypothesis:

$$\{\Delta_b^{(k)}\}_{b=1}^B.$$

The standard Monte-Carlo p-value correction formula for permutation tests calculated via this empirical distribution is:

$$p_c^{(k)} = P_{H_0}(\Delta^{(k)} \geq \Delta_{c,obs}^{(k)}) \approx \frac{1 + \sum_{b=1}^B \mathbb{1}(\Delta_b^{(k)} \geq \Delta_{c,obs}^{(k)})}{B + 1}, \quad (22)$$

when  $p_c^{(k)} < 0.05$ , we reject  $H_0$  as statistically significant. That is, the closed-source model exhibits significant ID  $\rightarrow$  OOD performance degradation on  $D_{\text{OOD}}$ , which is a typical behavioral signal when semantic OOD is applied to the model.

**Observation Conclusions:** For closed-source models (GPT-4o and Gemini), we calculated the observed measure of ID  $\rightarrow$  OOD performance degradation  $\Delta_{c,obs}^{(k)}$  across five performance dimensions (Accuracy, F1, Precision, Recall, MCC). We constructed the corresponding null hypothesis distribution based on  $B = 2000$  label permutations. Figure 9 presents violin plots illustrating the

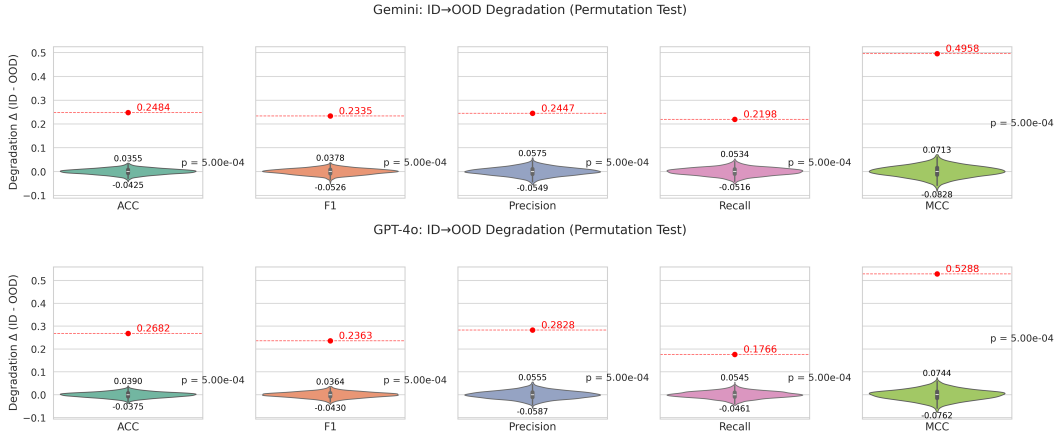


Figure 9: This figure illustrates the distribution of degradation amounts under the null hypothesis  $H_0$  (no performance difference between ID and OOD) and the positions of the observed values. As illustrated, for both closed-source models and across all metrics, the observed degradation falls in the extreme tail of the permutation distribution. The permutation p-values are all below  $10^{-3}$ , and out of 2000 permutations, none produced a degradation larger than the observed value.

distribution of degradation amounts and the positions of observed values for each metric under the null hypothesis  $H_0$  (no performance difference between ID and OOD). It is evident that for all metrics and both closed-source models, the observed degradation consistently resides in the extreme tail of the permutation distribution: their permutation p-values are all below  $10^{-3}$ , and across all 2000 permutations, not a single permutation resulted in degradation exceeding the actual observed values. From a statistical inference perspective, this implies: at the commonly used significance level  $\alpha = 0.05$ , the stricter  $\alpha = 0.01$ , and even  $\alpha = 0.001$ , we have sufficient evidence to reject the null hypothesis  $H_0 : T_{c,ID}^{(k)} = T_{c,OOD}^{(k)}$  and accept the one-tailed alternative hypothesis  $H_0 : T_{c,ID}^{(k)} > T_{c,OOD}^{(k)}$ . In other words, under the current data and testing framework, **the closed-source model exhibits statistically significant performance differences between the ID subset and the OOD-labeled subset, and this difference is highly unlikely to be caused by random label perturbations.**

(2) **Consistency of OOD Behavioral Signatures Between Closed-Source and Open-Source Models.** We aggregate the bootstrapped degradation values of the open-source models to obtain population-level statistics:

- Open-source mean degradation:  $\mu_{\text{open}}^{(k)}$ .
- Open-source variability (natural fluctuation scale):  $\sigma_{\text{open}}^{(k)}$ .

where  $\mu_{\text{open}}^{(k)}$  expresses the typical degradation magnitude of open-source models on semantic OOD;  $\sigma_{\text{open}}^{(k)}$  represents the natural fluctuation scale of degradation within the open-source community (model variance + sample noise). Furthermore, we define this natural fluctuation scale as the tolerance threshold:

$$\eta_k := \sigma_{\text{open}}^{(k)}.$$

Suppose the performance degradation of closed-source models falls within the same order of magnitude as the average degradation of open-source models. In that case, we consider their behavior equivalent in this metric dimension.

Comparing the difference between closed-source and open-source models:

$$\delta^{(k)} := \Delta_c^{(k)} - \mu_{\text{open}}^{(k)}. \quad (23)$$

- If  $\delta^{(k)} = 0$ : closed-source degradation = open-source average degradation;
- If  $|\delta^{(k)}|$  is small (*e.g.*, within  $\pm\eta_k$ ): closed-source degradation and open-source degradation are "of the same order of magnitude";
- If  $|\delta^{(k)}|$  is much larger than  $\eta_k$ , closed-source behavior and open-source OOD behavior are significantly inconsistent.

To obtain the distribution of  $\delta^{(k)}$  and its 90% confidence interval, we perform  $B$  bootstrap resamples on both ID and OOD samples. In each resample, we simultaneously estimate the following quantities:

- **Degradation of the closed-source model:**  
 $\Delta_{c,b}^{(k)}$ , denoting the performance degradation from ID to OOD for the closed-source model in the  $b$ th resample;
- **The average degradation of the open-source model population:**  
 $\mu_{open,b}^{(k)}$ , representing the average degradation of the open-source model population during the  $b$ th resampling (averaged across the two open-source models).

The difference between the two is defined as:

$$\delta_b^{(k)} := \Delta_{c,b}^{(k)} - \mu_{open,b}^{(k)}, \quad b = 1, \dots, B. \quad (24)$$

Thus we obtain a set  $\{\delta_b^{(k)}\}_{b=1}^B$ , approximating the sampling distribution of  $\delta^{(k)}$ .

Equivalence test hypotheses:

- Null hypothesis  $H_0$  (non-equivalence):

$$H_0 : |\delta^{(k)}| \geq \eta_k$$

*i.e.*, the difference between closed-source degradation and open-source degradation on metric  $k$  is at least  $\eta_k$ , indicating behavioral non-equivalence.

- Alternative hypothesis  $H_1$  (equivalence):

$$H_1 : |\delta^{(k)}| < \eta_k$$

*i.e.*, the difference is constrained within the interval  $(-\eta_k, \eta_k)$ , which can be regarded as equivalent.

In practice, we construct a confidence interval using  $\{\delta_b^{(k)}\}$  and employ CI to implement the TOST decision. We adopt the standard equivalent test framework, using a 90% confidence interval at a significance level of  $\alpha = 0.05$ . Specifically, we extract the 5th and 95th percentiles from  $\{\delta_b^{(k)}\}$  to obtain the 90% CI:

$$\text{CI}_{90}(\delta^{(k)}) = [L_\delta^{(k)}, U_\delta^{(k)}]. \quad (25)$$

Adopting the TOST equivalence criterion, if:

$$\text{CI}_{90}(\delta^{(k)}) \subset (-\eta_k, \eta_k), \quad (26)$$

At the  $\alpha = 0.05$  significance level, we reject the null hypothesis  $H_0 : |\delta^{(k)}| \geq \eta_k$  and accept  $H_1 : |\delta^{(k)}| < \eta_k$ . This indicates that:

The difference between closed-source degradation and open-source average degradation is constrained within the "open-source natural fluctuation scale  $\sigma_{open}^{(k)}$ " in dimension  $k$ , rendering it equivalent to open-source OOD behavior.

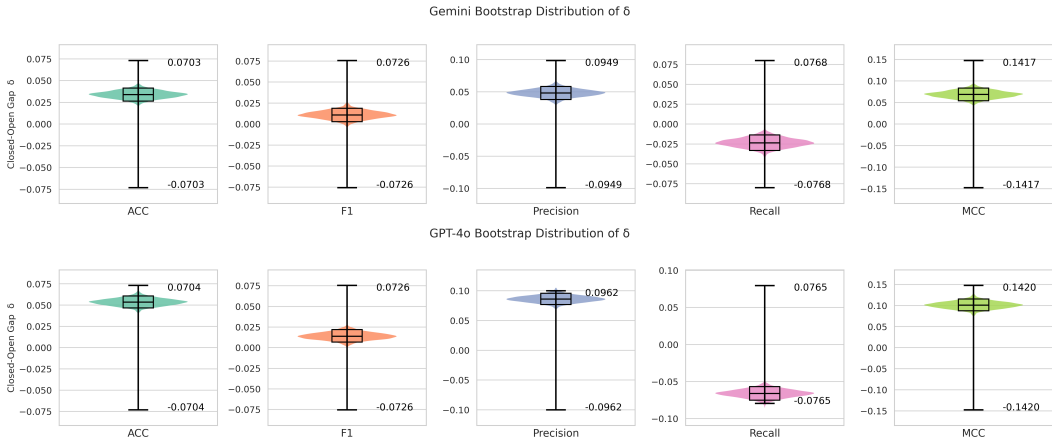


Figure 10: We selected InternVL2-8B and InternVL2.5-8B as open-source reference models and constructed their behavioral signatures based on bootstrap estimates of performance degradation. We then applied the TOST equivalence testing framework to assess the behavioral consistency of closed-source models (GPT-4o and Gemini) under semantic OOD conditions. The bootstrap distributions of  $\delta^{(k)}$  for the five evaluation metrics (Accuracy, F1, Precision, Recall, MCC), along with their 90% confidence intervals (CI90) and the corresponding tolerance thresholds  $\eta_k$ .

**Observational Conclusion:** In this study, we selected InternVL2-8B and InternVL2.5-8B as open-source reference models. Based on their bootstrap degeneracy estimates, we constructed open-source behavioral signatures. We then employed the TOST equivalence testing framework to evaluate the behavioral consistency of closed-source models (GPT-4o and Gemini) on semantic OOD data. Figure 10 displays the  $\delta^{(k)}$  bootstrap distributions of closed-source models across five metrics (Accuracy, F1, Precision, Recall, MCC), their 90% confidence intervals (CI<sub>90</sub>), and corresponding tolerance thresholds ( $\eta_k$ ).

The result from GPT-4o is:

Accuracy:	CI <sub>90</sub> = [4.67%, 6.09%],	$\eta_k = 7.04\%$
F1:	CI <sub>90</sub> = [0.68%, 2.19%],	$\eta_k = 7.26\%$
Precision:	CI <sub>90</sub> = [7.69%, 9.58%],	$\eta_k = 9.62\%$
Recall:	CI <sub>90</sub> = [-7.51%, -5.68%],	$\eta_k = 7.65\%$
MCC:	CI <sub>90</sub> = [8.74%, 11.55%],	$\eta_k = 14.20\%$

As shown, the 90% confidence intervals for all five metrics fall within the corresponding tolerance ranges  $(-\eta_k, \eta_k)$ , i.e., Equ. 26, satisfying the equivalence criterion. In other words, across all performance dimensions, the ID  $\rightarrow$  OOD degradation of GPT-4o fluctuates at the same order of magnitude as the average degradation of the open-source model population, without exceeding the natural variability exhibited by the open-source models.

Gemini’s results also yielded highly consistent conclusions:

Accuracy:	CI <sub>90</sub> = [2.65%, 4.14%],	$\eta_k = 7.03\%$
F1:	CI <sub>90</sub> = [0.28%, 1.89%],	$\eta_k = 7.26\%$
Precision:	CI <sub>90</sub> = [3.80%, 5.84%],	$\eta_k = 9.49\%$
Recall:	CI <sub>90</sub> = [-3.32%, -1.36%],	$\eta_k = 7.68\%$
MCC:	CI <sub>90</sub> = [5.39%, 8.33%],	$\eta_k = 14.17\%$

Similarly, all metrics satisfy Equ. 26, indicating that Gemini’s degradation also falls stably within the natural fluctuation range of degradation observed in open-source communities.

In summary, GPT-4o and Gemini are statistically equivalent to the open-source OOD behavioral signature across all five performance dimensions—Accuracy, F1, Precision, Recall, and MCC. Their

1566 ID  $\rightarrow$  OOD degradation magnitudes remain within the same order of magnitude as those of the open-  
1567 source models, exhibiting no deviations beyond the natural variability scale ( $\sigma_{\text{open}}$ ). These findings  
1568 indicate that the closed-source models display OOD response patterns on our semantic OOD data  
1569 that are consistent with those observed in the open-source models.

1570 From the two types of analyses above, we obtain the following conclusions. Combining (1) the  
1571 significance tests on ID vs. OOD performance differences for closed-source models and (2) the  
1572 consistency between closed-source behavior and the open-source OOD behavioral signature, we  
1573 observe that:

- 1574
- 1575 (1). Closed-source models exhibit statistically significant ID  $\rightarrow$  OOD performance degradation  
1576 on  $D_{\text{OOD}}$ .
  - 1577 (2). The magnitude of this degradation across all key metrics falls within the natural OOD vari-  
1578 ability range of the open-source models, consistent with the open-source OOD behavioral  
1579 signature.

1580 Thus, on the same semantic OOD dataset, the ID  $\rightarrow$  OOD degradation patterns of closed-source  
1581 models—both in direction and magnitude—align with those of open-source models on known OOD  
1582 tests. This supports the view that **closed-source models treat  $D_{\text{OOD}}$  as OOD data in a manner**  
1583 **equivalent to open-source models.**  
1584

1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619

## C DIFFERENCES BETWEEN OODBENCH AND HARD SAMPLE

Hard samples refer to data instances that exhibit high uncertainty, low confidence, or are prone to misclassification during model training or inference. Although they originate from the training distribution (in-distribution, ID), they typically reside in the tail or low-density regions — the marginal distribution portion. Typical hard samples include: blurred, occluded, or low-resolution images; instances with ambiguous semantic boundaries or highly similar categories (*e.g.*, dogs and wolves); and minority class samples under long-tail distributions. They essentially represent “challenging points” within the distribution, often amenable to fitting through enhanced generalization capabilities, sample reweighting, or increased sample size. In contrast, this paper refines the definition of out-of-distribution (OOD) samples from a human perception perspective, categorizing them into two types: (1) objects in an image that are non-main objects and semantically unrelated to the main object; and (2) variants or abnormal forms of the target object. These samples fundamentally represent “semantic OOD” data that multimodal large models struggle to cover during semantic learning.

To investigate the distinction between OODBench and hard samples, we drew inspiration from the OHEM method Shrivastava et al. (2016) to construct a set of hard samples on the COCO dataset and evaluated them using nine state-of-the-art models. By comparing the performance (*i.e.*, behavioral patterns) of different models on hard samples versus OODBench, we aim to reveal the differences between the two. In the data partitioning process, we adopted an offline hard sample mining approach, with the specific algorithm detailed in Table 1. First, a sufficiently converged detector is selected to define sample difficulty (YOLOv8-xlarge Ultralytics (2023) was used as the detector in this study). Perform inference on the input image to generate a set of candidate boxes (proposals). For each candidate box, compute the prediction distribution  $p(c | x)$ , where  $c \in \{1, \dots, C\}$ , and determine the predicted class  $\hat{c} = \arg \max p(c | x)$ . Subsequently, we computed a hardness score for each sample, using the commonly employed Focal Loss Lin et al. (2017) value as the metric:

$$h(x) = (1 - p(\hat{c} | x))^\gamma \cdot (-\log p(\hat{c} | x)), \quad \gamma = 2 \quad (27)$$

among these, scores for easily classifiable samples are significantly suppressed, retaining only hard samples. Next, for all proposals within each image, we group them by predicted category and select the top-k samples with the highest hardness within each category to add to the candidate hard sample list. After traversing the entire dataset, the top-k hard samples from each category within every image are aggregated. Finally, the hard sample list is globally sorted by hardness score, and the top-q% samples from each category are selected to form the final hard sample set.

Table 8 demonstrates the performance of ID data, OODBench data, and hard samples across nine state-of-the-art multimodal large language models (MLLMs). As shown in Figure 11, we plotted the performance differences of each model on hard samples and OOD-H compared to in-distribution data, then sorted the results in ascending order based on the magnitude of performance decline on hard samples. Analysis reveals significant variations in the challenge posed by hard samples across models: for instance, Gemini’s performance declined by only 5.96%, while Qwen2.5-VL-Instruct experienced a substantial 33.62% drop. This indicates that hard samples remain in-distribution data, with performance degradation primarily dependent on a model’s robustness and generalization capabilities. Specifically, certain models exhibited pronounced performance degradation on specific hard samples (*e.g.*, DeepSeek-VL-7B-Chat, Llama-3.2-11B-Vision-Instruct, and Qwen2.5-VL-7B-Instruct, all of which exceeded a 30% decline), while others remained largely unaffected (*e.g.*, Gemini and InternVL2-8B). In contrast, OODBench-defined OOD samples reveal a consistent performance decline across all nine models, with even robust models failing to significantly mitigate this effect. For example, GPT-4o experienced a 27.65% performance drop on OOD samples. This result indicates that OOD samples exhibit a distribution shift property consistent across models, rather than relying solely on individual model robustness.

### C.1 STATISTICAL DISTINCTION BETWEEN HARD AND OOD

**Setup.** For each model  $m = 1, \dots, M$  (here  $M = 9$ ), we compute the performance degradation on hard samples  $\Delta_{\text{Hard}}(m)$  and the performance degradation on our dataset  $\Delta_{\text{OOD}}(m)$ . We analyze the following aspects: (i) variability across models, (ii) alignment of performance degradation patterns, and (iii) overlap rate between data partitioned by OODBench and OHEM.

```

1674 Algorithm 1: Hard Sample Mining via Focal Loss Scoring
1675
1676 Input: A converged detector  $D$  (YOLOv8-X),
1677 Image set  $\mathcal{I}$ ,
1678 Focal loss parameter  $\gamma = 2$ ,
1679 Top- $k$  proposals per class per image,
1680 Top- $q\%$  selection threshold.
1681 Output: Hard sample list  $\mathcal{H}$ 
1682
1683 Initialize empty list  $\mathcal{H} \leftarrow \emptyset$ ;
1684 foreach image  $x \in \mathcal{I}$  do
1685     Generate proposal set  $\mathcal{P}(x)$  by  $D$ ;
1686     foreach proposal  $p \in \mathcal{P}(x)$  do
1687         Obtain class probability distribution  $p(c|x), c \in \{1, \dots, C\}$ ;
1688         Predicted class  $\hat{c} \leftarrow \arg \max_c p(c|x)$ ;
1689         Compute hardness score:
1690
1691         
$$h(x) = (1 - p(\hat{c}|x))^\gamma \cdot (-\log p(\hat{c}|x))$$

1692
1693     Group all proposals by predicted class  $c$ ;
1694     For each class  $c$ , select top- $k$  proposals with highest  $h(x)$ ;
1695     Add selected proposals into  $\mathcal{H}$ ;
1696
1697 Sort  $\mathcal{H}$  by hardness score  $h(x)$  in descending order;
1698 For each class  $c$ , retain top- $q\%$  proposals as final hard samples;
1699 return  $\mathcal{H}$ 

```

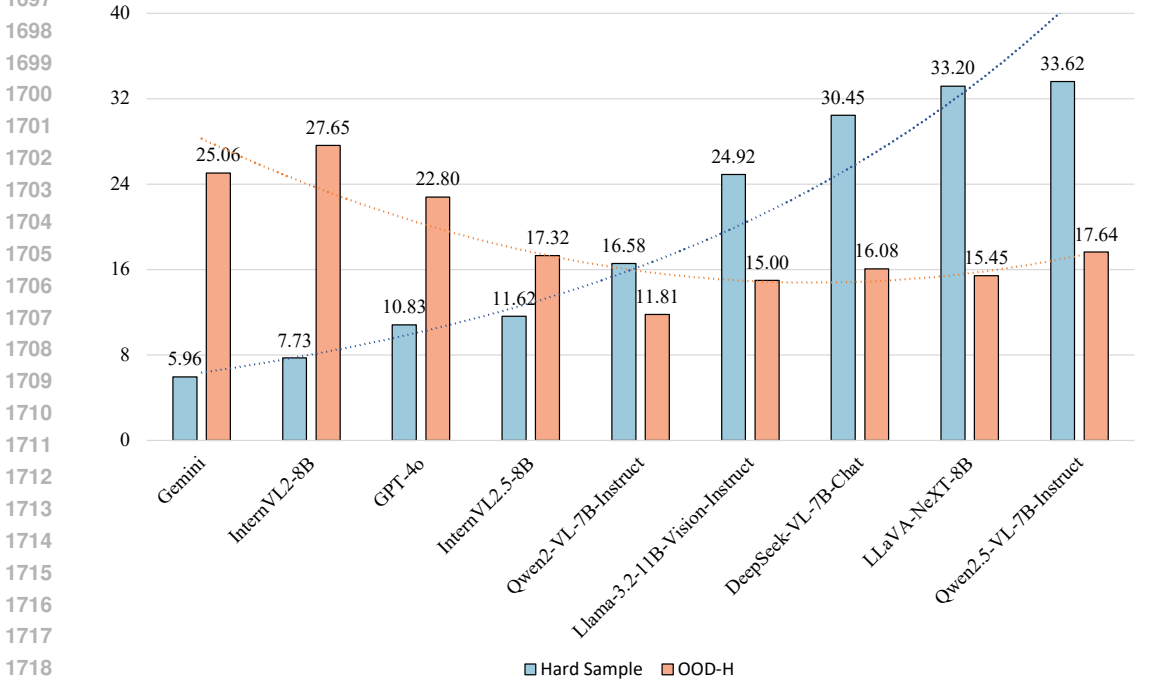


Figure 11: The figure displays the histogram of accuracy difference distributions between ID data and Hard samples, as well as OOD-H data on the COCO dataset. The bars are sorted in ascending order based on the Hard sample difference, with an exponential trendline fitted to characterize the overall trend.

**Evidence 1 — Variance Test on Full Sets (Hard vs OOD)**

We compare the variance across models using an F-test:

$$F = \frac{\text{Var}(\Delta_{\text{Hard}})}{\text{Var}(\Delta_{\text{OOD}})} \sim F(\nu_1, \nu_2), \nu_1 = \nu_2 = M - 1 \tag{28}$$

Model	Data Type	COCO-OOD-Hard Sample Performance					
		Num	Accuracy(%)	F1(%)	Precision(%)	Recall(%)	MCC(%)
Random Chance	-	-	50.00%	50.00%	50.00%	50.00%	0.00%
<b>Open-source Models</b>							
LLaVA-NeXT-8B	ID	6000	89.90	90.17	87.83	92.63	79.92
	OOD-H	6000	74.45	69.53	86.12	58.30	51.67
DeepSeek-VL-7B-Chat	Hard Sample	5282	56.74	30.27	77.99	18.78	20.71
	ID	6000	92.65	92.84	90.48	95.33	85.42
InternVL2-8B	OOD-H	6000	76.57	74.09	82.85	67.00	54.13
	Hard Sample	5282	62.23	47.38	78.09	34.00	29.63
InternVL2.5-8B	ID	6000	92.83	92.85	92.69	93.00	85.67
	OOD-H	6000	65.18	64.92	65.41	64.43	30.37
Llama-3.2-11B-Vision-Instruct	Hard Sample	5282	85.12	84.29	89.28	79.82	70.64
	ID	6000	97.02	97.01	97.25	96.77	94.03
Qwen2-VL-7B-Instruct	OOD-H	6000	79.70	78.65	82.95	74.77	59.69
	Hard Sample	5282	85.38	84.23	91.48	78.04	71.54
Qwen2.5-VL-7B-Instruct	ID	6000	97.02	97.04	96.14	97.97	94.05
	OOD-H	6000	82.02	82.91	78.99	87.23	64.38
GPT-4o	Hard Sample	5282	72.09	71.11	73.71	68.69	44.29
	ID	6000	89.48	90.11	85.01	95.87	79.62
Gemini	OOD-H	6000	77.67	75.48	83.66	68.77	56.23
	Hard Sample	5284	72.88	66.56	86.79	53.97	49.43
GPT-4o	ID	6000	95.92	95.79	98.93	92.83	92.01
	OOD-H	6000	78.28	74.04	92.03	61.93	59.86
GPT-4o	Hard Sample	5282	64.27	47.69	89.03	32.56	36.93
	<b>Close-source Models</b>						
Gemini	ID	5996	91.86	92.06	89.84	94.40	83.83
	OOD-H	5994	66.80	68.46	65.20	72.07	33.79
GPT-4o	Hard Sample	5276	85.90	85.91	85.84	85.97	71.80
	ID	5998	93.33	93.45	91.82	95.13	86.72
GPT-4o	OOD-H	6000	70.53	72.06	68.51	76.00	41.31
	Hard Sample	5282	82.49	81.94	84.60	79.44	65.10

Table 8: This table compares the performance of ID, OOD-H data, and in-Hard samples—segmented based on the COCO dataset—across nine leading SOTA MLLMs. We ensured that all testing configurations remained identical except for the data division method, guaranteeing that any observed differences stem solely from the inherent characteristics of the data.

In our experiments,  $F(8, 8) = 4.59$ ,  $p = 0.0227$ , indicating that the variance in performance degradation caused by Hard samples is significantly greater than that caused by OOD samples.

**Explanation:** Hard samples exhibit greater model dependency, while OOD samples induce a more consistent (model-independent) performance degradation.

### Evidence 2 — Correlation on full sets (Hard vs OOD)

We measure the alignment of performance degradation patterns through correlation analysis:

$$r = \text{corr}(\Delta_{\text{Hard}}, \Delta_{\text{OOD}}) \quad (29)$$

We observe negative correlations (Pearson  $r = -0.65$ ,  $p = 0.059$ ; Spearman  $\rho = -0.58$ ,  $p = 0.099$ ). These results indicate inconsistencies in the failure patterns of models on Hard and OOD samples: models exhibiting significant performance degradation on Hard samples do not necessarily show similar degradation on OOD samples, and vice versa. Furthermore, from the performance trend lines shown in Figure 11, we observe an X-shaped decline pattern, where models exhibit significant fluctuations or reversals at certain points. This phenomenon aligns with the correlation test results, further validating the misaligned patterns of performance shifts across Hard and OOD samples. The X-shaped trend lines reflect the models’ markedly divergent responses to different sample types (Hard vs. OOD).

### Hypotheses

- $H_0$  (Hard Hypothesis): The OODBench dataset shares the same distribution as the in-distribution Hard subset. In other words, the statistical characteristics of its cross-model performance degradation vector  $\Delta V$  (including variance and correlation/alignment with the baseline Hard) should fall within the “empirical distribution of the Hard subset.”

- $\mathbf{H}_1$  (OOD Hypothesis): OODBench does not belong to the Hard subset, and the same-distribution assumption does not hold; its statistical characteristics will significantly deviate from the “Hard baseline.”

**Test statistics.** Record the cross-model performance decline vector (ID performance - Hard sample performance) for the baseline Hard model as  $\Delta_{\text{Hard}}$  (length equal to the number of models  $M = 9$ ).

**Resample details.** We performed resampling based on Algorithm 1 to construct the empirical baseline for the Hard subset. Specifically,  $B=1000$  independent samples were conducted, each randomly selecting 500 Hard samples from the COCO validation pool. All samples were selected according to a unified criterion: scoring using focal loss and choosing the top-k samples within each category. Each sampling process was independent, yielding distinct sets of 1000 Hard samples, thereby ensuring the randomness and diversity of the baseline distribution.

### Evidence 3 — Empirical variance test

- **Statistical Measure:**  $s_{\text{OOD}}^2 = \text{Var}(\Delta_{\text{OOD}})$
- **Test Direction (one side):** left tail. Hard samples are typically more model-dependent, exhibiting greater cross-model variance; if  $s_{\text{OOD}}^2$  is significantly smaller, reject  $\mathbf{H}_0$ .
- **Baseline Construction:** Resample  $H^{(b)}$  from the COCO validation set using the same mechanism as Algorithm 1’s rule.
- **Baseline Distribution:**  $S_b = \text{Var}(\Delta_{H^{(b)}})$
- **Decision:** If the empirical p-value  $\hat{p}_{\text{emp}} < 0.05$  (i.e., OOD falls below the 5th percentile of the Hard baseline distribution), reject  $\mathbf{H}_0$ , concluding that the candidate set exhibits greater model-agnostic consistency in cross-model score drops, consistent with OOD characteristics. Results are shown in Figure 12 (left). The empirical p-value is less than  $1/(B+1)$  (i.e.,  $\hat{p}_{\text{emp}} < 0.001$ ), strongly rejecting  $\mathbf{H}_0$ . This indicates that the statistical characteristics of the candidate set significantly deviate from the Hard baseline.

### Evidence 4 — Empirical correlation test

- **Statistical Measure:**  $r_{\text{OOD}} = \text{corr}(\Delta_{\text{Hard}}, \Delta_{\text{OOD}})$
- **Test Direction (one side):** left tail. Hard vs Hard drop-off modes are typically non-negative or positively aligned; if  $r_{\text{OOD}}$  is significantly low, reject  $\mathbf{H}_0$ .
- **Baseline Construction:** Resample  $B$  hard subsets  $H^{(b)}$  from the COCO validation set and compute their correlation with the baseline hard set.
- **Baseline Distribution:**  $r_b = \text{corr}(\Delta_{\text{Hard}}, \Delta_{\text{OOD}})$
- **Decision:** If the empirical p-value  $\hat{p}_{\text{emp}} < 0.05$  (i.e.,  $r_{\text{OOD}}$  falls below the 5th percentile of the Hard baseline distribution), reject  $\mathbf{H}_0$ , indicating that the candidate set does not align with Hard’s failure pattern, consistent with OOD’s “different failure mode.” Results are shown in Figure 12: The middle subfigure displays the distribution of Pearson correlation coefficients from 1000 resamples, all exceeding 0.9; The right subfigure shows the distribution of Spearman correlation coefficients, all exceeding 0.8. In contrast, the OOD correlations with Hard are -0.65 (Pearson) and -0.58 (Spearman), significantly below the Hard baseline distribution. The empirical p-value is less than 0.001, strongly rejecting  $\mathbf{H}_0$  and further demonstrating that the statistical characteristics of the candidate set significantly deviate from the Hard baseline.

### Evidence 5 — Overlap Rate Between Hard Sample and OODBench

- **Statistical Measure:** We calculated the overlap rate between the full COCO-H dataset (5,079 samples) in OODBench and the Hard sample collection (2,641 samples) obtained by sampling according to Algorithm 1. This metric evaluates the consistency between our classified OOD data and the hard samples generated by conventional hard sample mining methods.
- **Results:** Statistical analysis revealed only 206 overlapping samples, accounting for 0.0406% of the total COCO-H dataset and 0.0780% of the total hard samples.

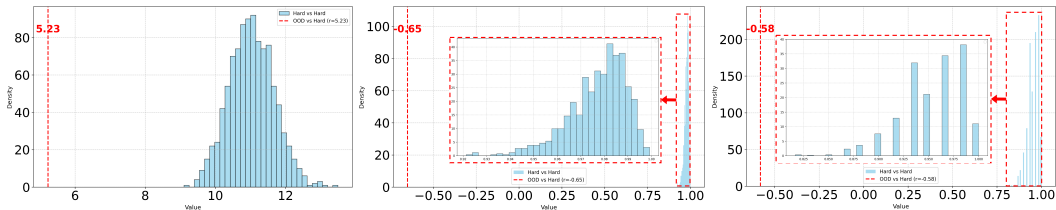


Figure 12: The figure displays the empirical distribution results for variance (left), Pearson correlation coefficient (center), and Spearman correlation coefficient (right). The x-axis of the variance subplot represents variance values, while the x-axes of the Pearson and Spearman subplots respectively denote the corresponding correlation coefficients. The y-axes of all three subplots indicate sample sizes. The red dashed lines in each subplot mark the statistical values of the out-of-distribution (OOD) data.

- **Conclusion:** Such a low overlap rate indicates that the OOD data categorized by OOD-Bench fundamentally differs from the hard samples obtained through traditional hard sample mining methods. In other words, OODBench captures more universally applicable out-of-distribution patterns rather than conventional in-distribution hard samples.

Taken together, the five lines of evidence consistently demonstrate that the OODBench data cannot be regarded as conventional hard samples. Evidence 1 (variance test on full sets) shows that performance degradation on Hard samples exhibits significantly larger variance across models than on OOD, indicating stronger model dependency for Hard. Evidence 2 (correlation analysis) reveals negative associations between Hard and OOD drop patterns, further corroborated by the X-shaped trend lines, highlighting misaligned failure modes. Evidence 3 (empirical variance test) confirms that the variance of OOD drops is significantly smaller than the empirical Hard baseline, rejecting  $H_0$ . Evidence 4 (empirical correlation test) further establishes that OOD correlations with Hard (-0.65, -0.58) fall far below the Hard-vs-Hard baseline (all  $> 0.9$  Pearson,  $> 0.8$  Spearman), again strongly rejecting  $H_0$ . Finally, Evidence 5 (overlap rate) shows that the intersection between OODBench samples and mined hard samples is minimal (206 out of 5,079), underscoring their fundamental distinction.

Across all five evidences, the results are significant and directionally consistent: OODBench captures distributional shifts and model-agnostic failure modes, whereas Hard samples remain in-distribution and model-dependent. Therefore, our OOD data are not merely hard samples but instead represent a fundamentally different category aligned with out-of-distribution characteristics.

## D DIFFERENCES BETWEEN OODBENCH AND HALLUCINATION

**Out-of-Distribution (OOD) may cause hallucinations, but hallucinations are not necessarily caused by OOD.** We analyse in depth the differences between OOD and hallucinations in terms of concepts, interrelationships and experimental results to demonstrate the distinctiveness of the both.

**Hallucination definitions in the Large Vision- Language Model (LVLM):** Hallucination in LVLM denotes a disagreement between factual content of images and corresponding generated textual content, akin to the solely textual hallucination encountered in large language models Liu et al. (2024a); Bai et al. (2024).

**Definition of OOD:** Traditionally, OOD is defined as a test example originating from a distribution different from the training data Hendrycks & Gimpel (2017). In recent studies Averly & Chao (2023); Yang et al. (2024b), the OOD detector identifies misclassified examples labelled as OOD Data to improve the safety and reliability of deep learning models in practical applications.

In LVLMs, hallucination refers to the inconsistency between the generated text description and the visual content. Our experiment shows that OOD data does not necessarily lead to hallucinations and that it is not fully equivalent to hallucinations. As shown in Tab. 1, OOD data does not necessarily trigger hallucinations. For example, on the OOD-H data, the accuracy of GPT-4o was 65.13%, implying that  $100\% - 65.13\% = 34.87\%$  of the OOD data led to hallucinations, while the remaining 65.13% of the OOD data did not trigger hallucinations. This indicates that there is not a one-to-one correspondence between OOD and hallucinations. Second, the causes of hallucinations are multiple, and not all hallucinations are caused by OOD data. For example, the ID data in Tab. 1 also triggered hallucinations. This phenomenon was also verified in the BAP experiments in Tab. 3. Taking GPT-4o as an example, we asked questions on ID data according to Basic-to-Advanced Progress, including existential description, counting description, and logical reasoning, and the performance was 94.84%, 48.26%, and 30.83% in decreasing order. This indicates that as the complexity of the inference task increases, the model becomes more hallucinatory and the match between the generated text and the visual content decreases. The results of the above experiments demonstrate that there is some intersection between OOD and hallucinations, but they are not equivalent.

Prior research Zhang et al. (2023) has shown that CoT (Chain of Thought) is an effective way to alleviate hallucinations. In our experiments, CoT did not perform as well on OOD data. For example, in Tab. 1, we observed that several models such as DeepSeek-VL-7B-Chat, DeepSeek-VL2-Small, Llama3.2-11B-Vision, Qwen2-7B-Instruct, and GPT-4o failed to improve their performance, but instead, their performance decreased to different degrees after using CoT. This phenomenon may stem from the fact that the distribution of the OOD data deviates from the training distribution, which makes CoT reason with out-of-distribution data as the starting point, thus reinforcing false assumptions, amplifying the inference bias, and ultimately reducing the model performance. We begin by hypothesising that there is a high degree of overlap between the data in OODBench and the hallucinations. Further, CoT, as a typical method to mitigate the hallucination problem, should improve the model performance on this benchmark. However, the experimental results were contrary to expectations: the model performance decreased rather than increased after the introduction of CoT. This phenomenon directly refutes our initial hypothesis that the OODBench data highly overlaps with hallucinations. Instead, it experimentally validates that there is an essential difference between our divided OOD dataset and hallucinations. Together, these experimental results demonstrate that the OOD dataset we constructed is significantly different from the hallucination.

## E ABLATION STUDY OF THE GENERALIZED OOD DETECTOR

### E.1 ABLATION STUDY OF DIFFERENT TYPES OF GENERALIZED OOD DETECTOR

In order to explore the impact of different types of OOD detectors on the performance of downstream tasks when dividing OOD data under the generalized OOD detection framework, we designed and conducted a series of systematic ablation experiments. In the specific setup, we selected GroupViT Xu et al. (2022) and LiT Zhai et al. (2022) to gradually replace the original CLIP Radford et al. (2021) with BLIP2 Zhai et al. (2022) to re-divide the samples in the COCO dataset into OOD, and we constructed two new sets of division data. Subsequently, we input the three sets of delineation results (*i.e.*, original division + division of the two alternative detectors) into two open-source multimodal models (InternVL2, Qwen2-VL) and two closed-source models (Gemini, GPT-4o), respectively, for consistency testing to assess the impact of different generalised OOD detectors on the downstream performance performance.

It should be noted that in order to control the testing overhead, the original OODBench benchmark introduces a random downsampling strategy when the number of samples exceeds 6,000. In order to avoid the possible data distribution bias introduced by downsampling, we retain all the samples judged as OOD, *i.e.*, we do not perform the downsampling operation again when we use GroupViT with LiT for reclassification. At the same time, in order to ensure the fairness of the experiment, we continue to use the sample sets corresponding to the CLIP and BLIP2 classification results in the initial benchmark to ensure the consistency of the processing flow among the three groups.

For ease of presentation, we define the three datasets as follows:

- The data divided by CLIP and BLIP2 are recorded as **Group I**.
- The data divided by BLIP2 and GroupViT are recorded as **Group II**.
- The data divided by GroupViT and LiT are recorded as **Group III**.

Model	OOD Detectors	COCO-OOD Performance						Time
		Num	Accuracy(%)	F1(%)	Precision(%)	Recall(%)	MCC(%)	
Random Chance	-	-	50.00%	50.00%	50.00%	50.00%	0.00%	-
<b>Open-source Models</b>								
Qwen2-VL-7B-Instruct	CLIP & BLIP2	10160	78.74	76.78	84.58	70.30	58.32	00:11:44 <sup>2</sup>
		$\Delta$	+ 0.00	+ 0.00	+ 0.00	+ 0.00	+ 0.00	-
	BLIP2 & GroupViT	9988	79.85	78.06	85.65	71.71	60.50	00:21:40 <sup>f</sup>
InternVL2-8B		$\Delta$	-1.11	-1.28	-1.07	-1.41	-2.18	-
	GroupViT & LiT	12090	80.17	78.46	85.88	72.22	61.12	00:26:15 <sup>f</sup>
		$\Delta$	-1.43	-1.68	-1.30	-1.92	-2.80	-
Gemini	CLIP & BLIP2	10160	66.22	65.99	66.44	65.55	32.44	01:03:50 <sup>2</sup>
		$\Delta$	+0.00	+0.00	+0.00	+0.00	+0.00	-
	BLIP2 & GroupViT	9988	67.08	66.91	67.25	66.58	34.16	01:49:45 <sup>f</sup>
GPT-4o <sup>3</sup>		$\Delta$	-0.86	-0.92	-0.81	-1.03	-1.72	-
	GroupViT & LiT	12090	68.17	67.91	68.48	67.34	36.35	02:13:10 <sup>f</sup>
		$\Delta$	-1.95	-1.92	-2.04	-1.79	-3.91	-
GPT-4o <sup>3</sup>	CLIP & BLIP2	10150	67.02	68.71	65.37	72.41	34.25	02:59:57
		$\Delta$	+0.00	+0.00	+0.00	+0.00	+0.00	-
	BLIP2 & GroupViT	9972	66.65	68.36	65.02	72.06	33.49	03:28:29
GPT-4o <sup>3</sup>		$\Delta$	+0.37	+0.35	+0.35	+0.35	+0.76	-
	GroupViT & LiT	12072	66.96	68.76	65.22	72.71	34.16	04:04:54
		$\Delta$	+0.06	-0.05	+0.15	-0.30	+0.09	-
GPT-4o <sup>3</sup>	CLIP & BLIP2	10158	73.66	74.35	72.44	76.37	47.38	09:58:34
		$\Delta$	+0.00	+0.00	+0.00	+0.00	+0.00	-
	BLIP2 & GroupViT	9908	73.74	74.57	72.28	77.01	47.58	12:32:43
GPT-4o <sup>3</sup>		$\Delta$	-0.08	-0.22	+0.16	-0.64	-0.20	-
	GroupViT & LiT	12090	75.86	76.56	74.39	78.86	51.81	08:41:31
		$\Delta$	-2.20	-2.21	-1.95	-2.49	-4.43	-

Table 9: Impact of replacing generalized OOD detectors on the downstream performance of different multimodal models. We compare the accuracy differences ( $\Delta$ ) of data divided by different combinations of detectors—Group I (CLIP + BLIP2), Group II (BLIP2 + GroupViT), and Group III (GroupViT + LiT)—across four representative VLMs (Qwen2-VL-7B-Instruct, InternVL2-8B, Gemini, and GPT-4o). Group I is used as the baseline, and the performance drop or gain of Group II and Group III is reported accordingly. The results show that although different detectors induce slight variance in OOD data division, their impact on downstream model performance is minimal, supporting the replaceability of generalized OOD detectors in the OODBench framework.

<sup>3</sup>There may be some differences between the performance results of GPT-4o in this experiment and those in Tab. 10, which mainly stems from the differences in the model versions used. To ensure the fairness and

We first evaluated Qwen2-VL by taking the performance of group **I** data as the baseline and calculating the accuracy difference ( $\delta$ ) between group **II** and group **III** data under the model relative to the baseline. The experimental results show that on Qwen2-VL, the accuracy of group **II** and group **III** data decreases by 1.11% and 1.43%, respectively, compared with that of group **I**. On Gemini, it improves by 0.86% and 0.06%, respectively, and the performance difference between the three groups of data on InternVL and GPT-4o is also minimal, and the details of the values are shown in Tab. 9. In the Time column, the time superscript indicates the number of graphics cards used in the test, and we used NVIDIA L40s graphics cards in the testing phase.

These results show that although there is some discriminative difference between the different OOD detectors when dividing the samples, this difference does not trigger a significant performance bias in the evaluation of the downstream multimodal model. **It is experimentally verified that the generalized OOD detectors in the OODBench division process are replaceable.** In other words, as long as the detector itself has good class discrimination ability, it can effectively classify OOD data for large visual language models.

In addition, we note that Tong *et al.* Tong et al. (2024; 2023). raised the issue of CLIP failure transfer, suggesting that current large VLMs may inherit transfer flaws from adopting CLIP as a visual or text encoder. The results in the Tab. 9 shows that the model’s performance in the OOD detection task remains stable even when the CLIP is replaced with other classifiers, ruling out the possibility of further failure transfer of the CLIP into the VLM in OOD detection. It is thus concluded that the performance bottleneck of VLM in such tasks does not originate from CLIP itself. This finding helps to rule out the hypothesis of further spreading of ‘CLIP failure transfer’ in VLMs, suggesting that a combination of more factors influences the performance of VLMs in this kind of task.

## E.2 ABLATION STUDIES WITH DIFFERENT NUMBERS OF GENERALIZED OOD DETECTORS

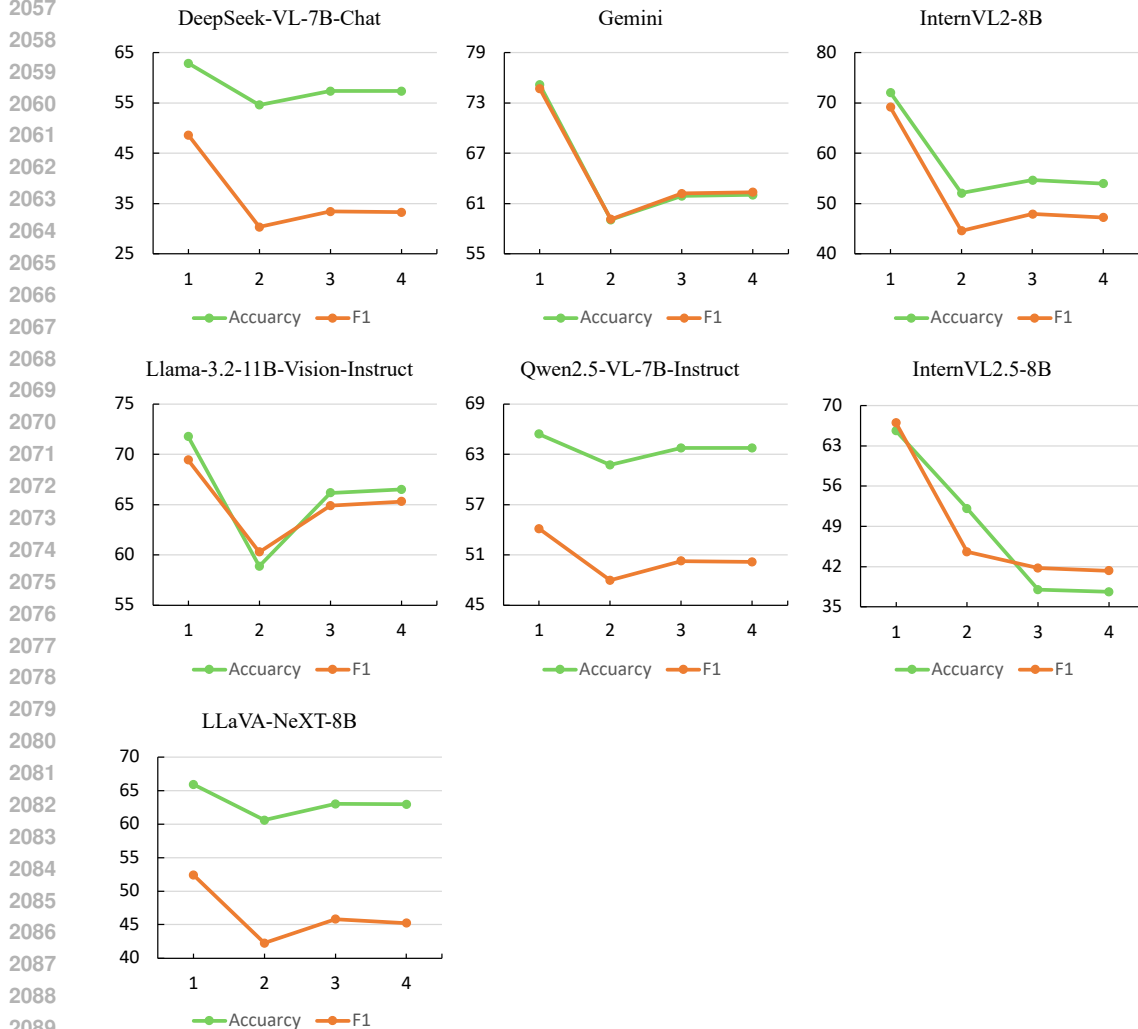
To further investigate the effect of different numbers of generalized OOD detectors on the division of OOD data under the cross-validation mechanism, we conducted ablation experiments on the number of detectors. We selected four models, CLIP Radford et al. (2021), BLIP2 Li et al. (2023b), GroupViT Xu et al. (2022), and LiT Zhai et al. (2022), as generalized OOD detectors, and sequentially increased the number of detectors involved in the division in a fixed order. Consistent with the OOD detector type ablation experiment E.1, to avoid introducing additional bias, we did not perform downsampling operations on the OOD data divided by the above models to maintain the consistency of the division process. To assess the impact of the number of detectors on the OOD data division results, we tested the OOD data divided by different numbers of detectors on several mainstream VLMs. We visualised the results in the form of a line graph (shown in Figure 13). The horizontal axis of the graph represents the number of detectors involved in the division, and the vertical axis represents the accuracy and F1 score obtained from testing on each VLM.

The experimental results show that when only a single OOD detector (number 1) is used, each VLM reaches the highest value in both Accuracy and F1, which indicates that a single detector may bring about model bias, which in turn leads to a mismatch between the divided OOD data and the potential OOD distribution of the large VLMs. When the number of detectors is increased to 2, the performance of each large VLM decreases significantly, indicating that the cross-validation mechanism can mitigate the error brought about by single-detector bias to some extent, thus obtaining data that more closely fits the OOD distribution of the large VLM. However, when the number of detectors is further increased to 3, the performance of all six VLMs except InternVL2.5-8B shows a performance rebound, *i.e.*, the data divided by three detectors generally performs higher than that of 2 detectors in terms of Accuracy and F1. Nonetheless, this performance rebound is still significantly lower than that seen with the single detector division. When the number of detectors reaches 4, we observe that the test performance is almost the same as with three detectors, suggesting that further increases in the number of detectors do not lead to substantial improvement in division, and that the division effect tends to stabilise.

Combining the results of the experiments with two to four detectors, **we found that a higher number of OOD detectors is not better. With two detectors, the division results are closest to the potential OOD distribution of the major VLMs.** Continuing to increase the number of detectors

consistency of the results of the OOD detector type ablation experiments, we uniformly used the same version of GPT-4o (20240806) for evaluation in this experiment.

2052 may instead introduce additional uncertainty, which leads to the division of data deviating from the  
 2053 potential OOD distribution of the major large VLMs and gradually converging. Based on perform-  
 2054 ance and division stability considerations, we ultimately chose to use 2 OOD detectors to construct  
 2055 the default configuration of the OOD data division scheme in OODBench.  
 2056



2091 Figure 13: Impact of the number of generalized OOD detectors on the quality of OOD data division  
 2092 under the cross-validation mechanism. The horizontal axis indicates the number of detectors in-  
 2093 volved in the division process (ranging from 1 to 4), while the vertical axis shows the corresponding  
 2094 Accuracy and F1 score (%) obtained by evaluating the divided OOD data on multiple mainstream  
 2095 VLMs. The results demonstrate that using a single detector yields the highest performance but suf-  
 2096 fers from potential bias. Incorporating two detectors significantly reduces this bias, producing OOD  
 2097 data more aligned with the VLMs’ potential OOD distribution. However, increasing the number of  
 2098 detectors beyond two offers limited benefits and may introduce instability, as evidenced by perform-  
 2099 ance rebound and eventual convergence.  
 2100  
 2101  
 2102  
 2103  
 2104  
 2105

## F OODBENCH SUB-DATASET PERFORMANCE

**Subdataset classification performance.** Tab. 10, Tab. 11, Tab. 12 and Tab. 13 show the performance of the different models on publicly available datasets collected in natural and autonomous driving scenarios, respectively. Where, OOD-S' and OOD-H' denote the initial benchmark datasets. In the Time column, the time superscript indicates the number of graphics cards used in the test, and we used NVIDIA L40s graphics cards in the testing phase.

Model	Image Encoder	Language Model	Data Type	COCO-ID-OOD Performance									
				Num	Accuracy(%)	F1(%)	Precision(%)	Recall(%)	MCC(%)	Time			
Random Chance	-	-	ID/OOD-S/OOD-H	-	50.00%	50.00%	50.00%	50.00%	0.00%	-			
<b>Open-source Models</b>													
LLaVA-NeXT-8B	CLIP-L-14	Llama-3-8B-Instruct	ID	6000	89.90	90.17	87.83	92.63	79.92	00:24:24 <sup>1</sup>			
			OOD-S	6000	74.32	70.26	83.45	60.67	50.55	00:27:16 <sup>2</sup>			
			OOD-H	6000	74.45	69.53	86.12	58.30	51.67	00:23:49 <sup>9</sup>			
			OOD-S'	7972	74.34	70.09	83.99	60.14	50.76	00:57:40 <sup>4</sup>			
			OOD-H'	10158	75.31	70.67	87.01	59.50	53.36	01:07:40 <sup>1</sup>			
			ID <sup>CoT</sup>	6000	74.92	79.02	67.91	94.47	54.14	00:22:57 <sup>2</sup>			
			OOD-S <sup>CoT</sup>	6000	73.43	70.13	80.09	62.37	48.06	00:28:34 <sup>2</sup>			
			OOD-H <sup>CoT</sup>	6000	76.65	71.52	91.66	58.63	57.14	00:27:14 <sup>2</sup>			
			ID	6000	92.65	92.84	90.48	95.33	85.42	00:32:42 <sup>1</sup>			
			OOD-S	6000	75.43	72.67	81.87	65.33	51.94	00:32:28 <sup>1</sup>			
DeepSeek-VL-7B-Chat	SigLIP-L & SAM-B	DeepSeek-LLM-7B	OOD-H	6000	76.57	74.09	82.85	67.00	54.13	00:32:26 <sup>1</sup>			
			OOD-S'	7972	75.60	72.82	82.20	65.35	52.31	00:42:57 <sup>1</sup>			
			OOD-H'	10158	77.13	74.79	83.32	67.85	55.22	00:53:33 <sup>1</sup>			
			ID <sup>CoT</sup>	6000	92.62	92.65	92.21	93.10	85.24	01:11:34 <sup>1</sup>			
			OOD-S <sup>CoT</sup>	6000	74.82	72.37	80.15	65.97	50.43	01:30:32 <sup>1</sup>			
			OOD-H <sup>CoT</sup>	6000	73.80	70.95	79.60	64.00	48.54	01:09:51 <sup>1</sup>			
			ID	6000	92.88	92.74	94.62	90.93	85.83	00:29:10 <sup>1</sup>			
			OOD-S	6000	73.97	67.74	89.03	54.67	51.96	00:27:58 <sup>1</sup>			
			OOD-H	6000	75.17	68.63	93.14	54.33	55.37	00:31:00 <sup>1</sup>			
			OOD-S'	7972	74.02	67.82	89.10	54.74	52.07	00:43:44 <sup>1</sup>			
DeepSeek-VL2-Small	SigLIP-SO400M	DeepSeekMoE LLM	OOD-H'	10158	75.59	69.32	93.24	55.17	56.06	01:00:14 <sup>1</sup>			
			ID <sup>CoT</sup>	6000	90.90	89.99	99.96	81.83	83.18	00:57:42 <sup>1</sup>			
			OOD-S <sup>CoT</sup>	6000	69.43	57.17	95.48	40.80	47.41	00:52:15 <sup>1</sup>			
			OOD-H <sup>CoT</sup>	6000	68.45	54.35	98.26	37.57	46.92	00:57:07 <sup>1</sup>			
			ID	6000	92.83	92.85	92.69	93.00	85.67	00:40:41 <sup>2</sup>			
			OOD-S	6000	77.22	76.40	79.23	73.77	54.56	00:39:38 <sup>2</sup>			
			OOD-H	6000	65.18	64.92	65.41	64.43	30.37	00:37:58 <sup>2</sup>			
			OOD-S'	7972	77.35	76.60	79.21	74.16	54.80	00:52:50 <sup>2</sup>			
			OOD-H'	10160	66.22	65.99	66.44	65.55	32.44	01:03:50 <sup>2</sup>			
			ID <sup>CoT</sup>	6000	96.18	96.07	99.01	93.30	92.52	01:20:50 <sup>2</sup>			
InternVL2-8B	InternViT-300M-448px	InternLM2.5-7b-chat	OOD-S <sup>CoT</sup>	6000	77.88	73.92	90.04	62.70	58.53	01:21:19 <sup>2</sup>			
			OOD-H <sup>CoT</sup>	6000	81.48	78.17	95.21	66.30	66.09	01:19:54 <sup>2</sup>			
			ID	6000	97.02	97.01	97.25	96.77	94.03	00:25:17 <sup>2</sup>			
			OOD-S	6000	84.88	84.18	88.29	80.43	70.04	00:26:28 <sup>2</sup>			
			OOD-H	6000	79.70	78.65	82.95	74.77	59.69	00:25:20 <sup>2</sup>			
			OOD-S'	7972	85.00	84.31	88.35	80.63	70.26	00:35:11 <sup>2</sup>			
			OOD-H'	10160	80.65	79.55	84.33	75.30	61.65	00:42:53 <sup>2</sup>			
			ID <sup>CoT</sup>	6000	96.88	96.87	97.40	96.33	93.77	01:59:05 <sup>2</sup>			
			OOD-S <sup>CoT</sup>	6000	82.02	80.31	88.71	73.37	65.01	01:48:03 <sup>2</sup>			
			OOD-H <sup>CoT</sup>	6000	81.92	80.08	89.13	72.70	64.95	01:49:08 <sup>2</sup>			
Llama-3.2-11B-Vision-Instruct	ViT-H-14	Llama3.1-8B	ID	6000	97.02	97.04	96.14	97.97	94.05	00:56:59 <sup>1</sup>			
			OOD-S	6000	82.87	83.05	82.16	83.97	65.75	01:00:41 <sup>1</sup>			
			OOD-H	6000	82.02	82.91	78.99	87.23	64.38	01:01:03 <sup>1</sup>			
			OOD-S'	7972	75.70	77.81	71.60	85.20	52.36	01:10:42 <sup>1</sup>			
			OOD-H'	10158	71.93	75.64	66.81	87.16	46.06	01:30:22 <sup>1</sup>			
			ID <sup>CoT</sup>	6000	78.32	80.58	72.97	89.97	58.24	02:26:01 <sup>1</sup>			
			OOD-S <sup>CoT</sup>	6000	67.13	69.04	65.25	73.30	34.53	02:19:54 <sup>1</sup>			
			OOD-H <sup>CoT</sup>	6000	68.78	72.68	64.62	83.03	39.19	02:16:51 <sup>1</sup>			
			ID	6000	89.48	90.11	85.01	95.87	79.62	00:06:54 <sup>2</sup>			
			OOD-S	6000	81.57	79.98	87.52	73.63	63.94	00:06:59 <sup>2</sup>			
Qwen2-VL-7B-Instruct	CLIP-L-14	Qwen2-7B	OOD-H	6000	77.67	75.48	83.66	68.77	56.23	00:07:00 <sup>2</sup>			
			OOD-S'	7972	81.18	79.49	87.35	72.93	63.24	00:09:13 <sup>2</sup>			
			OOD-H'	10160	78.74	76.78	84.58	70.30	58.32	00:11:44 <sup>2</sup>			
			ID <sup>CoT</sup>	6000	92.08	92.43	88.53	96.70	84.53	00:29:54 <sup>2</sup>			
			OOD-S <sup>CoT</sup>	6000	82.43	81.73	85.13	78.60	65.06	00:45:11 <sup>2</sup>			
			OOD-H <sup>CoT</sup>	6000	76.82	75.87	79.10	72.90	53.80	00:44:57 <sup>2</sup>			
			ID	6000	95.92	95.79	98.93	92.83	92.01	00:11:15 <sup>1</sup>			
			OOD-S	6000	79.10	75.00	93.30	62.70	61.61	00:11:10 <sup>1</sup>			
			OOD-H	6000	78.28	74.04	92.03	61.93	59.86	00:11:16 <sup>1</sup>			
			OOD-S'	7972	78.81	74.54	93.39	62.02	61.18	00:14:19 <sup>1</sup>			
Qwen2.5-VL-7B-Instruct	ViT(SwiGLU, RMSNorm)	Qwen2.5 LLM	OOD-H'	10160	79.42	75.52	93.13	63.52	62.05	00:18:40 <sup>1</sup>			
			ID <sup>CoT</sup>	6000	95.45	95.23	99.96	90.93	91.27	01:22:11 <sup>1</sup>			
			OOD-S <sup>CoT</sup>	6000	76.48	70.35	95.17	55.80	58.18	01:24:46 <sup>1</sup>			
			OOD-H <sup>CoT</sup>	6000	77.20	70.89	98.00	55.53	60.36	01:26:11 <sup>1</sup>			
			<b>Closed-source Models</b>										
			Gemini			ID	5996	91.86	92.06	89.84	94.40	83.83	01:41:54
						OOD-S	5998	75.88	76.44	74.70	78.26	51.81	01:47:49
						OOD-H	5994	66.80	68.46	65.20	72.07	33.79	01:49:24
						OOD-S'	7966	75.50	76.28	73.91	78.81	51.10	02:16:19
						OOD-H'	10150	67.02	68.71	65.37	72.41	34.25	02:59:57
ID <sup>CoT</sup>	5996	96.28				96.22	97.70	94.80	92.60	02:21:18			
OOD-S <sup>CoT</sup>	5998	78.26				76.04	84.69	68.99	57.52	02:29:46			
OOD-H <sup>CoT</sup>	5994	79.53				77.85	84.82	71.94	59.75	02:26:29			
ID	5998	93.33				93.45	91.82	95.13	86.72	03:53:40			
OOD-S	5992	76.72				78.22	73.48	83.61	53.95	04:38:08			
GPT-4o		GPT-4	OOD-H	6000	70.53	72.06	68.51	76.00	41.31	03:47:18			
			OOD-S'	7958	77.51	78.97	74.16	84.44	55.55	05:35:00			
			OOD-H'	10146	66.51	69.82	63.54	77.47	33.84	07:08:36			
			ID <sup>CoT</sup>	6000	82.92	84.48	77.39	93.00	67.21	06:37:40			
			OOD-S <sup>CoT</sup>	5994	67.52	66.39	68.78	64.16	35.11	06:50:23			
			OOD-H <sup>CoT</sup>	5996	69.31	69.72	68.81	70.65	38.64	05:58:43			

Table 10: **Performance on COCO-ID-OOD.** We report the performance of the 10 leading VLMs on OODBench. All models perform significantly lower on OOD-H than on ID.

Model	Image Encoder	Language Model	Data Type	LVIS-ID-OOD Performance						
				Num	Accuracy(%)	F1(%)	Precision(%)	Recall(%)	MCC(%)	Time
Random Chance	-	-	ID/OOD-S/OOD-H	-	50.00%	50.00%	50.00%	50.00%	0.00%	-
<b>Open-source Models</b>										
LLaVA-NeXT-8B	CLIP-L-14	Llama-3-8B-Instruct	ID	6000	85.17	85.06	85.67	84.47	70.34	00:24:54 <sup>2</sup>
			OOD-S	6000	47.90	35.20	46.55	28.30	-4.57	00:29:54 <sup>2</sup>
			OOD-H	2436	40.23	30.86	36.60	26.68	-20.30	00:10:45 <sup>2</sup>
			OOD-S <sup>COT</sup>	13282	47.10	35.08	45.40	28.58	-6.24	01:49:00 <sup>1</sup>
			ID <sup>COT</sup>	6000	72.45	75.57	67.88	85.23	46.44	00:25:28 <sup>2</sup>
			OOD-S <sup>COT</sup>	6000	49.65	41.62	49.52	35.90	-0.73	00:36:39 <sup>2</sup>
			OOD-H <sup>COT</sup>	2436	48.07	46.47	47.95	45.07	-3.87	00:12:55 <sup>2</sup>
DeepSeek-VL-7B-Chat	SigLIP-L & SAM-B	DeepSeek-LLM-7B	ID	6000	89.85	90.14	87.63	92.80	79.84	00:27:58 <sup>1</sup>
			OOD-S	6000	47.42	41.65	46.78	37.53	-5.27	00:33:43 <sup>1</sup>
			OOD-H	2436	42.69	41.39	42.35	40.48	-14.63	00:08:51 <sup>1</sup>
			OOD-S <sup>COT</sup>	13282	47.33	42.06	46.74	38.23	-5.42	00:49:51 <sup>1</sup>
			ID <sup>COT</sup>	6000	88.42	88.23	89.65	86.87	76.87	01:15:31 <sup>1</sup>
			OOD-S <sup>COT</sup>	6000	46.18	38.51	44.91	33.70	-7.88	01:58:38 <sup>1</sup>
			OOD-H <sup>COT</sup>	2436	35.18	30.16	32.69	28.00	-29.95	00:41:00 <sup>1</sup>
DeepSeek-VL2-Small	SigLIP-SO400M	DeepSeekMoE LLM	ID	6000	87.75	87.13	91.74	82.97	75.85	00:30:31 <sup>1</sup>
			OOD-S	6000	47.23	30.90	44.75	23.60	-6.28	00:29:37 <sup>1</sup>
			OOD-H	2436	39.66	27.08	34.21	22.41	-22.04	00:12:38 <sup>1</sup>
			OOD-S <sup>COT</sup>	13282	46.94	31.50	44.43	24.39	-6.85	01:40:48 <sup>1</sup>
			ID <sup>COT</sup>	6000	82.47	78.82	99.49	65.27	69.15	00:51:51 <sup>1</sup>
			OOD-S <sup>COT</sup>	6000	44.30	15.61	32.19	10.30	-15.55	00:49:32 <sup>1</sup>
			OOD-H <sup>COT</sup>	2436	34.85	8.64	14.45	6.16	-36.99	00:24:25 <sup>1</sup>
InternVL2-8B	InternViT-300M-448px	InternLM2.5-7b-chat	ID	6000	87.20	87.31	86.57	88.07	74.41	00:40:08 <sup>2</sup>
			OOD-S	6000	49.12	46.16	49.01	43.63	-1.78	00:41:49 <sup>2</sup>
			OOD-H	2436	34.56	33.80	34.20	33.42	-30.88	00:16:27 <sup>2</sup>
			OOD-S <sup>COT</sup>	13284	48.34	45.32	48.14	42.82	-3.33	01:32:38 <sup>2</sup>
			ID <sup>COT</sup>	6000	94.55	94.28	99.12	89.90	89.49	01:18:29 <sup>2</sup>
			OOD-S <sup>COT</sup>	6000	48.23	38.32	47.40	32.17	-3.73	01:31:17 <sup>2</sup>
			OOD-H <sup>COT</sup>	2436	43.23	36.47	41.40	32.59	-13.86	00:35:24 <sup>2</sup>
InternVL2.5-8B	InternViT-300M-448px-V2.5	InternLM2.5-7b-chat	ID	6000	94.17	94.09	95.28	92.93	88.36	00:25:50 <sup>2</sup>
			OOD-S	6000	52.87	47.65	53.58	42.90	5.85	00:28:34 <sup>2</sup>
			OOD-H	2436	39.24	36.04	38.05	34.24	-21.62	00:10:55 <sup>2</sup>
			OOD-S <sup>COT</sup>	13284	51.71	46.47	52.12	41.93	3.48	01:03:03 <sup>2</sup>
			ID <sup>COT</sup>	6000	94.52	94.40	96.42	92.47	89.11	01:47:40 <sup>2</sup>
			OOD-S <sup>COT</sup>	6000	52.30	46.24	52.97	41.03	4.72	01:48:51 <sup>2</sup>
			OOD-H <sup>COT</sup>	2436	41.67	37.76	40.47	35.39	-16.80	00:44:34 <sup>2</sup>
Llama-3.2-11B-Vision-Instruct	ViT-H-14	Llama3.1-8B	ID	6000	92.60	92.88	89.54	96.47	85.46	00:57:42 <sup>1</sup>
			OOD-S	6000	50.62	51.70	50.59	52.87	1.23	01:01:18 <sup>1</sup>
			OOD-H	2436	40.23	43.61	41.28	46.22	-19.68	00:23:34 <sup>1</sup>
			OOD-S <sup>COT</sup>	13282	49.99	51.01	49.99	52.07	-0.02	02:14:40 <sup>1</sup>
			ID <sup>COT</sup>	6000	75.57	78.71	69.74	90.33	53.52	02:18:56 <sup>1</sup>
			OOD-S <sup>COT</sup>	6000	49.87	51.31	49.87	52.83	-0.27	02:16:41 <sup>1</sup>
			OOD-H <sup>COT</sup>	2436	41.67	37.76	40.47	35.39	-16.80	00:44:34 <sup>2</sup>
Qwen2-VL-7B-Instruct	CLIP-L-14	Qwen2-7B	ID	6000	86.78	87.63	82.37	93.60	74.26	00:06:51 <sup>2</sup>
			OOD-S	6000	55.60	53.15	56.25	50.37	11.26	00:06:56 <sup>2</sup>
			OOD-H	2436	49.14	50.50	49.18	51.89	-1.73	00:02:50 <sup>2</sup>
			OOD-S <sup>COT</sup>	13284	54.91	52.80	55.39	50.45	9.86	00:15:26 <sup>2</sup>
			ID <sup>COT</sup>	6000	89.03	89.58	85.33	94.27	78.50	00:35:25 <sup>2</sup>
			OOD-S <sup>COT</sup>	6000	56.23	55.69	56.39	55.00	12.47	00:54:50 <sup>2</sup>
			OOD-H <sup>COT</sup>	2436	48.11	49.68	48.22	51.23	-3.78	00:19:20 <sup>2</sup>
Qwen2.5-VL-7B-Instruct	ViT(SwiGLU, RMSNorm)	Qwen2.5 LLM	ID	6000	93.03	92.81	95.94	89.87	86.24	00:11:14 <sup>1</sup>
			OOD-S	6000	52.63	41.43	54.27	33.50	5.70	00:10:55 <sup>1</sup>
			OOD-H	2436	46.31	39.67	45.26	35.30	-7.57	00:04:20 <sup>1</sup>
			OOD-S <sup>COT</sup>	13284	52.29	41.66	53.60	34.08	4.92	00:23:43 <sup>1</sup>
			ID <sup>COT</sup>	6000	92.70	92.13	99.92	85.47	86.31	01:31:56 <sup>1</sup>
			OOD-S <sup>COT</sup>	6000	51.38	35.05	52.78	26.23	3.20	01:48:30 <sup>1</sup>
			OOD-H <sup>COT</sup>	2436	43.19	32.36	39.98	27.18	-14.39	00:46:05 <sup>1</sup>
<b>Closed-source Models</b>										
Gemini	-	-	ID	5996	87.68	88.13	84.98	91.53	75.57	01:41:01
			OOD-S	5994	51.58	52.05	51.55	52.55	3.17	01:47:55
			OOD-H	2436	35.18	37.47	36.19	38.83	-29.72	00:42:07
			OOD-S <sup>COT</sup>	13272	50.63	51.49	50.61	52.40	1.27	03:51:01
			ID <sup>COT</sup>	5996	92.48	92.42	93.15	91.69	84.97	02:21:13
			OOD-S <sup>COT</sup>	5994	50.10	43.41	50.13	38.27	0.21	02:38:14
			OOD-H <sup>COT</sup>	2436	40.72	37.27	39.58	35.22	-18.67	01:03:47
GPT-4o	-	GPT-4	ID	5960	93.09	93.26	91.00	95.64	86.29	05:46:52
			OOD-S	5990	58.30	60.15	57.59	62.94	16.67	04:29:12
			OOD-H	2432	41.69	45.08	42.61	47.86	-16.74	01:45:24
			OOD-S <sup>COT</sup>	13282	57.90	58.53	57.66	59.43	15.80	08:40:59
			ID <sup>COT</sup>	6000	81.10	83.09	75.16	92.90	64.01	06:35:48
			OOD-S <sup>COT</sup>	5958	51.88	48.98	52.12	46.19	3.78	08:42:01
			OOD-H <sup>COT</sup>	2392	45.07	46.76	45.36	48.24	-9.89	04:43:32

Table 11: **Performance on LVIS-ID-OOD.** We report the performance of the **10** leading VLMs on OODBench. All models perform significantly lower on OOD-H than on ID.

2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258  
2259  
2260  
2261  
2262  
2263  
2264  
2265  
2266  
2267

Model	Image Encoder	Language Model	Data Type	nuScenes-ID-OOD Performance						
				Num	Accuracy(%)	F1(%)	Precision(%)	Recall(%)	MCC(%)	Time
Random Chance	-	-	ID/OOD-S/OOD-H	-	50.00%	50.00%	50.00%	50.00%	0.00%	-
<b>Open-source Models</b>										
			ID	6000	81.43	80.05	86.49	74.50	63.48	00:27:02 <sup>2</sup>
			OOD-S	6000	62.68	45.38	84.62	31.00	32.79	00:31:30 <sup>2</sup>
			OOD-H	6000	59.03	40.34	74.20	27.70	23.18	00:30:31 <sup>2</sup>
			OOD-S <sup>†</sup>	25798	62.57	45.42	83.83	31.15	32.32	03:45:13 <sup>1</sup>
			OOD-H <sup>†</sup>	9882	60.58	42.24	79.02	28.82	27.41	01:23:29 <sup>1</sup>
			ID <sup>CoT</sup>	6000	50.90	58.48	50.66	69.17	1.93	01:11:18 <sup>1</sup>
			OOD-S <sup>CoT</sup>	6000	52.12	41.14	53.38	33.47	4.56	01:06:26 <sup>2</sup>
			OOD-H <sup>CoT</sup>	6000	54.87	42.58	58.51	33.47	10.77	00:57:57 <sup>2</sup>
			ID	6000	83.40	82.09	89.14	76.07	67.53	00:33:53 <sup>1</sup>
			OOD-S	6000	61.12	47.16	73.57	34.70	26.19	00:34:22 <sup>1</sup>
			OOD-H	6000	52.95	28.77	59.19	19.00	8.04	00:23:41 <sup>1</sup>
			OOD-S <sup>†</sup>	25798	60.47	46.13	72.38	33.85	24.73	02:03:01 <sup>1</sup>
			OOD-H <sup>†</sup>	9882	54.62	30.33	65.28	19.75	12.91	00:55:52 <sup>1</sup>
			ID <sup>CoT</sup>	6000	74.20	71.15	80.68	63.63	49.52	02:20:00 <sup>1</sup>
			OOD-S <sup>CoT</sup>	6000	59.18	42.66	71.68	30.37	22.47	03:04:27 <sup>1</sup>
			OOD-H <sup>CoT</sup>	6000	46.30	24.54	41.26	17.47	-9.06	02:44:52 <sup>1</sup>
			ID	6000	76.08	70.42	92.27	56.93	56.47	00:43:54 <sup>1</sup>
			OOD-S	6000	58.73	32.64	88.76	20.00	27.62	00:41:44 <sup>1</sup>
			OOD-H	6000	51.72	15.81	61.68	9.07	6.58	00:44:10 <sup>1</sup>
			OOD-S <sup>†</sup>	25798	58.09	30.67	88.69	18.54	26.44	03:27:02 <sup>1</sup>
			OOD-H <sup>†</sup>	9882	52.74	16.49	70.81	9.33	11.05	01:32:27 <sup>1</sup>
			ID <sup>CoT</sup>	6000	77.77	71.59	99.12	56.03	61.66	00:55:41 <sup>1</sup>
			OOD-S <sup>CoT</sup>	6000	58.73	31.90	91.19	19.33	28.37	01:01:12 <sup>1</sup>
			OOD-H <sup>CoT</sup>	6000	51.05	11.56	59.81	6.40	4.67	01:01:55 <sup>1</sup>
			ID	6000	89.77	89.13	95.09	83.87	80.09	00:29:17 <sup>2</sup>
			OOD-S	6000	71.27	68.14	76.45	61.47	43.37	00:31:04 <sup>2</sup>
			OOD-H	6000	51.08	43.20	51.50	37.20	2.26	00:30:46 <sup>2</sup>
			OOD-S <sup>†</sup>	25800	70.90	67.60	76.26	60.71	42.70	02:12:54 <sup>2</sup>
			OOD-H <sup>†</sup>	9884	52.08	44.57	52.86	38.53	4.33	00:50:14 <sup>2</sup>
			ID <sup>CoT</sup>	6000	90.20	89.30	98.32	81.80	81.56	03:56:47 <sup>2</sup>
			OOD-S <sup>CoT</sup>	6000	64.73	50.24	85.30	35.60	36.26	04:20:00 <sup>2</sup>
			OOD-H <sup>CoT</sup>	6000	62.23	45.48	81.75	31.50	31.02	04:24:46 <sup>2</sup>
			ID	6000	82.70	83.75	78.96	89.17	65.95	00:23:12 <sup>2</sup>
			OOD-S	6000	70.13	71.02	68.97	73.20	40.34	00:24:38 <sup>2</sup>
			OOD-H	6000	36.23	39.61	37.62	41.83	-27.71	00:24:29 <sup>2</sup>
			OOD-S <sup>†</sup>	25800	69.65	70.57	68.50	72.78	39.38	01:45:43 <sup>2</sup>
			OOD-H <sup>†</sup>	9884	36.68	39.90	37.97	42.05	-26.80	00:40:22 <sup>2</sup>
			ID <sup>CoT</sup>	6000	78.00	79.16	75.19	83.57	56.35	02:53:33 <sup>2</sup>
			OOD-S <sup>CoT</sup>	6000	65.18	57.43	73.89	46.97	32.61	03:21:22 <sup>2</sup>
			OOD-H <sup>CoT</sup>	6000	54.82	48.99	56.24	43.40	9.89	03:24:02 <sup>2</sup>
			ID	6000	88.78	88.74	89.06	88.43	77.57	01:11:40 <sup>1</sup>
			OOD-S	6000	69.58	64.56	77.34	55.40	40.84	01:15:19 <sup>1</sup>
			OOD-H	6000	61.72	60.02	62.81	57.47	23.52	01:18:18 <sup>1</sup>
			OOD-S <sup>†</sup>	25798	66.30	63.34	69.43	58.24	33.02	04:51:10 <sup>1</sup>
			OOD-H <sup>†</sup>	9882	58.86	60.28	58.28	62.42	17.77	01:51:33 <sup>1</sup>
			ID <sup>CoT</sup>	6000	71.63	72.33	70.60	74.13	43.32	02:34:13 <sup>1</sup>
			OOD-S <sup>CoT</sup>	6000	59.33	49.90	64.97	40.50	20.15	02:27:42 <sup>1</sup>
			OOD-H <sup>CoT</sup>	6000	59.98	55.56	62.46	50.03	20.37	02:29:42 <sup>1</sup>
			ID	6000	92.72	92.58	94.39	90.83	85.49	00:50:55 <sup>2</sup>
			OOD-S	6000	73.53	69.84	81.16	61.30	48.54	00:51:02 <sup>2</sup>
			OOD-H	6000	57.55	50.05	60.79	42.53	15.83	00:50:54 <sup>2</sup>
			OOD-S <sup>†</sup>	25800	73.12	69.32	80.75	60.72	47.74	03:40:30 <sup>2</sup>
			OOD-H <sup>†</sup>	9884	59.02	51.87	62.85	44.15	18.91	01:24:18 <sup>2</sup>
			ID <sup>CoT</sup>	6000	87.82	88.04	86.47	89.67	75.69	01:35:02 <sup>2</sup>
			OOD-S <sup>CoT</sup>	6000	68.58	59.95	82.66	47.03	41.19	02:02:32 <sup>2</sup>
			OOD-H <sup>CoT</sup>	6000	57.13	47.27	61.40	38.43	15.38	02:02:22 <sup>2</sup>
			ID	6000	80.62	81.20	78.82	83.73	61.35	01:09:27 <sup>1</sup>
			OOD-S	6000	63.77	51.17	78.44	37.97	32.14	00:50:18 <sup>1</sup>
			OOD-H	6000	60.12	46.02	71.18	34.00	23.73	00:59:19 <sup>1</sup>
			OOD-S <sup>†</sup>	25800	63.54	50.57	78.51	37.30	31.82	03:19:44 <sup>1</sup>
			OOD-H <sup>†</sup>	9884	61.72	47.97	74.85	35.30	27.61	01:20:31 <sup>1</sup>
			ID <sup>CoT</sup>	6000	91.20	90.35	100.00	82.40	83.71	02:05:30 <sup>1</sup>
			OOD-S <sup>CoT</sup>	6000	66.52	51.83	92.31	36.03	41.67	02:06:14 <sup>1</sup>
			OOD-H <sup>CoT</sup>	6000	63.02	45.28	87.01	30.60	34.19	02:20:17 <sup>1</sup>
<b>Closed-source Models</b>										
			ID	6000	90.40	90.48	89.76	91.20	80.81	02:06:56
			OOD-S	6000	77.25	76.71	78.57	74.93	54.56	02:15:23
			OOD-H	6000	56.95	56.74	57.02	56.47	13.90	02:17:32
			OOD-S <sup>†</sup>	25798	77.19	76.68	78.42	75.03	54.43	08:43:47
			OOD-H <sup>†</sup>	9882	59.07	59.13	59.04	59.22	18.13	03:25:56
			ID <sup>CoT</sup>	6000	92.70	92.55	94.45	90.73	85.47	02:57:49
			OOD-S <sup>CoT</sup>	6000	68.55	62.16	78.01	51.67	39.42	03:03:12
			OOD-H <sup>CoT</sup>	6000	68.93	64.36	75.47	56.10	39.18	02:58:06
			ID	5996	86.72	87.72	81.60	94.83	74.43	05:14:11
			OOD-S	5994	74.94	74.55	75.73	73.41	49.91	05:51:09
			OOD-H	5998	58.92	61.29	57.94	65.06	17.98	05:48:40
			OOD-S <sup>†</sup>	25798	75.83	75.62	76.27	74.98	51.66	24:39:04
			OOD-H <sup>†</sup>	9882	58.78	61.58	57.67	66.06	17.76	09:52:58
			ID <sup>CoT</sup>	6000	70.83	75.76	64.81	91.17	45.61	07:48:32
			OOD-S <sup>CoT</sup>	5998	62.70	58.41	66.01	52.38	25.97	08:11:22
			OOD-H <sup>CoT</sup>	5992	61.35	60.50	61.85	59.21	22.72	07:21:03

Table 12: **Performance on nuScenes-ID-OOD.** We report the performance of the 10 leading VLMs on OODBench. All models perform significantly lower on OOD-H than on ID.

Model	Image Encoder	Language Model	Data Type	Cityscapes-ID-OOD Performance						
				Num	Accuracy(%)	F1(%)	Precision(%)	Recall(%)	MCC(%)	Time
Random Chance	-	-	ID/OOD-S/OOD-H	-	50.00%	50.00%	50.00%	50.00%	0.00%	-
<b>Open-source Models</b>										
LLaVA-NeXT-8B	CLIP-L-14	Llama-3-8B-Instruct	ID	3556	81.21	82.14	78.29	86.39	62.77	00:18:39 <sup>2</sup>
			OOD-S	3820	78.22	76.55	82.91	71.10	57.02	00:17:43 <sup>2</sup>
			OOD-H	3836	66.79	58.09	78.70	46.04	36.90	00:18:58 <sup>2</sup>
			ID <sup>COT</sup>	3554	59.23	69.67	55.47	93.64	25.44	00:32:01 <sup>1</sup>
			OOD-S <sup>COT</sup>	3820	65.58	70.01	62.02	80.37	32.61	00:20:07 <sup>2</sup>
			OOD-H <sup>COT</sup>	3836	70.52	65.47	79.00	55.89	42.91	00:19:56 <sup>2</sup>
DeepSeek-VL-7B-Chat	SigLIP-L & SAM-B	DeepSeek-LLM-7B	ID	3554	91.73	91.37	95.52	87.56	83.75	00:25:48 <sup>1</sup>
			OOD-S	3818	78.71	75.95	87.23	67.26	58.98	00:27:57 <sup>1</sup>
			OOD-H	3834	61.53	54.99	66.25	47.00	24.10	00:26:57 <sup>1</sup>
			ID <sup>COT</sup>	3554	85.45	85.18	86.80	83.62	70.95	00:49:00 <sup>1</sup>
			OOD-S <sup>COT</sup>	3818	77.71	75.33	84.35	68.05	56.49	00:51:54 <sup>1</sup>
			OOD-H <sup>COT</sup>	3834	63.43	54.60	71.99	43.97	29.16	01:25:13 <sup>1</sup>
DeepSeek-VL2-Small	SigLIP-SO400M	DeepSeekMoE LLM	ID	3554	80.95	76.55	99.55	62.18	66.79	00:32:18 <sup>1</sup>
			OOD-S	3818	77.42	71.46	97.12	56.52	60.37	00:30:48 <sup>1</sup>
			OOD-H	3834	65.39	47.49	98.36	31.30	42.07	00:31:18 <sup>1</sup>
			ID <sup>COT</sup>	3554	79.07	73.52	100.00	58.13	64.01	00:42:59 <sup>1</sup>
			OOD-S <sup>COT</sup>	3818	70.77	59.80	95.73	43.48	49.58	00:52:23 <sup>1</sup>
			OOD-H <sup>COT</sup>	3834	57.90	27.88	97.20	16.28	28.53	00:50:00 <sup>1</sup>
InternVL2-8B	InternViT-300M-448px	InternLM2.5-7b-chat	ID	3556	95.95	95.82	98.98	92.86	92.08	00:17:34 <sup>2</sup>
			OOD-S	3820	86.94	86.20	91.38	81.57	74.30	00:21:15 <sup>2</sup>
			OOD-H	3836	77.48	74.78	84.95	66.79	56.25	00:19:21 <sup>2</sup>
			ID <sup>COT</sup>	3556	90.83	89.94	99.59	82.00	82.97	02:12:08 <sup>2</sup>
			OOD-S <sup>COT</sup>	3820	82.38	79.44	95.38	68.06	67.60	02:23:47 <sup>2</sup>
			OOD-H <sup>COT</sup>	3836	77.69	71.66	98.19	56.41	61.18	02:27:58 <sup>2</sup>
InternVL2.5-8B	InternViT-300M-448px-V2.5	InternLM2.5-7b-chat	ID	3556	93.53	93.61	92.43	94.83	87.09	00:14:58 <sup>2</sup>
			OOD-S	3820	86.88	86.95	86.52	87.38	73.77	00:16:55 <sup>2</sup>
			OOD-H	3836	72.34	71.07	74.50	67.94	44.86	00:15:46 <sup>2</sup>
			ID <sup>COT</sup>	3556	93.98	93.95	94.38	93.53	87.97	01:46:20 <sup>2</sup>
			OOD-S <sup>COT</sup>	3820	85.73	84.80	90.75	79.58	72.01	01:55:57 <sup>2</sup>
			OOD-H <sup>COT</sup>	3836	79.25	76.99	86.38	69.45	59.66	01:58:27 <sup>2</sup>
Llama-3.2-11B-Vision-Instruct	ViT-H-14	Llama3.1-8B	ID	3554	92.40	92.70	89.22	96.45	85.09	00:45:22 <sup>1</sup>
			OOD-S	3818	73.21	76.24	68.49	85.96	48.00	00:43:55 <sup>1</sup>
			OOD-H	3834	65.26	69.41	62.00	78.82	31.71	00:41:42 <sup>1</sup>
			ID <sup>COT</sup>	3554	71.05	70.41	72.00	68.88	42.13	01:32:09 <sup>1</sup>
			OOD-S <sup>COT</sup>	3818	64.64	62.42	66.61	58.72	29.49	01:37:42 <sup>1</sup>
			OOD-H <sup>COT</sup>	3834	61.22	61.29	61.17	61.40	22.43	01:34:27 <sup>1</sup>
Qwen2-VL-7B-Instruct	CLIP-L-14	Qwen2-7B	ID	3556	95.78	95.69	97.88	93.59	91.65	01:16:54 <sup>2</sup>
			OOD-S	3820	86.10	85.58	88.89	82.51	72.39	01:22:25 <sup>2</sup>
			OOD-H	3836	75.36	72.42	82.24	64.70	51.92	01:22:52 <sup>2</sup>
			ID <sup>COT</sup>	3556	95.73	95.67	96.94	94.43	91.48	01:37:58 <sup>2</sup>
			OOD-S <sup>COT</sup>	3820	83.53	83.24	84.75	81.78	67.11	01:34:26 <sup>2</sup>
			OOD-H <sup>COT</sup>	3836	70.62	68.77	73.39	64.70	41.53	01:47:12 <sup>2</sup>
Qwen2.5-VL-7B-Instruct	ViT(SwiGLU, RMSNorm)	Qwen2.5 LLM	ID	3556	92.82	92.28	99.87	85.76	86.52	00:44:03 <sup>1</sup>
			OOD-S	3820	86.30	84.69	96.02	75.75	74.28	00:51:29 <sup>1</sup>
			OOD-H	3836	81.35	77.39	98.23	63.85	66.94	00:47:35 <sup>1</sup>
			ID <sup>COT</sup>	3556	94.15	93.78	100.00	88.29	88.91	02:01:18 <sup>1</sup>
			OOD-S <sup>COT</sup>	3820	84.65	82.56	95.59	72.66	71.39	02:12:09 <sup>1</sup>
			OOD-H <sup>COT</sup>	3836	81.61	77.61	99.19	63.75	67.69	02:04:44 <sup>1</sup>
<b>Closed-source Models</b>										
Gemini	-	-	ID	3554	96.88	96.88	96.75	97.02	93.75	03:11:49
			OOD-S	3818	88.45	88.48	88.27	88.69	76.90	03:35:00
			OOD-H	3834	86.78	86.96	85.79	88.16	73.58	03:38:15
			ID <sup>COT</sup>	3554	97.16	97.14	97.89	96.40	94.33	03:32:09
			OOD-S <sup>COT</sup>	3818	86.90	85.96	92.62	80.20	74.48	03:42:41
			OOD-H <sup>COT</sup>	3834	91.78	91.40	95.88	87.32	83.90	03:42:08
GPT-4o	-	GPT-4	ID	3552	96.54	96.59	95.09	98.14	93.12	07:12:17
			OOD-S	3800	89.34	89.48	88.35	90.63	78.71	07:51:47
			OOD-H	3830	81.28	82.32	77.99	87.15	63.00	07:21:43
			ID <sup>COT</sup>	3514	77.32	80.81	70.03	95.50	58.66	10:25:24
			OOD-S <sup>COT</sup>	3816	70.55	74.68	65.48	86.90	43.48	08:13:24
			OOD-H <sup>COT</sup>	3824	69.87	73.36	65.75	82.95	41.18	09:15:38

Table 13: **Performance on Cityscapes-ID-OOD.** We report the performance of the **10** leading VLMs on OODBench. All models perform significantly lower on OOD-H than on ID.

**Subdataset BAP performance.** Tab. 14, 15, 16, and 17 show the evaluation results of the BAP metrics on four datasets, namely COCO, LVIS, nuScenes, and Cityscapes, respectively. The BAP metrics progressively increase the complexity of the questions from the perspective of task construction, and consist of Existence, Counting, and Logic Reasoning, with the corresponding performance evaluation metrics of E-Acc, C-Acc, and L-Acc, respectively.

The results of the four datasets show a consistent trend. Under the same question type, as the data distribution migrates from ID to OOD-S to OOD-H, the model performance generally decreases. This phenomenon indicates that the performance of the multimodal large language model degrades significantly when the data distribution deviates from the training distribution, reflecting that its generalization ability in dealing with complex environments is still limited.

On the other hand, while keeping the test data unchanged, when the complexity of questioning is gradually increased from Existence to Counting to Logic Reasoning, the accuracy of the model on the three corresponding metrics (E-Acc, C-Acc, and L-Acc) also shows a decreasing trend, which indicates that the higher the requirement of reasoning ability is, the more difficult it is for the model to answer correctly.

Specifically, Tab. 14 to 17 systematically show the performance of major mainstream VLMs on the four datasets of COCO, LVIS, nuScenes, and Cityscapes. The results are generally consistent, further validating the generality of the two trends mentioned above. This indicates that current VLMs still have significant performance bottlenecks when facing more challenging inference tasks and more complex data distributions.

Model	Data Type	COCO BAP Performance			Time
		E-Acc(%)	C-Acc(%)	L-Acc(%)	
LLaVA-NeXT-8B	ID	88.04	61.72	37.08	00:03:48 <sup>2</sup>
	OOD-S	79.98	47.40	33.23	00:10:14 <sup>2</sup>
	OOD-H	55.85	33.83	21.10	00:09:47 <sup>2</sup>
DeepSeek-VL-7B-Chat	ID	91.39	58.61	44.26	00:02:58 <sup>2</sup>
	OOD-S	80.30	36.26	27.81	00:06:47 <sup>2</sup>
	OOD-H	60.89	33.26	23.17	00:06:16 <sup>2</sup>
DeepSeek-VL2-Small	ID	85.85	55.40	37.65	00:08:53 <sup>1</sup>
	OOD-S	73.89	41.28	29.14	00:23:00 <sup>1</sup>
	OOD-H	49.77	34.86	23.74	00:22:08 <sup>1</sup>
InternVL2-8B	ID	88.04	69.14	57.18	00:05:34 <sup>2</sup>
	OOD-S	79.87	53.57	41.45	00:12:32 <sup>2</sup>
	OOD-H	59.63	49.43	36.35	00:11:26 <sup>2</sup>
InternVL2.5-8B	ID	94.02	67.94	53.83	00:04:35 <sup>2</sup>
	OOD-S	85.93	59.20	47.62	00:11:04 <sup>2</sup>
	OOD-H	70.87	49.31	38.76	00:10:10 <sup>2</sup>
Llama-3.2-11B-Vision-Instruct	ID	96.65	66.99	51.20	00:04:26 <sup>2</sup>
	OOD-S	86.90	56.17	35.61	00:10:17 <sup>2</sup>
	OOD-H	86.93	48.17	31.08	00:09:59 <sup>2</sup>
Qwen2-VL-7B-Instruct	ID	91.87	65.55	48.56	00:02:13 <sup>2</sup>
	OOD-S	81.82	62.12	48.16	00:05:09 <sup>2</sup>
	OOD-H	66.06	45.53	33.60	00:05:39 <sup>2</sup>
Qwen2.5-VL-7B-Instruct	ID	89.21	66.19	54.44	00:06:01 <sup>1</sup>
	OOD-S	78.11	59.05	45.50	00:15:26 <sup>1</sup>
	OOD-H	58.83	47.36	35.78	00:15:06 <sup>1</sup>
Gemini	ID	88.49	75.78	42.21	02:57:38
	OOD-S	81.47	69.45	48.43	00:59:56
	OOD-H	66.97	52.29	34.40	03:54:16
gpt-4o	ID	90.41	66.67	36.93	00:45:11
	OOD-S	85.70	58.07	42.90	01:44:58
	OOD-H	74.77	48.17	33.72	01:45:12

Table 14: Experimental performance of 10 leading VLMs on COCO for Basic-to-Advanced Progress.

2376

2377

2378

2379

2380

2381

2382

2383

2384

2385

2386

2387

2388

2389

2390

2391

2392

2393

2394

2395

2396

2397

2398

2399

2400

2401

2402

2403

2404

2405

2406

2407

2408

2409

2410

2411

2412

2413

2414

2415

2416

2417

2418

2419

2420

2421

2422

2423

2424

2425

2426

2427

2428

2429

Model	Data Type	LVIS BAP Performance			
		E-Acc(%)	C-Acc(%)	L-Acc(%)	Time
LLaVA-NeXT-8B	ID	81.27	52.07	29.84	00:07:14 <sup>2</sup>
	OOD-S	49.04	35.01	17.39	00:10:15 <sup>2</sup>
	OOD-H	31.25	27.73	14.06	00:02:45 <sup>2</sup>
DeepSeek-VL-7B-Chat	ID	90.44	55.04	39.79	00:05:38 <sup>2</sup>
	OOD-S	53.36	22.66	16.19	00:06:24 <sup>2</sup>
	OOD-H	50.78	23.05	13.67	00:01:56 <sup>2</sup>
DeepSeek-VL2-Small	ID	77.23	43.86	28.07	00:23:36 <sup>1</sup>
	OOD-S	51.14	34.21	19.93	00:25:16 <sup>1</sup>
	OOD-H	29.41	26.67	12.94	00:08:02 <sup>1</sup>
InternVL2-8B	ID	85.14	65.37	48.32	00:11:28 <sup>2</sup>
	OOD-S	55.16	35.97	24.82	00:12:22 <sup>2</sup>
	OOD-H	42.19	37.89	28.12	00:03:34 <sup>2</sup>
InternVL2.5-8B	ID	90.18	61.50	46.51	00:08:46 <sup>2</sup>
	OOD-S	56.12	38.13	25.06	00:10:41 <sup>2</sup>
	OOD-H	48.83	36.72	21.48	00:03:12 <sup>2</sup>
Llama-3.2-11B-Vision-Instruct	ID	96.12	62.02	41.99	00:08:27 <sup>2</sup>
	OOD-S	59.11	37.17	25.78	00:09:29 <sup>2</sup>
	OOD-H	62.50	32.42	21.88	00:02:56 <sup>2</sup>
Qwen2-VL-7B-Instruct	ID	91.60	63.95	45.48	00:04:38 <sup>2</sup>
	OOD-S	57.07	44.36	28.78	00:05:36 <sup>2</sup>
	OOD-H	51.95	31.64	23.44	00:01:35 <sup>2</sup>
Qwen2.5-VL-7B-Instruct	ID	88.75	59.38	43.73	00:11:42 <sup>1</sup>
	OOD-S	60.98	39.98	26.05	00:15:16 <sup>1</sup>
	OOD-H	54.51	32.16	19.61	00:04:16 <sup>1</sup>
Gemini	ID	89.65	70.25	37.39	00:49:55
	OOD-S	57.21	50.72	25.72	04:56:20
	OOD-H	54.12	41.57	23.53	00:17:33
gpt-4o	ID	94.44	63.39	39.07	01:29:08
	OOD-S	63.03	41.54	21.37	01:39:38
	OOD-H	58.43	35.29	16.08	00:29:38

Table 15: Experimental performance of 10 leading VLMs on LVIS for Basic-to-Advanced Progress.

2403

2404

2405

2406

2407

2408

2409

2410

2411

2412

2413

2414

2415

2416

2417

2418

2419

2420

2421

2422

2423

2424

2425

2426

2427

2428

2429

Model	Data Type	nuScenes BAP Performance			
		E-Acc(%)	C-Acc(%)	L-Acc(%)	Time
LLaVA-NeXT-8B	ID	68.31	21.38	15.54	00:08:03 <sup>2</sup>
	OOD-S	65.15	40.88	28.65	00:07:10 <sup>2</sup>
	OOD-H	35.10	20.65	13.27	00:08:46 <sup>2</sup>
DeepSeek-VL-7B-Chat	ID	68.46	30.77	13.08	00:05:05 <sup>2</sup>
	OOD-S	59.85	34.67	19.89	00:04:29 <sup>2</sup>
	OOD-H	27.14	22.57	15.34	00:05:23 <sup>2</sup>
DeepSeek-VL2-Small	ID	56.39	5.39	3.54	00:22:42 <sup>1</sup>
	OOD-S	57.04	43.88	22.30	00:25:28 <sup>1</sup>
	OOD-H	20.21	22.71	10.18	00:28:18 <sup>1</sup>
InternVL2-8B	ID	82.15	38.92	25.85	00:10:03 <sup>2</sup>
	OOD-S	69.89	41.42	25.73	00:07:42 <sup>2</sup>
	OOD-H	44.99	41.30	28.76	00:09:43 <sup>2</sup>
InternVL2.5-8B	ID	85.69	38.92	31.54	00:05:56 <sup>2</sup>
	OOD-S	75.73	45.26	34.85	00:05:11 <sup>2</sup>
	OOD-H	46.02	35.10	24.78	00:06:36 <sup>2</sup>
Llama-3.2-11B-Vision-Instruct	ID	87.38	37.69	19.54	00:08:15 <sup>2</sup>
	OOD-S	77.37	51.46	23.54	00:07:08 <sup>2</sup>
	OOD-H	58.11	28.47	17.11	00:08:38 <sup>2</sup>
Qwen2-VL-7B-Instruct	ID	88.00	34.00	24.62	00:08:47 <sup>2</sup>
	OOD-S	75.73	50.36	40.51	00:07:29 <sup>2</sup>
	OOD-H	47.49	36.28	25.07	00:09:10 <sup>2</sup>
Qwen2.5-VL-7B-Instruct	ID	81.51	32.05	23.42	00:22:13 <sup>1</sup>
	OOD-S	72.39	44.79	35.47	00:18:53 <sup>1</sup>
	OOD-H	44.69	35.40	25.22	00:22:51 <sup>1</sup>
Gemini	ID	86.44	47.15	35.75	00:55:20
	OOD-S	79.71	53.38	40.95	00:43:04
	OOD-H	62.09	46.31	29.94	00:55:34
gpt-4o	ID	92.91	36.21	29.12	01:53:19
	OOD-S	82.97	47.07	36.08	01:34:33
	OOD-H	68.83	38.26	26.88	01:55:43

Table 16: Experimental performance of 10 leading VLMs on nuScenes for Basic-to-Advanced Progress.

2430  
2431  
2432  
2433  
2434  
2435  
2436  
2437  
2438  
2439  
2440  
2441  
2442  
2443  
2444  
2445  
2446  
2447  
2448  
2449  
2450  
2451  
2452  
2453  
2454  
2455  
2456  
2457  
2458  
2459  
2460  
2461  
2462  
2463  
2464  
2465  
2466  
2467  
2468  
2469  
2470  
2471  
2472  
2473  
2474  
2475  
2476  
2477  
2478  
2479  
2480  
2481  
2482  
2483

Model	Data Type	Cityscapes BAP Performance			
		E-Acc(%)	C-Acc(%)	L-Acc(%)	Time
LLaVA-NeXT-8B	ID	87.37	28.44	12.73	00:11:48 <sup>2</sup>
	OOD-S	69.53	17.17	10.75	00:13:41 <sup>2</sup>
	OOD-H	43.62	10.86	7.05	00:12:38 <sup>2</sup>
DeepSeek-VL-7B-Chat	ID	87.99	22.38	11.19	00:08:56 <sup>2</sup>
	OOD-S	70.19	15.19	8.21	00:09:51 <sup>2</sup>
	OOD-H	45.33	12.95	6.86	00:09:43 <sup>2</sup>
DeepSeek-VL2-Small	ID	68.76	8.12	2.67	00:50:22 <sup>1</sup>
	OOD-S	67.64	12.08	4.43	01:07:38 <sup>1</sup>
	OOD-H	41.09	5.15	3.43	00:45:52 <sup>1</sup>
InternVL2-8B	ID	92.92	40.04	16.22	00:15:03 <sup>2</sup>
	OOD-S	83.30	31.98	15.57	00:27:59 <sup>2</sup>
	OOD-H	65.43	32.38	21.71	00:18:06 <sup>2</sup>
InternVL2.5-8B	ID	94.87	27.93	12.63	00:10:02 <sup>2</sup>
	OOD-S	85.94	17.83	12.92	00:11:15 <sup>2</sup>
	OOD-H	67.14	23.71	18.19	00:10:22 <sup>2</sup>
Llama-3.2-11B-Vision-Instruct	ID	97.23	33.47	13.86	00:13:29 <sup>2</sup>
	OOD-S	85.38	18.87	7.92	00:15:05 <sup>2</sup>
	OOD-H	77.24	19.62	8.29	00:15:06 <sup>2</sup>
Qwen2-VL-7B-Instruct	ID	94.25	31.72	13.35	00:33:15 <sup>2</sup>
	OOD-S	83.77	19.43	11.98	00:35:41 <sup>2</sup>
	OOD-H	64.38	22.00	15.14	00:35:37 <sup>2</sup>
Qwen2.5-VL-7B-Instruct	ID	86.43	22.40	10.69	00:50:03 <sup>1</sup>
	OOD-S	79.53	18.58	10.85	00:57:08 <sup>1</sup>
	OOD-H	62.73	21.54	15.54	00:54:25 <sup>1</sup>
Gemini	ID	97.43	48.92	31.14	02:30:39
	OOD-S	89.91	37.74	24.72	03:00:34
	OOD-H	87.70	32.98	22.12	03:46:30
gpt-4o	ID	98.36	36.38	22.82	05:50:51
	OOD-S	90.85	23.96	15.85	06:37:16
	OOD-H	86.94	24.59	16.97	06:31:57

Table 17: Experimental performance of **10** leading VLMs on **Cityscapes** for Basic-to-Advanced Progress.

## G EFFECT OF MODEL SCALE ON OODBENCH PERFORMANCE

To further analyze the performance of models of different scales on OODBench under the same architecture, we conducted a systematic comparison of different parameter scales within the same model family. Specifically, we selected the representative Qwen2-VL series. We performed experiments on the complete sets and subsets of the four domains covered by OODBench (COCO, LVIS, nuScenes, Cityscapes) for its 2B and 7B versions. The results are shown in Table 18: The 2B model achieved accuracy scores of 80.6, 64.0, and 81.2 on the OOD-H metrics for COCO, nuScenes, and Cityscapes, respectively, all surpassing the 7B model’s scores of 77.7, 57.6, and 75.4. However, on the LVIS dataset, the 7B model slightly outperformed the 2B model. These results reveal a distinct mixed trend, **indicating that model size does not maintain a monotonically increasing relationship with OOD robustness. Increasing parameter size does not consistently enhance model performance under semantic OOD conditions.**

This phenomenon is further validated in larger-scale models. For instance, as shown in Tables 1, 10, 11, 12, and 13, models like GPT-4o and Gemini, which have significantly more parameters than 7B, do not systematically outperform smaller-scale models on OODBench. Overall, expanding model scale cannot fundamentally resolve semantic OOD issues. The root cause lies in the fact that when test samples deviate from the support domain of training data in the joint image-text distribution, their OOD characteristics remain independent of model scale and thus do not naturally disappear with increased capacity.

Model	Data Type	OODBench Performance					
		Num	Accuracy(%)	F1(%)	Precision(%)	Recall(%)	MCC(%)
Random Chance	-	-	50.00	50.00	50.00	50.00	0.00
<b>COCO</b>							
Qwen2-VL-2B	OOD-S	6000	78.78	74.62	92.85	62.37	60.95
	OOD-H	6000	80.63	76.53	97.13	63.13	65.40
	OOD-S'	7972	78.84	74.61	93.23	62.19	61.17
	OOD-H'	10160	81.81	78.13	97.95	64.98	67.56
Qwen2-VL-7B	OOD-S	6000	81.57	79.98	87.52	73.63	63.94
	OOD-H	6000	77.67	75.48	83.66	68.77	56.23
	OOD-S'	7972	81.18	79.49	87.35	72.93	63.24
	OOD-H'	10160	78.74	76.78	84.58	70.33	58.32
<b>LVIS</b>							
Qwen2-VL-2B	OOD-S	6000	48.23	34.91	47.01	27.77	-3.87
	OOD-H	2436	41.67	32.69	38.63	28.33	-17.29
	OOD-S'	13284	47.70	35.03	46.24	28.19	-4.99
Qwen2-VL-7B	OOD-S	6000	55.60	53.15	56.25	50.37	11.26
	OOD-H	2436	49.14	50.50	49.18	51.89	-1.73
	OOD-S'	13284	54.91	52.80	55.39	50.45	9.86
<b>nuScenes</b>							
Qwen2-VL-2B	OOD-S	6000	68.00	57.62	85.29	43.50	41.30
	OOD-H	6000	64.00	50.21	81.39	36.30	33.63
	OOD-S'	25800	67.30	56.76	83.77	42.92	39.64
	OOD-H'	9884	66.04	52.68	82.76	37.82	38.55
Qwen2-VL-7B	OOD-S	6000	73.53	69.84	81.16	61.30	48.54
	OOD-H	6000	57.55	50.05	70.99	42.53	15.83
	OOD-S'	25800	73.12	69.32	80.75	60.72	47.74
	OOD-H'	9884	59.02	51.87	62.85	44.15	18.91
<b>CityScapes</b>							
Qwen2-VL-2B	OOD-S	3820	88.85	88.05	94.81	82.20	78.39
	OOD-H	3836	81.20	77.28	97.69	63.92	66.51
Qwen2-VL-7B	OOD-S	3820	86.10	85.58	88.89	82.51	72.39
	OOD-H	3836	75.36	72.42	82.24	64.70	51.92

Table 18: Performance of different model scales on OODBench. The table shows results for the 2B and 7B variants of the Qwen2-VL series across various data domains on OODBench. It can be observed that there is no monotonic relationship between model scale and OOD robustness; increasing parameter scale does not consistently improve model performance under semantic OOD conditions. Where, OOD-S' and OOD-H' denote the initial benchmark datasets.

## H CLIP AND BLIP2 VISUALIZATION DISTRIBUTION LOGITS ON COCO VAL

**Why is OODBench Challenging?** The appearance of OOD data is usually caused by semantic shift or covariate shift, which means that the generation mechanism of the test data has changed from the training data. Although a model performs well on the training set, it may show degraded or unstable performance when faced with a change in the data distribution. Models usually lack sufficient robustness to such changes, especially when dealing with changes that are not supported by explicit labels or historical data. We divide the OODBench data containing covariate shift through the OOD detector, which has shifted its data distribution despite sharing the same label space as the training data. Deep learning models often give high-confidence predictions or even make very confident false predictions when dealing with these data. The fundamental reason why OODBench poses a challenge to models is that it does not follow the distributional assumptions of the training data. In contrast, OOD data from the real world is highly uncertain and diverse. When confronted with these data, models typically lack the necessary knowledge and ability to make accurate predictions, which can lead to misclassification or high-confidence false predictions. We shows a visualization of the distribution of logits predicted by VLMs when confronted with real-world data.

Figures 15 and 16 show the results of CLIP and BLIP2 for computing logits on the COCO validation set for 80 labeled categories. Figure 14, on the other hand, summarises the logit results of CLIP and BLIP2 on the COCO validation set for 80 categories. As shown in Figure 14, the logits distribution shows a double peak, appearing on both sides of the distribution, which indicates that the CLIP and BLIP2 models have high confidence in the instance category. However, the single peak on the far left side of Figure 14 indicates that both models also have high confidence in determining instance-category mismatches, resulting in high-confidence false predictions.

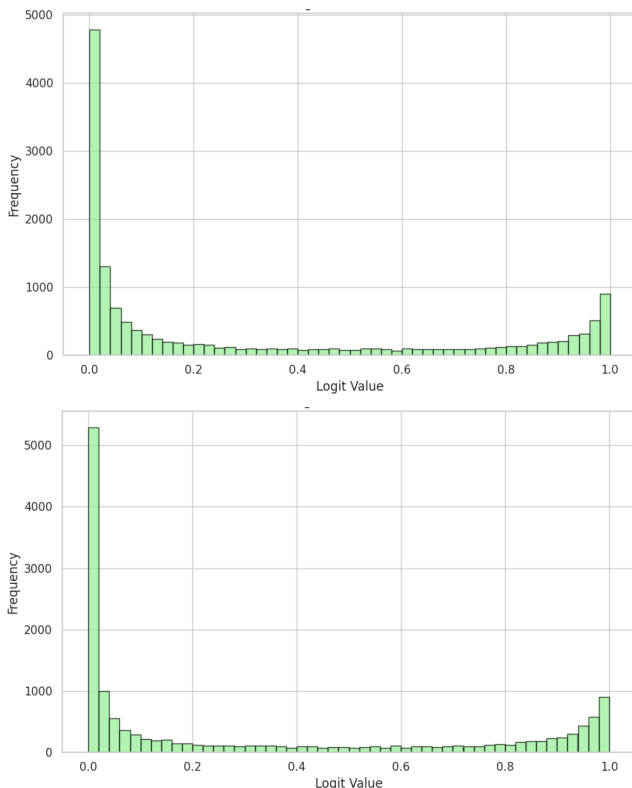


Figure 14: Logistically visualized distributions of CLIP (top image) and BLIP2 (bottom image) on coco val.

2592  
 2593  
 2594  
 2595  
 2596  
 2597  
 2598  
 2599  
 2600  
 2601  
 2602  
 2603  
 2604  
 2605  
 2606  
 2607  
 2608  
 2609  
 2610  
 2611  
 2612  
 2613  
 2614  
 2615  
 2616  
 2617  
 2618  
 2619  
 2620  
 2621  
 2622  
 2623  
 2624  
 2625  
 2626  
 2627  
 2628  
 2629  
 2630  
 2631  
 2632  
 2633  
 2634  
 2635  
 2636  
 2637  
 2638  
 2639  
 2640  
 2641  
 2642  
 2643  
 2644  
 2645

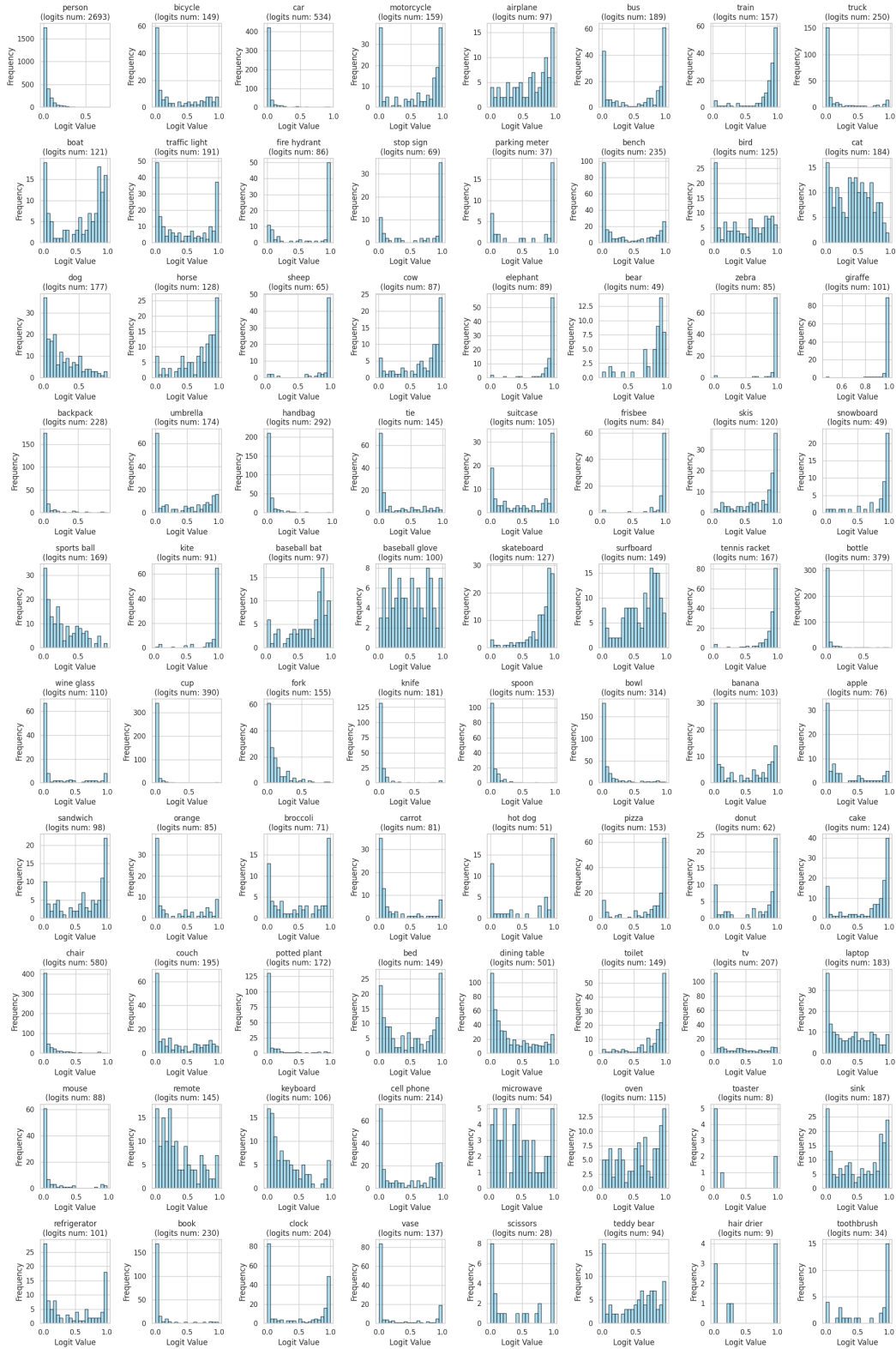


Figure 15: CLIP visualization distribution of 80 category logits on coco val.

2646  
 2647  
 2648  
 2649  
 2650  
 2651  
 2652  
 2653  
 2654  
 2655  
 2656  
 2657  
 2658  
 2659  
 2660  
 2661  
 2662  
 2663  
 2664  
 2665  
 2666  
 2667  
 2668  
 2669  
 2670  
 2671  
 2672  
 2673  
 2674  
 2675  
 2676  
 2677  
 2678  
 2679  
 2680  
 2681  
 2682  
 2683  
 2684  
 2685  
 2686  
 2687  
 2688  
 2689  
 2690  
 2691  
 2692  
 2693  
 2694  
 2695  
 2696  
 2697  
 2698  
 2699

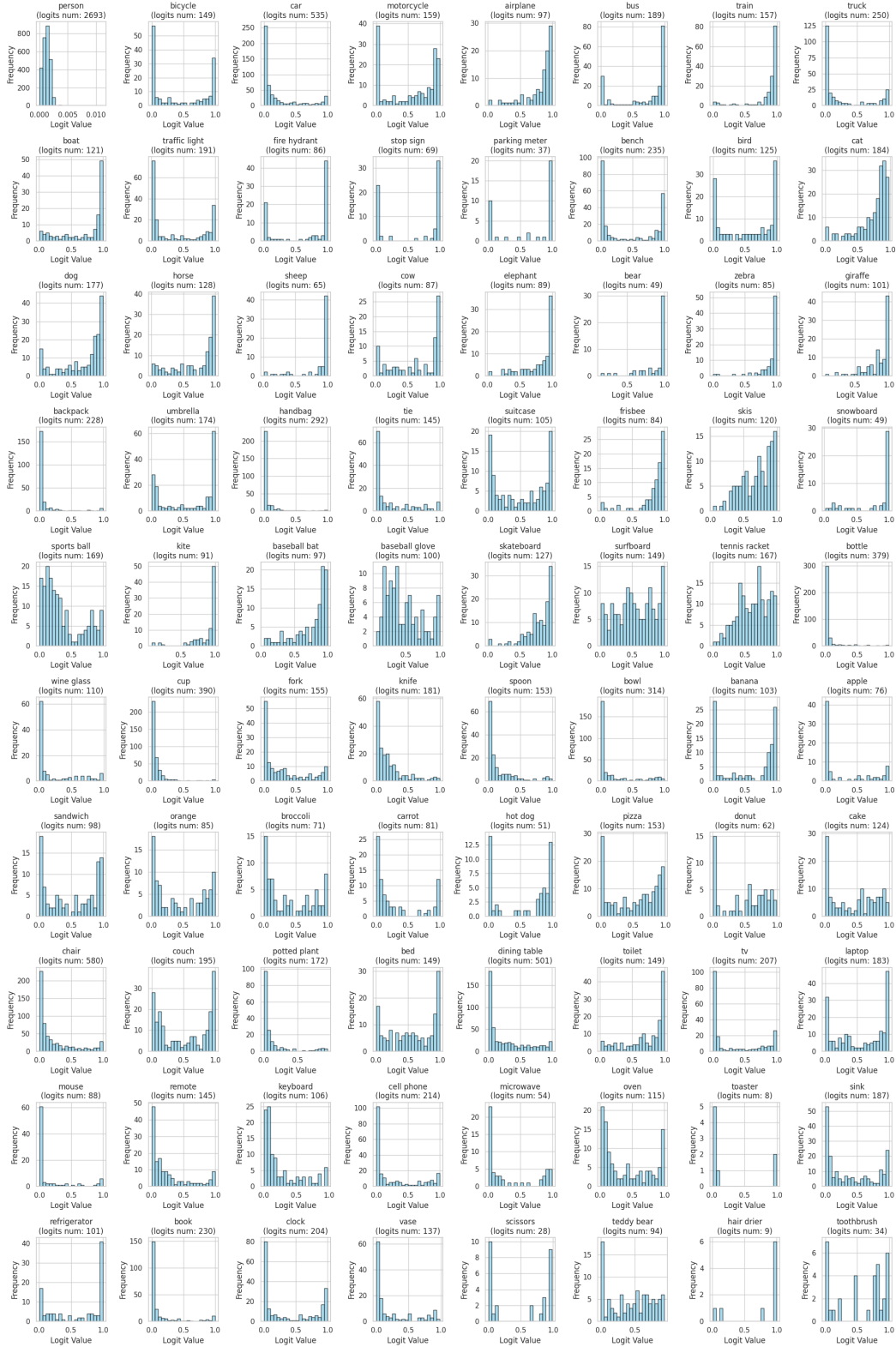


Figure 16: BLP2 visualization distribution of 80 category logits on coco val.

2700  
2701  
2702  
2703  
2704  
2705  
2706  
2707  
2708  
2709  
2710  
2711  
2712  
2713  
2714  
2715  
2716  
2717  
2718  
2719  
2720  
2721  
2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729  
2730  
2731  
2732  
2733  
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753

## I OODBENCH DATA AND CASE DEMONSTRATION

In this section, we show some of the data examples constructed by OODBench, covering OOD scenarios divided from datasets such as COCO, LVIS, nuScenes, and Cityscapes. Figures 17 and 18 show the associated visualisations. Through these examples, we aim to help readers visualise the characteristics of different types of out-of-distribution data and the challenges they pose under different tasks and perceptual conditions faced by large multimodal models.

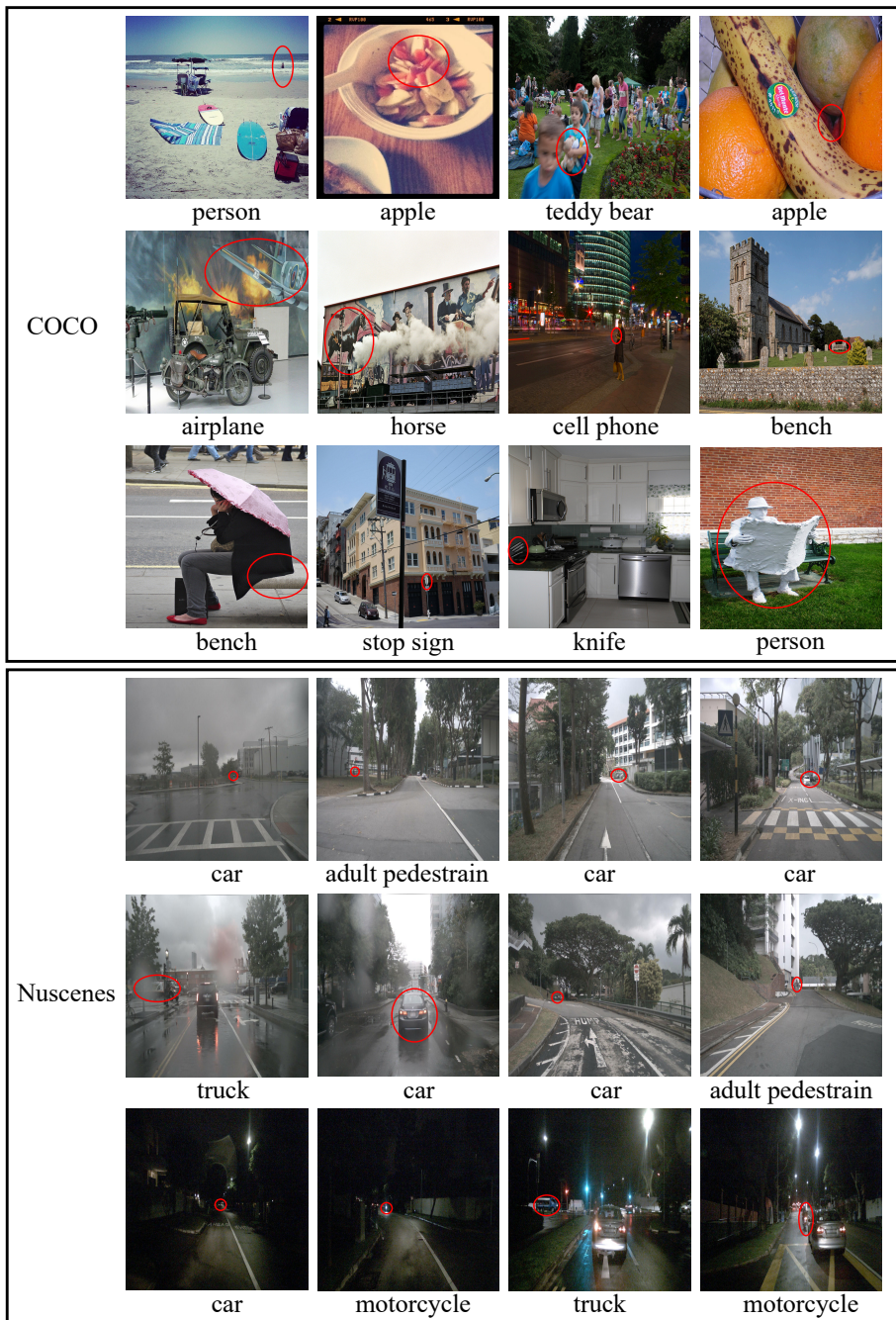


Figure 17: Sample OOD instances from the COCO and nuScenes datasets are divided by OOD-Bench. These examples illustrate diverse out-of-distribution categories across different scenarios and environments.

2754  
 2755  
 2756  
 2757  
 2758  
 2759  
 2760  
 2761  
 2762  
 2763  
 2764  
 2765  
 2766  
 2767  
 2768  
 2769  
 2770  
 2771  
 2772  
 2773  
 2774  
 2775  
 2776  
 2777  
 2778  
 2779  
 2780  
 2781  
 2782  
 2783  
 2784  
 2785  
 2786  
 2787  
 2788  
 2789  
 2790  
 2791  
 2792  
 2793  
 2794  
 2795  
 2796  
 2797  
 2798  
 2799  
 2800  
 2801  
 2802  
 2803  
 2804  
 2805  
 2806  
 2807

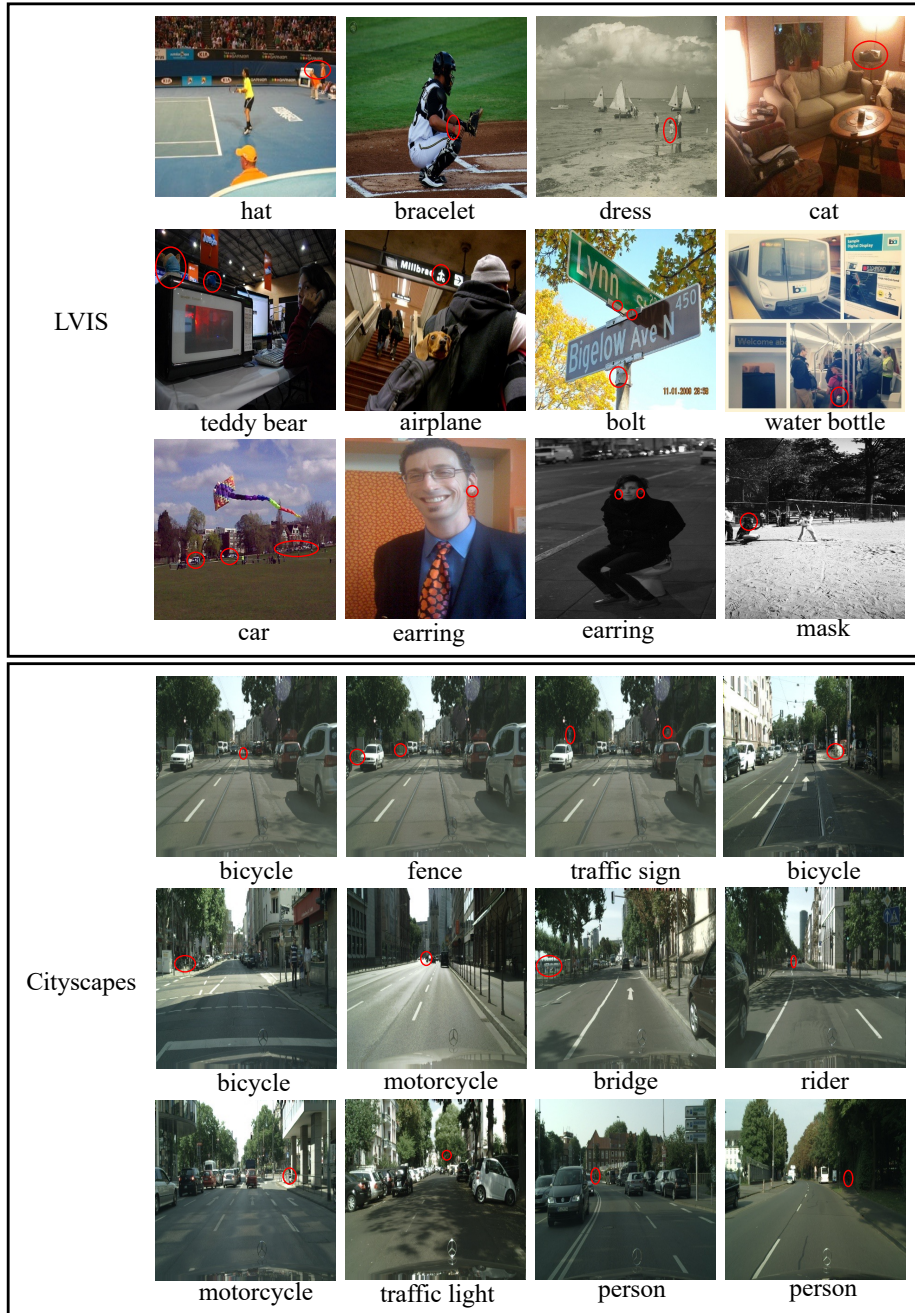


Figure 18: Sample OOD instances divided from the LVIS and Cityscapes datasets by OODBench. The samples highlight distributional shifts in object-centric.

2808 To further demonstrate the applicability and challenge of OODBench, we constructed typi-  
2809 cal examples on four datasets, COCO, LVIS, nuScenes, and Cityscapes, which are shown in  
2810 Figs. 19, 20, 21, 22, respectively. Each set of examples covers a variety of question types, including  
2811 standard question answering, Chain-of-Thought prompting, and tasks such as existential, counting,  
2812 and logical reasoning under BAP metrics. These cases visualise the challenges of perception and  
2813 reasoning in large multimodal models in different out-of-distribution scenarios.

2814  
2815  
2816  
2817  
2818  
2819  
2820  
2821  
2822  
2823  
2824  
2825  
2826  
2827  
2828  
2829  
2830  
2831  
2832  
2833  
2834  
2835  
2836  
2837  
2838  
2839  
2840  
2841  
2842  
2843  
2844  
2845  
2846  
2847  
2848  
2849  
2850  
2851  
2852  
2853  
2854  
2855  
2856  
2857  
2858  
2859  
2860  
2861

2862  
2863  
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879  
2880  
2881  
2882  
2883  
2884  
2885  
2886  
2887  
2888  
2889  
2890  
2891  
2892  
2893  
2894  
2895  
2896  
2897  
2898  
2899  
2900  
2901  
2902  
2903  
2904  
2905  
2906  
2907  
2908  
2909  
2910  
2911  
2912  
2913  
2914  
2915

### Case Demonstration (COCO)



Book

**User:** Does this image contain a book? (yes or no)  
**Gemini:** **No.**  
**GPT-4o:** **No, the image does not contain a book.**

---

**User:** Does this image not contain a book? (yes or no)  
**Gemini:** **No.**  
**GPT-4o:** **No, the image does not contain a book.**

**CoT**

**User:** Does this image contain a book? Yes or No? Let's break down the information step by step.  
**Gemini:** **No. I don't see any books in this image.**  
**GPT-4o:** To determine if the image contains a book, let's examine the visible elements step by step:

1. Television Area: In the area with the television, there are some items on the table, but they appear to be decorative objects and not books.
2. Dining Area: Near the dining table, there doesn't seem to be any visible books. The table has some decorative items, such as vases.
3. Kitchen Area: There are no visible books in the kitchen area; only kitchen appliances and cabinets.
4. General Overview: There are shelves and furniture, but no apparent books.

**Based on the visible elements, the image does not seem to contain a book.**

**CoT**

**User:** Does this image not contain a book? Yes or No? Let's break down the information step by step.  
**Gemini:** **Yes.** There are no visible books in this image.  
**GPT-4o:** **No,** the image does not contain a book. Here's the breakdown:

1. Living Area: The image shows a living area that includes furniture such as a TV, chairs, and a table.
2. Kitchen/Dining Area: There is a dining table with chairs and other kitchen appliances visible.
3. Decor: Vases and decorative items are present, but no books are visible.

Thus, based on the visual information, a book is not present in the image.



Bottle: 0      Person: 4

**BAP**

**User:** Does this image contain a bottle? (yes or no)  
**Gemini:** **No.**  
**GPT-4o:** **No,** the image does not contain a bottle.  
**User:** How many person are there in the picture?  
**Gemini:** There are at least **three** people visible:

1. A person in white near a van.
2. A person walking away from the camera near a small white shelter.
3. Another person behind the first, also near the shelter. It's possible there are more people obscured by the cars or inside the white shelter.

**GPT-4o:** I **can't determine** the number of people in the picture.  
**User:** Is the number of bottle in the image greater than the number of person? Answer with 'yes' or 'no'.  
**Gemini:** **No.**  
**GPT-4o:** **No.**

Figure 19: This is a showcase of OODBench examples on the COCO dataset. Examples cover standard question-and-answer, chain-of-thought prompts, and task types such as existential, counting, and logical reasoning. Green font indicates correct responses given by the model, and red font indicates incorrect responses.

2916  
2917  
2918  
2919  
2920  
2921  
2922  
2923  
2924  
2925  
2926  
2927  
2928  
2929  
2930  
2931  
2932  
2933  
2934  
2935  
2936  
2937  
2938  
2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953  
2954  
2955  
2956  
2957  
2958  
2959  
2960  
2961  
2962  
2963  
2964  
2965  
2966  
2967  
2968  
2969

### Case Demonstration (LVIS)



Sunglasses

**User:** Does this image contain a sunglasses? (yes or no)

**Gemini:** **No.**

**GPT-4o:** **No.**

---

**User:** Does this image not contain a sunglasses? (yes or no)

**Gemini:** **No.**

**GPT-4o:** **No, the image does not contain sunglasses.**

**CoT**

**User:** Does this image contain sunglasses? Yes or No? Let's break down the information step by step.

**Gemini:** **No.** The image contains a baseball player, but he is not wearing sunglasses. The player behind him is wearing some type of eyewear, but it's hard to determine if they are sunglasses or protective eyewear.

**GPT-4o:** **Yes,** the image contains sunglasses. The person in the foreground is wearing them on their cap.

**CoT**

**User:** Does this image not contain sunglasses? Yes or No? Let's break down the information step by step.

**Gemini:** **No.** None of the people in the image are wearing sunglasses.

**GPT-4o:** **Yes,** the image contains sunglasses. One of the people in the background appears to be wearing them.



Cup: 1
Egg: 1

**BAP**

**User:** Does this image contain a cup? (yes or no)

**Gemini:** **No.**

**GPT-4o:** **Yes.**

---

**User:** How many egg are there in the picture?

**Gemini:** I can only see **one** whole egg on the table, right next to the orange and the cucumber. There might be more ingredients in the blender, but it's impossible to tell.

**GPT-4o:** There are **no** eggs visible in the picture.

---

**User:** Is the number of cup in the image greater than the number of egg? Answer with 'yes' or 'no'.

**Gemini:** **No.**

**GPT-4o:** **No.**

Figure 20: Demonstration of OODBench cases on the LVIS dataset. The questions shown cover a variety of task formats and visualise the performance of the model in an out-of-distribution scenario. Green font indicates correct responses, and red font indicates incorrect responses.

2970  
2971  
2972  
2973  
2974  
2975  
2976  
2977  
2978  
2979  
2980  
2981  
2982  
2983  
2984  
2985  
2986  
2987  
2988  
2989  
2990  
2991  
2992  
2993  
2994  
2995  
2996  
2997  
2998  
2999  
3000  
3001  
3002  
3003  
3004  
3005  
3006  
3007  
3008  
3009  
3010  
3011  
3012  
3013  
3014  
3015  
3016  
3017  
3018  
3019  
3020  
3021  
3022  
3023

### Case Demonstration (nuScenes)



Car

**User:** Does this image contain a car? (yes or no)

**Gemini:** Yes.

**GPT-4o:** Yes.

---

**User:** Does this image not contain a sunglasses? (yes or no)

**Gemini:** No.

**GPT-4o:** No.

**CoT**

**User:** Does this image contain car? Yes or No? Let's break down the information step by step.

**Gemini:** Yes. There are several cars parked along the road and at least one driving in the distance.

**GPT-4o:** No, the image does not contain a car. Here's the breakdown:

1. Foreground: The road appears to be empty. There's no car visible in the immediate foreground.
2. Midground: There is a bicycle visible, but no car is present.
3. Background: Although there are some objects and structures, there isn't a clear presence of a car in the background.

Therefore, based on this analysis, the image does not contain a car.

**CoT**

**User:** Does this image not contain car? Yes or No? Let's break down the information step by step.

**Gemini:** No. There are several cars visible in the image, parked along the side of the road and driving down the street.

**GPT-4o:** Let's break down the information in the image:

1. Roadway: The image shows a roadway leading into the distance.
2. Presence of Vehicles: A car is visible in the distance on the road.

Therefore, the answer is No, the image does contain a car.



Construction Worker : 3  
Construction Vehicle : 1

**BAP**

**User:** Does this image contain a construction worker? (yes or no)

**Gemini:** Yes. There is a person visible in the cab of the excavator on the right side of the image, who is likely a construction worker.

**GPT-4o:** Yes.

**User:** How many construction vehicle are there in the picture?

**Gemini:** I can see one construction vehicle in the image - a small excavator parked on the right side of the road.

**GPT-4o:** There is one construction vehicle in the image, located on the right side.

**User:** Is the number of construction worker in the image greater than the number of construction vehicle? Answer with 'yes' or 'no'.

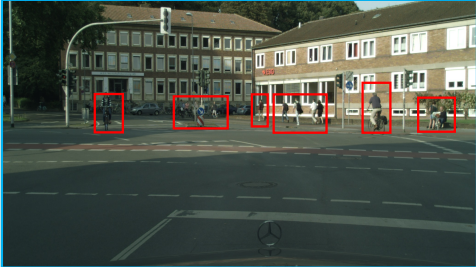
**Gemini:** No.

**GPT-4o:** No.

Figure 21: OODBench case demonstrations on the nuScenes dataset. It includes visual Q&A and complex inference tasks in multimodality; green font indicates the model's correct predictions, and red font indicates incorrect predictions.

3024  
3025  
3026  
3027  
3028  
3029  
3030  
3031  
3032  
3033  
3034  
3035  
3036  
3037  
3038  
3039  
3040  
3041  
3042  
3043  
3044  
3045  
3046  
3047  
3048  
3049  
3050  
3051  
3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077

### Case Demonstration (Cityscapes)



Person

**User:** Does this image contain a person? (yes or no)

**Gemini:** Yes.

**GPT-4o:** Yes.

---

**User:** Does this image not contain a person? (yes or no)

**Gemini:** No.

**GPT-4o:** No.

**CoT**

**User:** Does this image contain person? Yes or No? Let's break down the information step by step.

**Gemini:** Yes. There are multiple people visible in the image, some walking and some cycling.

**GPT-4o:** Yes, the image contains people.

1. Foreground: There are people on bicycles crossing the road.
2. Background: There are pedestrians walking on the sidewalk.

These observations confirm the presence of multiple people in the image.

**CoT**

**User:** Does this image not contain person? Yes or No? Let's break down the information step by step.

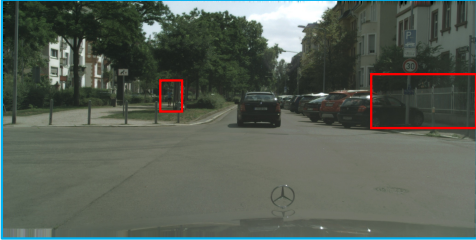
**Gemini:** No. There are several people visible in the image, some walking and some riding bicycles.

**GPT-4o:** Yes, the image contains people.

Now, let's break it down:

1. The image shows a street scene.
2. There are individuals visible in various positions in the scene: walking on the sidewalk, riding bicycles, etc.

So, the answer is "Yes," the image does contain people.



Person: 1      Fence: 1

**BAP**

**User:** Does this image contain a person? (yes or no)

**Gemini:** Yes.

**GPT-4o:** Yes.

---

**User:** How many fence are there in the picture?

**Gemini:** There are multiple fences visible in the image.

1. A metal fence along the sidewalk in the lower left corner. Several metal posts are spaced along the curb.
2. A white metal fence running along the right sidewalk, in front of the buildings and parked cars.

There may be other fences obscured by buildings or trees.

**GPT-4o:** There are two fences visible in the picture. One is on the left side, consisting of several vertical posts, and the other is on the right side, along the sidewalk.

---

**User:** Is the number of person in the image greater than the number of fence? Answer with 'yes' or 'no'.

**Gemini:** No.

**GPT-4o:** No.

Figure 22: A showcase of OODBench cases on the Cityscapes dataset. A wide range of question types is covered to assess the model’s ability to generalize to out-of-distribution environments such as urban streetscapes. Green font represents correct responses, and red font represents incorrect responses.

## 3078 J LIMITATIONS

3079

3080

3081

3082

3083

3084

3085

3086

3087

3088

3089

3090

3091

3092

3093

3094

3095

3096

3097

3098

3099

3100

3101

3102

3103

3104

3105

3106

3107

3108

3109

3110

3111

3112

3113

3114

3115

3116

3117

3118

3119

3120

3121

3122

3123

3124

3125

3126

3127

3128

3129

3130

3131

## K LLMs USAGE

Large language models (LLMs) were used solely for language polishing and minor improvements in the clarity and readability of the paper. No LLMs were used for research ideation, experimental design, implementation, analysis, or drafting of technical content. All research contributions and core writing were conducted entirely by the authors.