Tricking Open-World Object Recognition Models: Uncertainty in Out-of-Distribution Detection

Wout Teillers and Matias Valdenegro-Toro
Bernoulli Institute, Faculty of Science and Engineering, University of Groningen
w.j.a.teillers@student.rug.nl, m.a.valdenegro.toro@rug.nl

Abstract

Object recognition models are well studied on benchmark datasets, typically focusing on performance in retrieving objects that exist in images. However, in real-life scenarios there is no prior knowledge of an object's existence, and current research fails to assess model performance in these situations. This research aims to shed light on this problem by testing three Open-World models, YOLO-World, Grounding Dino and GPT-4o, on the LVIS, Open Images, and JUS datasets. We design an experiment where models are confronted with impossible prompts by instructing them to retrieve non-existing objects. This allows us to observe the models' uncertainty performance. Overall, GPT-4o performed poorest with regard to object recognition and uncertainty estimation. GPT-4o showed to be highly overconfident. In contrast, YOLO-World and Grounding Dino are slightly underconfident, but they are superior in their uncertainty calibration in comparison to GPT-4o. However, all three models occasionally assign high confident predictions to non-existing objects. Showing that improvement can still be made to the uncertainty estimation of these models when confronted with impossible prompts.

1 Introduction

Object recognition is one of the most important fundamental parts of computer vision, and the last two decades the amount of research on this topic has increased tremendously (Zou et al., 2023). Object recognition is an essential task in computer vision that involves classifying and localizing objects within images or video frames. This technology has found widespread application across numerous domains, such as autonomous vehicles, industrial automation, video surveillance, and various other fields.

Object recognition models first classify zero or more objects within an image. In contrast to image classification, these models can recognize multiple objects within a picture, whereas image classification aims to classify a picture as a whole. In addition to classification, object recognition models are capable of the localization of objects through the application of bounding boxes. These bounding boxes show the region where the object can be found within the picture. In Figure 2 it can be seen how a model retrieves objects from an image depending on which classes are prompted. Other examples can be found in Appendix I. Moreover, object recognition models are trained on a dataset with a finite set of object classes. This limits their ability to classify objects that are beyond the classes within the training data. As a result, such objects can not be classified correctly. However, research has found that whenever faced with unknown objects, these models frequently misclassify them as an object class in the training data (Joseph et al., 2021). This behavior is undesirable, as it decreases the reliability of the predictions. Furthermore, research has found that even small modifications to specific sub-regions of an image can affect the model's ability to detect objects in other, non-local parts of the scene (Rosenfeld et al., 2018). In particular, adding objects whose classes were present during training can lead the model to

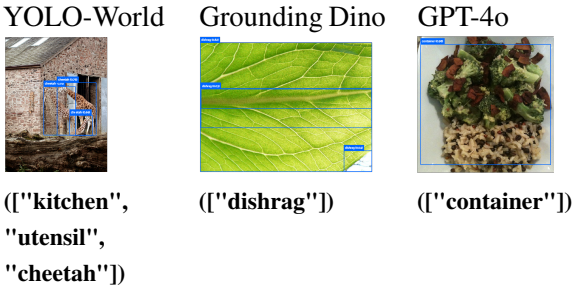

Figure 1: Predictions of missing labels of the three various models, these examples showcase that the models give prediction for non-existing labels with a high confidence. Showing the inability to handle their uncertainty.

Proceedings of the 4th Workshop on Towards Knowledgeable Foundation Models (KnowFM 2026), pages 147–164
July 3, 2026 ©2026 Association for Computational Linguistics

Ground truth	['skirt']	['skirt', 'flag-pole']	['skirt', 'flag-pole', 'flip-flop_(sandal)']	'skirt', 'flag-pole', 'flip-flop_(sandal)', 'flag']	['skirt', 'flag-pole', 'flip-flop_(sandal)', 'flag', kite']

Figure 2: Subset of ground-truth classes evaluated with YOLO-World, illustrating how the model reassigns object labels when it predicts higher confidence for an alternate class.

flip or suppress the classification of other objects in the image. This shows the brittleness of the models to misclassify objects. These misclassifications presents a serious issue, particularly in high-risk scenarios where incorrect detections could lead to severe consequences (Andres et al., 2024).

Current research is focused on the performance of these models on benchmark labeled datasets. The focus lies in detecting objects that are present in the images. However, current research on these models does not take into account what happens when these models try to detect objects that are not present in an image. Object recognition models often fail to correctly refrain from identifying missing object classes or fail to demonstrate its uncertainty correctly as shown in Figure 1. Therefore, this research aims to shed light to this problem, as in real-life scenarios models often don't know if an object is present in the image or not. Therefore, understanding their behavior in these situation is crucial. This gives rise to the following research question and sub research questions:

1. How do Open-World object recognition models perform when confronted with impossible prompts?

2. How do these models react when asked to identify objects that are not visible?

3. To what extent are these models capable of recognizing when an object is absent and cannot be detected?

The goal is that models can correctly show how confident they are with their predictions, also known as their uncertainty. This uncertainty is crucial such that users can trust the output of the models. As such, understanding and addressing the uncertainty in these models is vital for their responsible development. These models function as foundational models as they can be implemented in a wide range of applications. The goal of this research is to test the knowledge of these models in their open-world capabilities. This is done by exploring the performance and limitation of them in scenarios where models should be highly uncertain by including objects that are absent from the images. The contributions of this work are: an evaluation of several open world object detection models under labels that are prompted but missing from the image, results evaluating calibration and task performance, and our evaluation reveals shortcomings of open world object detection models in terms of calibration under a form of distribution shift.

1.1 State-of-the-art

Classical object recognition models are only capable to detect object classes which were present in the training dataset, leading to the inability of detecting unseen objects. However, it is crucial that these models are able to perform out-of-distribution detection (Yang et al., 2024). Several solutions are proposed to address this issue. First of all, open-set object recognition models can recognize unknown objects as "unknown", thereby eliminating the issue of the incorrect labeling of object classes beyond the training data. This addresses the issue of the inability to generalize beyond the scope of trained classes (Joseph et al., 2021). To extend this solution the term 'Open-World Object Recognition models' has been coined. Research on Open-World object recognition aims to solve the issues regarding the limited classes of the training data by allowing the model to learn object classes beyond these classes (Li et al., 2024). For example, Open-World object recognition models can classify objects that are not included in the training data as "unknown". Subsequently, these "unknown" classes can be externally classified, which allows the models to learn these new classes without the need for retraining. Hereby,

open-set and Open-World object recognition models help mitigate the problem of encountering unknown objects.

Furthermore, zero-shot models have been introduced to allow models to identify object classes beyond the classes that it was trained on (Cao et al., 2025). This is done by looking at the proximity of words in the word embedding space. Zero-shot models utilize this principle to allow for the classification of classes outside of the training data. This allows zero-shot models to classify objects it has not seen in training.

The issue of handling unknown objects has been addressed, but uncertainty remains a potential challenge in object recognition. Even though confidence scores are already present, these do not always reflect correct uncertainty. Models must not be overconfident as this can be problematic, especially in high-stakes decision-making scenarios (Valdenegro-Toro, 2021). Therefore evaluating this uncertainty remains crucial.

Utilizing this confidence score increases user trust, however this does not help mitigate model hallucinations. Model hallucinations are high confidence incorrect outputs and reducing them often has a negative impact on object recall (Ren et al., 2024). These hallucinations limit the ability for proper application, as observed in Vision-Language Models (Liu et al., 2024a). Therefore, hallucinations are also major limitation in object recognition.

In conclusion, object recognition models have been developed to address the challenges of out-of-distribution detection. Models incorporate uncertainty scores reflecting the confidence in their predictions. Both features are designed to enhance the trustworthiness of object recognition systems and improve their ability to manage uncertainty.

Since limited research has been done on the behavior of models when faced with impossible prompts, this study aims to address this gap. We will compare open-world object recognition models on benchmark datasets through an experiment in which the models are presented with impossible prompts. Previous work has shown that large language models are prone to hallucinate (Huang et al., 2025) and this problem has also been observed in large vision language models (Sahoo et al., 2024). In general, hallucinations are a significant issue in large foundation models (Jin et al., 2025), therefore, we hypothesize that these foundation object detection models will also hallucinate in these impossible scenarios.

2 Methodology

2.1 Models

In this research, three models are evaluated on three datasets to aid in understanding the overconfidence level of Open-World object recognition models. Each model has shown individual excellence in the performance of object recognition tasks, by implementing new model designs to overcome the limitations of previous models, and by scoring high mAP scores on benchmark datasets.

The first model that will be used in this experiment is YOLO-World specifically the YOLOv8l-world version. YOLO-World is an Open-Vocabulary Open-World object recognition model that extends the limitation of models trained on a relatively small number of class datasets and extends previous YOLO (You Only Look Once) detectors. This enables this model to recognize objects outside of its initial training data (Cheng et al., 2024). Open-Vocabulary models, such as YOLO-World, are designed to classify objects beyond the predefined classes they were trained on, offering a significant advantage in real-life environments. The weights of the YOLO-World model are imported via the Ultralytics library.

The second Open-World model is Grounding DINO, a model built upon DINO (DETR with Improved deNoising anchOr boxes)(Zhang et al., 2022). Grounding DINO is an Open-Set object recognition model that is capable of recognizing objects given text commands as input (Liu et al., 2024b). The design of Grounding DINO is made to better combine the cross modality information. Hereby, Grounding DINO is able to fuse the text features with the image features, which results in better overall performance such as a 52.5 AP on the COCO (Common Object in Context) detection zero-shot transfer benchmark (Liu et al., 2024b). The Grounding DINO model is implemented using a Hugging Face environment, and the grounding-dino-base model is used in this research.

The final model is GPT-4o, a Generative Pre-trained Transformer, which is a multimodal system capable of processing various types of data including images and text, making it suitable for object recognition tasks (Yang et al., 2023). Trained on a diverse number of data sources, GPT-4o's ability to handle object detection presents an interesting topic for comparison in this study. GPT-4o is accessed using a chat completion API provided by OpenAI, including the image and the prompt. The prompt

(in Appendix C) is designed such that the output format is in line with YOLO-World and Grounding Dino.

This comparison aims to shed light on the capabilities and limitations of these models, particularly in handling impossible prompts and uncertainty.

2.2 Datasets

To test these models, this research will make use of the LVIS validation set. The LVIS dataset serves as a widely used benchmark for object recognition tasks and is used to evaluate both the YOLO-World model and the Grounding DINO model. This dataset consists of 164,000 images, spanning 1000 distinct object categories (Gupta et al., 2019).

The second dataset is the Open Images dataset, which is a benchmark dataset used for the training and validation of a wide range of state-of-the-art computer vision tasks (Kuznetsova et al., 2020). The dataset consists of around 9 million images including 600 object categories. This research will use its validation set for its experiment.

Lastly, a dataset containing Japanese uncertainty scenes will be used. This dataset contains pictures designed for testing the uncertainty of Vision-Language Models (VLMs) (Groot and Valdenegro-Toro, 2024), and is therefore expected to give interesting insights into the performance of the state-of-the-art object recognition models. The dataset is available via `https://github.com/ML-RUG/jus-dataset` and contains images, including prompts used for Vision Language Evaluation.

2.3 Data Gathering

Both the LVIS and Open Images datasets are imported using a FiftyOne library, which provides an easy interaction with the datasets. In this study, 1000 images of each of these two datasets are tested by setting the *max_samples* parameter to 1000.

The full dataset of the Japanese Uncertainty Scenes is used, containing 39 images. This dataset does not contain ground truth values for the objects. To manually label these objects, the information in the prompt is used. When no object classes are present in the prompt, the objects in the image will be manually assigned. This ensures that all images have labeled existing objects. The labels assigned for all images can be found in Appendix H.

Each model is tested on all three datasets. For each sample in the dataset, a list of ground truth object classes is retrieved. Afterwards, a list of missing object classes is generated. The aim for the missing classes is to test the out-of-distribution detection capabilities of the models. This list is generated as follows:

Let $\mathcal{C}_{\text{LVIS}}$ denote the set of classes in the LVIS dataset, $\mathcal{C}_{\text{OI}}$ denote the set of classes in the Open Images (OI) dataset and $\mathcal{C}_{\text{JUS}}$ denote the set of classes in the Japanese Uncertainty Scenes (JUS) dataset.

We then define the missing classes set used for testing as follows:

1. **Missing classes for testing on Open Images:**

$$\mathcal{C}_{\text{miss, OI}} = \mathcal{C}_{\text{LVIS}} \setminus \mathcal{C}_{\text{OI}} \tag{1}$$

2. **Missing classes for testing on LVIS:**

$$\mathcal{C}_{\text{miss, LVIS}} = \mathcal{C}_{\text{OI}} \setminus \mathcal{C}_{\text{LVIS}} \tag{2}$$

3. **Missing classes for testing on JUS:**

$$\mathcal{C}_{\text{miss, JUS}} = \mathcal{C}_{\text{LVIS}} \setminus \mathcal{C}_{\text{JUS}} \tag{3}$$

To further ensure the quality of the missing classes list, we remove common classes such as 'human face' from the $\mathcal{C}_{\text{LVIS}}$, $\mathcal{C}_{\text{OI}}$ and $\mathcal{C}_{\text{JUS}}$ as defined in Table 6. These classes might not be labeled but are expected to be present in the images and are therefore removed. The full list of removed classes can be found in Appendix G.

Table 1: Class sets and their description.

Class set	Description
$\mathcal{C}_{\text{LVIS}}$	LVIS classes
$\mathcal{C}_{\text{OI}}$	Open Images (OI) classes
$\mathcal{C}_{\text{JUS}}$	Japanese Uncertainty Scenes (JUS) classes
$\mathcal{C}_{\text{miss, OI}}$	Missing classes for OI evaluation
$\mathcal{C}_{\text{miss, LVIS}}$	Missing classes for LVIS evaluation
$\mathcal{C}_{\text{miss, JUS}}$	Missing classes for JUS evaluation

After computing these unique sets, we further filter the synset lemmas for each object class using Wordnet. Object classes that have a synset relation with the original class are removed. Synsets represent related concepts, so retaining these object classes could reduce the distinctiveness of the missing classes. This process ensures the distinctiveness of the object classes. In our experimental setup, as found in Table 1:

- The set $\mathcal{C}_{\text{miss, OI}}$ is used for retrieving the missing classes for testing on the Open Images dataset.

Table 2: Metrics of Various Models on Different Datasets.

	LVIS			OI		
Model	mAP	AUC	ECE	mAP	AUC	ECE
YOLO-World	0.294	0.85	0.136	0.394	0.82	0.090
Grounding DINO	0.056	0.73	0.032	0.283	0.72	0.024
GPT-4o	0.009	0.83	0.484	0.052	0.87	0.435

- The set $\mathcal{C}_{\text{miss, LVIS}}$ is used for retrieving the missing classes for testing on the LVIS dataset.
- The set $\mathcal{C}_{\text{miss, JUS}}$ is used for retrieving the missing classes for testing on the JUS dataset.

This approach to deriving and filtering the set of missing labels is intended to provide a label set suitable for sampling labels that are likely to be absent from the images used during evaluation.

For each sample, both the known and missing classes are evaluated using each model. To maintain balance, the number of missing classes selected matches the number of known classes. Specifically, if the number of known classes is n, then n random samples are drawn from the missing class list. To ensure reproducibility, a seed is used for the random number generator.

Lastly, more settings on the project regarding hyperparameters and postprocessing can be found in Appendix J and K.

3 Results

mAP: In Table 2, the mAP scores of the different models and datasets can be found. The mAP score is calculated using the COCO evaluation protocol (Lin et al., 2014), using an IOU value of 0.5. The mAP scores show the performance of the classification and localization of a model and are used as a standard metric in the evaluation and comparison of object recognition models.

ROC curves: In Figure 3, the ROC curves of the models' predictions are shown. The ROC curve plots the false positive rate against the true positive rate and includes the Area Under the Curve (AUC). It is a useful tool for visualizing the trade-off between precision and recall of a model (Fawcett, 2006). The diagonal line shows random prediction behavior of a model with an AUC score of 0.5. An AUC score of 1.0 indicates perfect discrimination between true positives and false positives. Figure 3 shows the differences between the LVIS and the

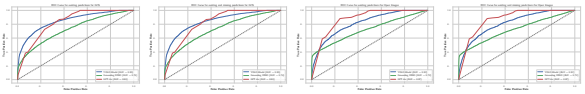

Figure 3: ROC curves for the three models on the LVIS and Open Images datasets. For each dataset, there is one plot displaying the curve for the predictions of the existing labels. The first two plots depict ROC curves of the models tested on the LVIS dataset: the first for existing labels and the second for both existing and missing labels. The last two plots extend this analysis to the Open Images dataset. The plots show the models' ability to discriminate between true and false positive predictions based on their confidence scores. Grounding Dino shows the poorest performance, with its ROC line closest to the diagonal and the lowest AUC score. Note that there is a marginal difference in the third to fourth significant figure of the AUC score when including the missing labels.

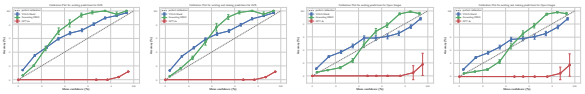

Figure 4: Calibration plots for three different models evaluated on the LVIS and Open Images datasets. These plots visualize the correspondence between predicted confidence scores and actual accuracy. The first two plots depict model calibration using the LVIS dataset: the first for existing labels and the second for both existing and missing labels. The last two plots extend this analysis to the Open Images dataset. The plots show that GPT-4o is highly overconfident and both YOLO-World and Grounding Dino are slightly underconfident.

Open Images dataset. The figure clearly shows that the choice of dataset can result in a difference in AUC score. Overall, Grounding Dino has the lowest AUC scores for both datasets, showing the worst performance in distinguishing between positive and negative predictions. This results in a higher false positive rate and consequently a lower AUC score. Furthermore, YOLO World shows the best performance on the LVIS dataset and GPT-4o on the Open Images dataset. In Figure 3, both plots of the ROC curve with and without the missing classes predictions are shown. The figure shows no difference in ROC curves when plotted using only the existing predictions and plotted using a combination of the existing and missing predictions.

Calibration Error: Figure 4 shows the calibration plot of testing the various models on the three datasets. The grey dotted line shows the perfect calibration where the confidence and the accuracy scores are equal. The error bars are included and

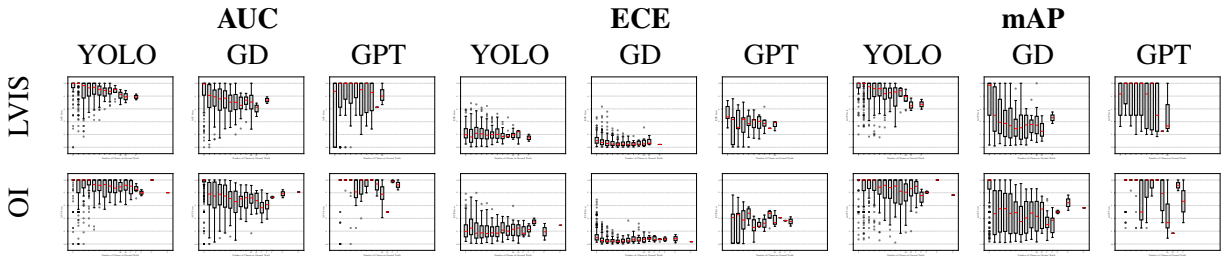

Figure 5: Boxplots of metrics versus the number of ground-truth class labels. The columns show YOLO-World (YOLO), Grounding DINO (GD), and GPT-4o (GPT), and the rows show the LVIS and Open Images (OI) datasets. AUC and ECE variability decreases with more classes, with AUC shifting higher and ECE lower, whereas mAP shows no clear relationship. GPT-4o shows more random behavior.

calculated using following formula: $\sigma = \sqrt{\frac{p(1-p)}{n}}$, where p is the probability representing the confidence scores, n is the total number of samples, and σ represents the standard deviation. Lastly, the plot is divided into 10 confidence bins, as the default number of bins for the calibration plot. This resulted in well-formed curves and was therefore kept unchanged.

The calibration plot reveals whether a model exhibits over/underconfidence. Models that are positioned mostly below the perfect calibration line are considered overconfident, whilst those above the line are viewed as underconfident. A perfectly calibrated model should align closely with the perfect calibration line. In Figure 4, it can be seen that GPT-4o is entirely under the perfect calibration line and therefore is seen as overconfident for both datasets. In contrast, YOLO-World and Grounding DINO are slightly above the perfect calibrated line for the LVIS dataset and are therefore seen as underconfident. YOLO-World and Grounding Dino both showcase better calibration for the Open Images dataset. Lastly, the results show that over/underconfidence occurs mostly on the LVIS dataset.

Additionally, when plotting the calibration for both existing and missing labels, the models' calibration is evaluated in a more realistic test scenario. These models are intended for real-world applications where pre-labeled images are not available, meaning it is uncertain whether an object is present in the image. Therefore, comparing predictions with both existing and missing labels provides valuable insights. Figure 4 illustrates these insights, showing that when missing predictions are included, the calibration line of Grounding DINO on the LVIS dataset shifts slightly closer to the perfect calibration line. However, no significant differences for the other calibration lines are observed.

The calibration errors of the models can be seen in Table 2. The calibration error figures allow for a good visual comparison between the models, but it is also beneficial to have a numeric representation of the calibration error (Guo et al., 2017). Therefore, the Expected Calibration Error (ECE) is used to calculate the numeric calibration errors of the models. The ECE is calculated as follows:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|, \quad (4)$$

where $\text{acc}(B_m)$ represents the fraction of true positive samples and $\text{conf}(B_m)$ represents the average confidence in each bin number m. In this formula, n represents the total number of samples and $m = 10$, as the calibration errors are divided into 10 bins.

Distribution of metrics for different class sizes: In Figure 5, the distribution of the different metrics for different numbers of classes in the ground truth can be seen. The figure shows that the variance of both the AUC score and ECE score decreases when the number of classes increases for YOLO-World and Grounding Dino. The AUC increases and the ECE decreases when the number of classes increases. However, GPT-4o shows no relation between the number of classes and the metrics. Furthermore, the figure shows that there is no relation between the number of classes and the mAP score.

Confidence Score Distribution: The distribution of confidence scores of each model on each dataset can be seen in Figure 7. Confidence scores are distinguished based on whether the labels were missing or existing. In Figure 7 it can be seen that GPT-4o predictions are skewed to the left in

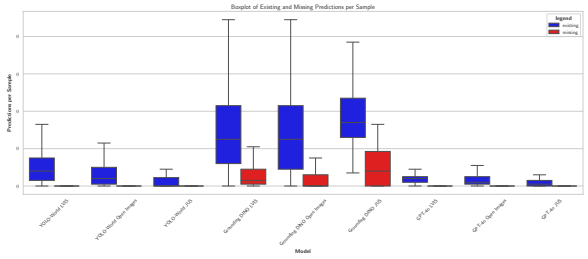

Figure 6: Boxplot illustrating per-sample prediction counts from various models applied to the same dataset. The figure compares the number of existing and missing predictions across models. Overall, Grounding DINO gives the highest number of predictions and all models give fewer predictions for the missing labels than for the existing labels.

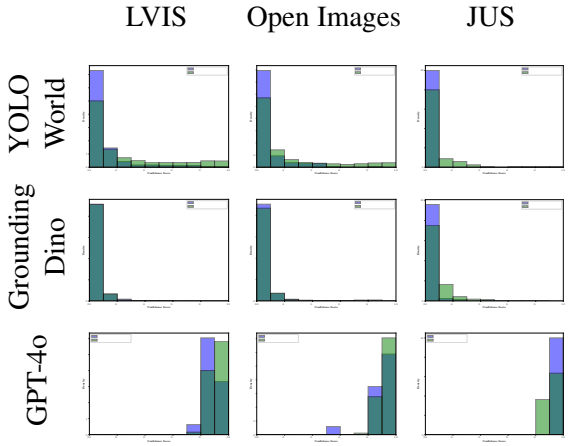

Figure 7: Density histogram showing the distribution of confidence scores for the models' predictions for the missing and existing labels across all datasets. The plots show that GPT-4o exclusively gives predictions at high confidence levels, whereas YOLO-World and Grounding Dino give a wider range of confidence scores, though still showing right-skewed behavior.

terms of their confidence on both the existing labels and missing labels. This demonstrates that GPT-4o frequently provides high-confidence predictions. In contrast, both YOLO-World and Grounding DINO show a right-skewed distribution, showing that most predictions are made with low confidence. YOLO-World, in contrast to Grounding DINO, shows more predictions in the higher confidence bins. These high confidence predictions occur mostly for existing labels, but there are also more high confidence predictions of the missing labels compared with Grounding DINO. The confidence distribution of the missing labels and the existing labels overlap, for greater clarity the separate plots of the distributions can be seen in Figure 9 and Figure 10 in Appendix B.

Qualitative Comparison: For a qualitative comparison of the models, a plot of the predictions is shown in Figure 8. This sample received the highest average confidence scores for its existing labels. It can be seen that all three models are able to correctly identify the giraffe in the picture with high confidence. However, Grounding DINO also misidentifies a tree as a giraffe with low confidence. When looking at the prediction for the missing labels it can be seen that both YOLO-World and GPT-4o are able to correctly "see" that the image does not contain these missing labels. However, Grounding DINO does classify a number of objects in the image even though they are not present. These predictions of Grounding DINO do showcase that the model is uncertain about its predictions due to the low confidence scores. More examples can be found in Appendix E.

GPT-4o Response Error: The prompt used for receiving the prediction of GPT-4o can be found in Appendix C. While this prompt resulted in responses in the desired format, GPT-4o remains a Large Language Model (LLM) which occasionally results in hallucinations. Therefore, some images were not processed correctly as the model deviated from the instructions. In Table 3, the accuracy scores can be found for each dataset. For example, GPT-4o provided predictions for 89.2% of the 2,000 samples in the LVIS dataset, with half of the predictions corresponding to existing labels and the other half to missing labels.

Table 3: GPT-4o prompt quality and output consistency. The success rate indicates the percentage of API calls that returned a valid response.

Datasets	**Success rate**
LVIS	89.2%
Open Images	81.6%
Japanese Uncertainty Scenes	76.9%

4 Discussion

A comparison of the three models reveals that GPT-4o exhibits the poorest performance in terms of uncertainty. GPT-4o is notably overconfident and demonstrates poor calibration in representing its uncertainty. Despite achieving the highest AUC score, GPT-4o records the lowest mAP score. This is due to GPT-4o's tendency to generate predictions with high confidence in combination with its inability to correctly localize objects. This contributes

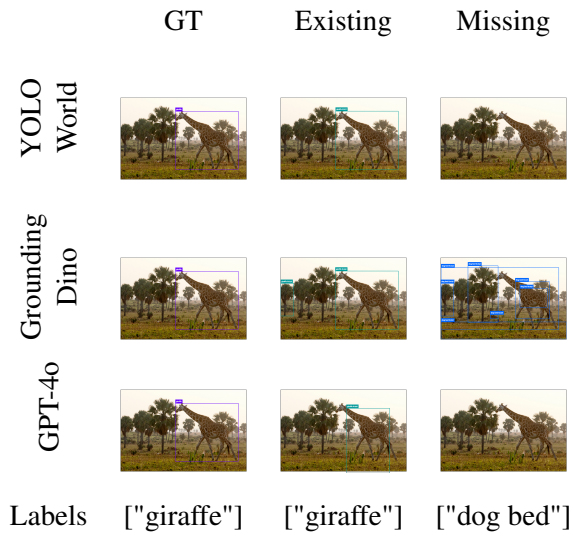

Figure 8: Visual comparison of ground truth (GT), detections of existing labels (Existing), and detections of missing labels (Missing) across different models with the highest average confidence for the existing labels of the LVIS dataset. All three models successfully detect the giraffe in the image, however, Grounding Dino also mislabels a tree. Above all, Grounding Dino is the only model that makes predictions for the missing classes.

to the overconfidence and poor mAP score. Additionally, GPT-4o frequently hallucinates due to defiance with the API call instructions.

Analysis of YOLO-World and Grounding DINO reveals that both models exhibit slight underconfidence for the LVIS dataset. However, both models are calibrated relatively well for the Open Images dataset. In terms of mAP and AUC scores, YOLO-World outperforms Grounding DINO, as it achieves higher values in both metrics. This shows that YOLO-World is superior at recognition and localization of objects and at discriminating between classes. Furthermore, Grounding DINO is more likely to hallucinate missing classes, whilst YOLO-World more often refrains from predicting these classes as seen in Figure 6.

Lastly, even though there are some manually removed incorrect missing labels in the experiment setup, as mentioned in Appendix K. It does not influence the results, due to the large difference between performance metrics in Table 2.

5 Conclusions and Future Work

To answer RQ1, the performance of these models when faced with impossible prompts varies across models. GPT-4o rarely predicts missing labels, but when it does, it assigns them high confidence. This is undesirable, as it fails to adequately convey uncertainty, particularly in response to impossible prompts. YOLO-World and Grounding DINO show a difference in their confidence scores when faced with existing or missing labels, showcasing the ability to give uncertainty estimation. This is shown by relatively good calibration of these models, where both models are only slightly underconfident. To address RQ2, GPT-4o struggles with object localization. Grounding DINO and YOLO-World both demonstrate an ability to represent their uncertainty. However, Grounding DINO's performance is hindered by a high number of unwanted predictions. In contrast, YOLO-World achieves the best performance in object detection and the most accurate representation of uncertainty when faced with impossible prompts. To answer RQ3, we can see that both YOLO-World and Grounding Dino show good discriminability between existing and missing classes. Where both models are only slightly underconfident. In contrast, GPT-4o shows poor performance as the model is highly overconfident. GPT-4o does show a high AUC score, this is likely due to the fact that the model produces a low number of predictions. Therefore, this AUC score does not represent the performance of GPT-4o accurately. In conclusion, YOLO-World and Grounding Dino show an ability to express uncertainty, while GPT-4o fails to do so correctly. However, improvements to the uncertainty estimations can still be made. Enhancing their performance in this regard is crucial for their application in real-life high-stakes scenarios.

Limitations. An important next step is to evaluate the models on a larger dataset such as the full validation set of LVIS and Open Images, this will better assess their performance on a wider range of scenarios. We observed that some missing classes were actually present in the image but not labeled, this resulted in unwanted predictions, these were removed from the results. For future implementation robustness can be enhanced by preventing these errors. In this research there is no contrast made whether missing classes are present in the training data of a model, however, this could potentially bias results. Lastly, a limitation with the ROC curves is that these are constructed using only the true positive and false positive predictions. In object detection the true negatives and false negatives are ill defined and are consequentially not used in the ROC curves. Therefore, the AUC values should be interpreted with caution.

References

Alain Andres, Aitor Martinez-Seras, Ibai Laña, and Javier Del Ser. 2024. On the black-box explainability of object detection models for safe and trustworthy industrial applications. *Results in Engineering*, 24:103498.

Weipeng Cao, Xuyang Yao, Zhiwu Xu, Ye Liu, Yinghui Pan, and Zhong Ming. 2025. A survey of zero-shot object detection. *Big Data Mining and Analytics*, 8(3):726–750.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. 2024. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16901–16911.

Tom Fawcett. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

Tobias Groot and Matias Valdenegro-Toro. 2024. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. *arXiv preprint arXiv:2405.02917*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Agrim Gupta, Piotr Dollar, and Ross Girshick. 2019. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Haibo Jin, Peiyan Zhang, Peiran Wang, Man Luo, and Haohan Wang. 2025. From hallucinations to jailbreaks: Rethinking the vulnerability of large foundation models. *arXiv preprint arXiv:2505.24232*.

KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. 2021. Towards open world object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5830–5840.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. 2020. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*.

Yiming Li, Yi Wang, Wenqian Wang, Dan Lin, Bingbing Li, and Kim-Hui Yap. 2024. Open world object detection: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(2):988–1008.

Tsung-Yi Lin, Licheng Ma, and Serge Belongie. 2014. Coco dataset: Detection evaluation.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024a. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, and 1 others. 2024b. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer.

Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, and 1 others. 2024. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*.

Amir Rosenfeld, Richard Zemel, and John K. Tsotsos. 2018. The elephant in the room. *Preprint*, arXiv:1808.03305.

Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. A comprehensive survey of hallucination in large language, image, video and audio foundation models. *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11709–11724.

Matias Valdenegro-Toro. 2021. I find your lack of uncertainty in computer vision disturbing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1263–1272.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *Preprint*, arXiv:2310.11441.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2024. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. 2022. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*.

Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. 2023. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276.

A Broader Impact Statement

Uncertainty estimation for large models is socially relevant as it is desirable to detect hallucinations or incorrect predictions, which is needed for critical applications. This paper reveals that even open world object detection models struggle with uncertainty estimation when object classes are prompted but not present and are miscalibrated and often overconfident.

Uncertainty estimation and Computer vision models require extensive experimental validation with data representative of the use case before being used in real-world applications, and unfortunately there are no guarantees on uncertainty estimation quality and performance.

B Separate Density Histogram Confidence Scores

Figures 9 and 10 show the density plots of the confidence scores for the the existing and missing confidence scores are shown in separate figures to see the differences more clearly.

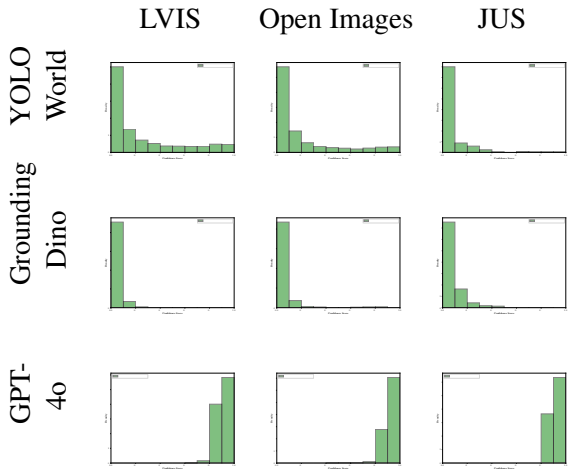

Figure 9: Density histogram showing the distribution of confidence scores for the model predictions of the existing labels across all datasets. The plots show that GPT-4o only gives high confidence predictions, whereas YOLO-World gives a broader range of scores whilst still being right skewed. Furthermore, Grounding Dino, whilst also predicting in a larger range, gives lower confidence predictions more often.

C Prompt GPT-4o

In Figure 11 the prompt used for the GPT-4o API. The prompt is designed to let GPT-4o give predictions similar to YOLO-World and Grounding Dino.

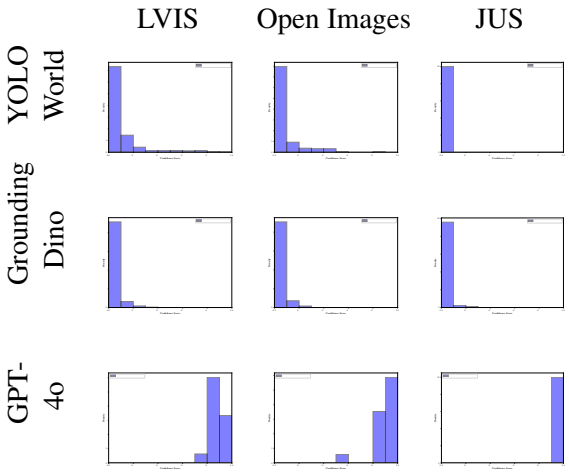

Figure 10: Density histogram showing the distribution of confidence scores for the model predictions of the missing labels across all datasets. The plots show that GPT-4o only gives high confidence predictions for the missing labels, whilst YOLO-World and Grounding Dino give predictions in a wider range but mainly giving low confidence predictions.

The json output of the response API is parsed and saved for further processing. More details about the implementation can be found in the code repository found in Appendix D.

D Code Repository

https://github.com/WoutTeillers/openworld-uncertainty-ood.git

In the GitHub repository (in the url above) the code used in the experiment can be found which include the scripts for running the models and the notebooks for visualizing the results. The scripts are executed on "python3.10.11" and the necessary libraries and their versions can be found in the "requirements.txt" file. Lastly, an API is used for the predictions of GPT-4o which requires a valid OpenAI API key.

E Qualitative comparison

This section provides examples of the prediction of the various models on the different datasets. These examples give visual insights in their performance.

Figure 16 shows the highest average confidence scores for the missing labels of the Open Images dataset. It shows that GPT-4o gives a high prediction for a missing label in the image, whereas the other two models correctly refrain from classifying the missing labels. Furthermore the Figure shows

You are an object recognition model capable of detecting and localizing objects within an image. Given an image with width = `{img_width}` and height = `{img_height}`, you will receive a list of object classes that I want you to detect.
Your task is to find all objects in the image that match these class labels and provide the following details for each object:
1. The confidence score (ranging from 0 to 1) of the detection, ensuring that only objects with a confidence score greater than or equal to `{conf_threshold}` are included.
2. The bounding box for each detected object, given by the center (x_center, y_center) of the bounding box (in pixel coordinates) and the width and height of the bounding box (in pixel width). The bounding box should tightly enclose the object and should be calculated with respect to the object's aspect ratio and position.
3. The class label for each object, corresponding to one of the classes in the provided list.
Make sure that each object is localized as accurately as possible within the image. The origin point $(0, 0)$ is at the top-left corner of the image.
The format for your response should be a JSON string like the following:

`{'scores': [], 'boxes': [[x_center, y_center, width, height]], 'labels': [String]}`

Where:
- `scores`: A list of confidence scores for each detection.
- `boxes`: A list of bounding boxes for each object, where each bounding box is a list of four values: $[x\_center, y\_center, width, height]$, representing the center of the bounding box and its dimensions.
- `labels`: A list of strings, where each string is the class label for the corresponding object in the image.
Please ensure that you only include detections that meet the confidence threshold and that the bounding boxes are as precise as possible, accurately matching the position of each object in the image. And that the output contains only the JSON without comments.

Figure 11: Prompt used for the GPT-4o API. In this prompt, the variables `img_width`, `img_height`, and `conf_threshold` are dynamically assigned based on the image dimensions and the specified confidence threshold.

the difference in the models' prediction capabilities. Figure 16 shows that GPT-4o provides only a few predictions, but with inaccurate localization. In contrast, Grounding DINO generates a larger number of predictions with varying confidence levels. YOLO-World makes correct predictions for several objects but, interestingly, fails to predict relatively clear classes, such as the wheels of the truck.

Figure 12 provides a clearer view of GPT-4o's difficulty in localizing objects within an image. This issue contributes to GPT-4o's low mAP score as seen in Table 2, as most of its predictions are false positives due to the poor localization.

Figures 13 and 14 illustrate instances where YOLO-World and Grounding DINO assign high confidence scores to missing labels. These examples reveal that the models tend to assign high confidence when there are semantic similarities between two classes or when the object class names are similar. In Figure 13, YOLO-World gives a high confidence prediction for "pickup_truck" because it shares a strong semantic similarity with 'car'. However, it assigns a lower confidence score to "pickup_truck", indicating that YOLO-World recognizes it resembles a car more than a pickup truck. Additionally, in Figure 14, the model assigns a high confidence prediction for the missing class "cutting board". This is because both "cutting board" and "surfboard" share the common feature of being "boards". Once again, the model assigns higher confidence to the existing class, demonstrating that the model predicts the object class "surfboard" as more likely.

Figure 15 shows the sample with the most missing predictions on the JUS dataset. Where only Grounding DINO incorrectly predicts the missing class "pillow". The model assigns the entire group of statues the label "pillow" in addition with a number of predictions for the individual statues. This is inline with the prediction of the existing labels.

Figure 12: Visual comparison of ground truth, detections, of the image with most false positive predictions of GPT-4o. Showing the inability of GPT-4o to correctly localize objects in an image.

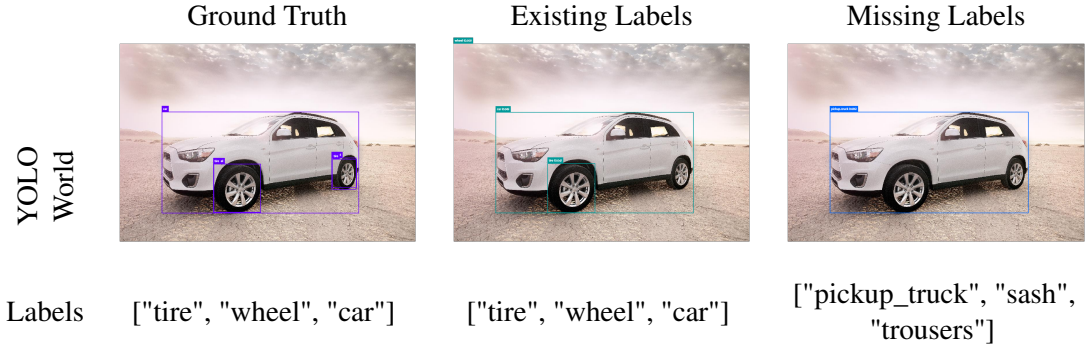

Figure 13: Visual comparison of ground truth, detections, of the image with highest average confidence for the missing predictions of YOLO-World on the Open Images dataset. YOLO-World shows a high confidence prediction for the missing label "pickup_truck", likely due to semantic similarity between the ground truth label "car". The model does show a difference in confidence for these predictions.

Grounding DINO also gives a prediction to the entire group, in combination with prediction on the individual statues. However, for the existing labels, Grounding DINO shows much higher confidence scores. Furthermore, GPT-4o gives a few high confidence predictions on the existing labels. The predictions are not properly localized and GPT-4o misses a large number of predictions for statues. Also, YOLO-World is not able to recognize any of the statues in the image. YOLO-World and GPT-4o both correctly refrain from predicting the missing label in this image.

The main findings of the qualitative comparisons are that Grounding DINO gives the most predictions of the three models, thereby missing fewer objects in the image but occasionally labeling incorrect objects. The difference between the number of predictions between the models can be seen in Figure 6. Additionally, Grounding DINO shows a clear distinction in its confidence for existing and missing labels. GPT-4o shows poor performance in image localization, with its predictions always being of high confidence, which is in line with the density distribution shown in Figure 7. Furthermore, this model does not classify all the desired objects in an image consistently. YOLO-World shows good distinguishability between existing and missing labels, but occasionally misses some of the existing ones.

F Incorrect Missing Class Labels

Table 4 and 5 the incorrect missing labels can be found for the LVIS and Open Images dataset. The predictions for these labels are removed from the specific images to ensure that the predictions are for objects that are actually not present.

Figure 17 shows this problem, the image contains a numerous labeled object classes, but the class of "power plugs and sockets" is not labeled in the image resulting in this complication.

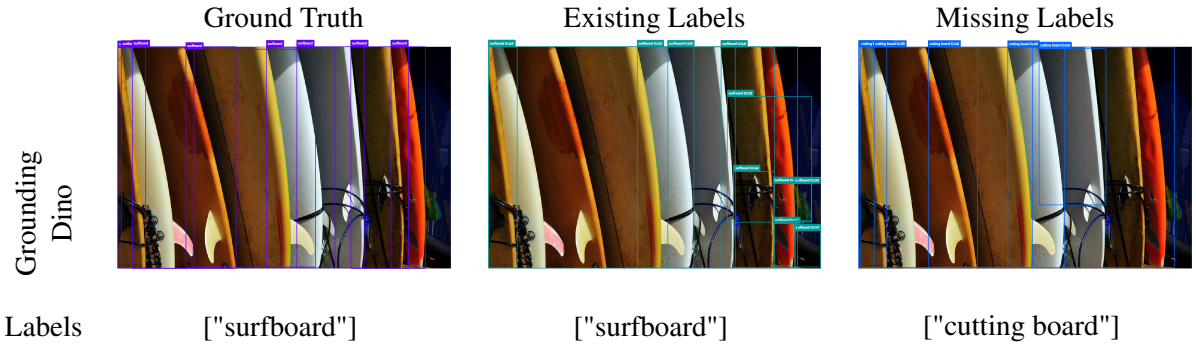

Figure 14: Visual comparison of ground truth, detections, of the image with highest average confidence for the missing predictions of Grounding Dino on the LVIS dataset. Grounding Dino gives high confidence predictions for the missing labels, likely due to the fact that both the ground truth label "surfboard" and the missing label "cutting board" are both type of "boards". The models does show difference in confidence level for these predictions.

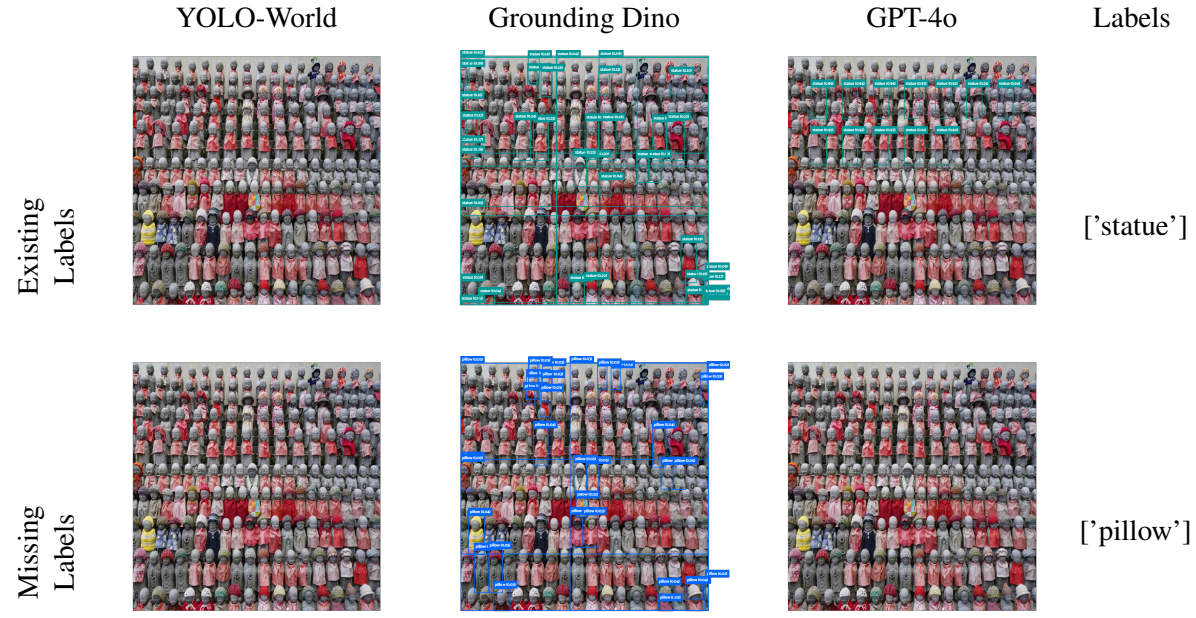

Figure 15: Visual comparison of ground truth, detections, and missing labels across the different models with the highest average confidence for the missing labels of the Japanese Uncertainty Scenes dataset. YOLO-World gives no predictions for both the existing and missing labels, whereas Grounding DINO gives a high number of predictions for both. GPT-4o shows a small number of incorrectly localized predictions for the existing labels and no prediction for the missing labels.

Figure 16: Visual comparison of ground truth, detections, and missing labels across the different models with the highest average confidence for the missing labels of the Open Images dataset. The three models show notably differences in their predictions for the existing labels and GPT-4o is the only model that gives a prediction for the missing labels.

G Common Classes Removed from Class List

A list of common classes were removed from the list of classes as these classes are expected to be found in the images even though they might not be labeled. In Table 6 the list of classes that was removed can be found for the LVIS, Open Images and Japanese Uncertainty Scenes (JUS) datasets. Note that there were no classes removed from the JUS dataset.

H Class labels for Japanese Uncertainty Scenes

The ground truth values for the Japanese Uncertainty Scenes were manually assigned. In Table 7 the ground truth labels for each image can be found. The JUS dataset can be found at `https://github.com/ML-RUG/jus-dataset`.

I Prediction per subset of Labels

Figure 18 illustrates model predictions on a subset of the ground truth classes. In this figure predictions are shown where one class is added to the prompt at each step. This shows how the models make predictions based on the desired labels. It is evident that the models can give incorrect labels with low confidence when the correct label is not available yet. Whenever the label gets prompted it can be seen that the model changes its prediction.

J Hyperparameters

Running the object recognition models there is one key hyperparameter to tune, the confidence level threshold. This threshold indicates above which confidence level predictions are accepted. Hereby, allowing to manually tune the most optimal threshold to allow the correct predictions to pass and unwanted prediction to be omitted. During this experiment, the lower predictions give insights into the model's confidence level. Therefore, in the experiment this confidence threshold is set to a small value to achieve a wide range of confidence levels. The threshold is set to get almost all predictions to receive data on the performance of the models. This confidence level can not be set to 0, due to the fact that all prediction will be showcased, resulting in too much unwanted predictions. For consistency across models, this hyperparameter is set to 0.02 for all models, ensuring that predictions fall within the

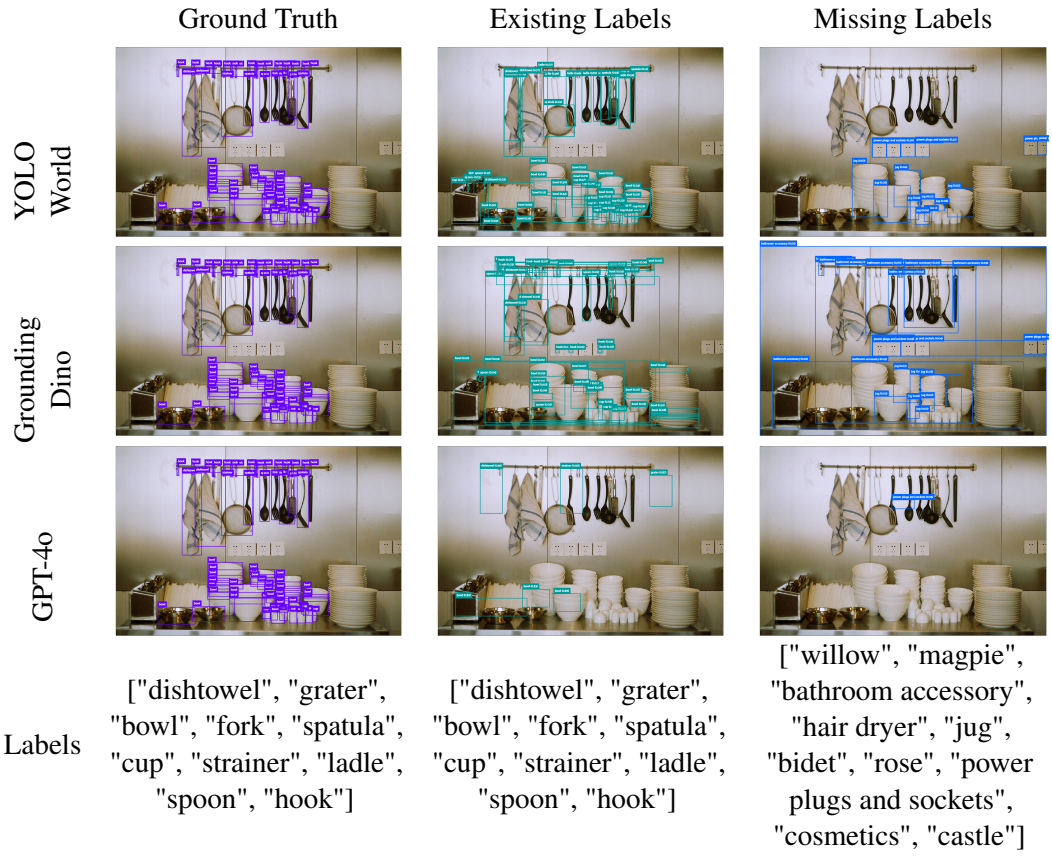

Figure 17: Visual comparison of ground truth, detections, and missing labels across the different models showcasing an incorrect missing label in the image. Showing that classes that are not present in the ground truth can be present in the image. This can lead to incorrect labels in the missing classes list such as "power plugs and sockets" in this example.

same confidence range (0.02-1.00) and minimizing the number of unwanted predictions.

K Post Processing

Running the Grounding DINO base model on the datasets, resulted in some unwanted predictions. The model occasionally gave prediction where a new combined label was used. For example given the two classes ["dog", "bed"], the model could give a prediction of "dog bed". These predictions were removed as they were not of interest in this study.

Furthermore, the Grounding DINO base model did not make use of Non-maximum Suppression (NMS). This lead to the model giving many overlapping predictions. Due to the fact that YOLO-World did make use of NMS, it was decided to implement the NMS algorithm for the predictions made by Grounding-DINO. This to make sure of a fair comparison between the models. This study used the same IOU value as YOLO-World for a fair comparison. YOLO-World uses an IOU threshold of 0.5, so this value was sub-sequentially used for Grounding DINO. NMS filters predictions of the same label with an IOU value exceeding the specified threshold, retaining the prediction with the higher confidence score.

As GPT-4o did not have many overlapping detections, NMS was not applied to the predictions of this model.

A minor issue with the experiment setup is that while requiring the list of missing classes for each image, it sometimes occurs that the object of that class are present in the image, even-though it was not labeled in the original dataset. This will cause the models to recognize the object with high confidence which reduced the reliability of this experiment, as the experiment assumes that these missing classes are not present in the image. To resolve this, the predictions of the missing labels are manually checked for validity and the missing classes that are present in the image are removed from the predictions. This causes a slight imbalance between the number of existing classes and missing classes

Figure 18: Subset of ground-truth classes evaluated with YOLO-World and Grounding Dino, illustrating how the models reassigns object labels when it predicts higher confidence for an alternate class.

in an image. However, this problem only occurs for 4.3% of all the tested images. The prediction of classes that were removed of specific images can be found in Appendix F, including a visual example that illustrates this issue.

Table 4: Incorrectly retrieved missing labels of the LVIS dataset. The predictions of each label have been removed from the specified image.

Image	Missing LVIS Labels
000000192722.jpg	['footwear']
000000569652.jpg	['power plugs and sockets']
000000452334.jpg	['home appliance', 'kitchen appliance']
000000006777.jpg	['stairs']
000000269417.jpg	['kitchen appliance']
000000394879.jpg	['coffeemaker', 'dessert']
000000375317.jpg	['gas stove', 'home appliance']
000000244965.jpg	['mammal']
000000125247.jpg	['candy']
000000090122.jpg	['kitchen utensil']
000000175205.jpg	['dairy product']
000000290911.jpg	['vehicle']
000000497875.jpg	['tree']
000000424044.jpg	['baked goods', 'mixing bowl']
000000216863.jpg	['picture frame']
000000051618.jpg	['tableware']
000000263589.jpg	['building']
000000044611.jpg	['trousers']
000000048432.jpg	['land vehicle']
000000277858.jpg	['mammal']
000000205055.jpg	['building', 'footwear']
000000406013.jpg	['plastic bag']
000000534751.jpg	['furniture']
000000244157.jpg	['tree']
000000094052.jpg	['microwave oven', 'picture frame', 'food']
000000471842.jpg	['kitchenware']
000000261893.jpg	['bicycle wheel']
000000364210.jpg	['baked goods']
000000442298.jpg	['furniture', 'mammal']
000000468917.jpg	['microwave oven']
000000362140.jpg	['fashion accessory', 'picture frame']
000000555273.jpg	['picture frame']
000000127100.jpg	['footwear']
000000372980.jpg	['mammal']
000000043692.jpg	['countertop', 'furniture']
000000315902.jpg	['furniture']
000000505152.jpg	['kitchen utensil']
000000223032.jpg	['mammal']
000000070164.jpg	['remote control']
000000356153.jpg	['food']
000000334352.jpg	['sports equipment']
000000455691.jpg	['countertop']
000000526794.jpg	['home appliance']
000000086208.jpg	['snack']
000000103223.jpg	['home appliance']
000000501247.jpg	['land vehicle']
000000160142.jpg	['furniture']
000000521200.jpg	['hand dryer']
000000040930.jpg	['window']
000000120527.jpg	['sports equipment']
000000460442.jpg	['fruit', 'footwear']
000000423161.jpg	['flower']
000000053037.jpg	['coffeemaker', 'microwave oven']
000000218751.jpg	['land vehicle']
000000361497.jpg	['aircraft']
000000278303.jpg	['mammal']
000000095841.jpg	['footwear']
000000018090.jpg	['footwear']
000000457737.jpg	['furniture']
000000404698.jpg	['power plugs and sockets']
000000513604.jpg	['tin can']
000000019441.jpg	['tableware']
000000442875.jpg	['furniture']
000000007288.jpg	['bidet']
000000071726.jpg	['soap dispenser', 'cabinetry']

Table 5: Incorrectly retrieved missing labels of the Open Images dataset. The predictions of each label have been removed from the specified image

Image	Missing OI Labels
00141571d986d241.jpg	['hand_towel', 't-shirt']
00146ba1e50ed8d8.jpg	['cylinder']
0035c28612c035fd.jpg	['green_bean']
00acf53b127218c2.jpg	['radiator_grille']
00dc0530e6779ca6.jpg	['baby_buggy']
01491bf840ae9939.jpg	['activewear']
015f5cd905204962.jpg	['trousers']
0197df7725980004.jpg	['rearview_mirror']
01b405e0cab3add3.jpg	['baseball_cap']
01f26ca52e27a8d9.jpg	['pencil_case']
023a57536e17b7b1.jpg	['figurine']
025ffa27eb2ba851.jpg	['printing_machine']
030033e1b4137e3b.jpg	['dog_collar']
03650b9fde97f523.jpg	['wristwatch']
049720d842de2d3e.jpg	['paper_towel']
04d9284ebdc41aeb.jpg	['cordial']
04ec0b057014a648.jpg	['jockey_cap']
006f87bf928f9ba3.jpg	['jewellery']
00c9616a917be867.jpg	['fin_(footwear)']
01c79b8cc239037d.jpg	['wedding_ring']
038ee0bf31929792.jpg	['flip-flop_(sandal)']
05d69a9470032674.jpg	['sport_shirt']

Table 6: List of common classes removed from list of classes.

Class set	List of removed classes
$\mathcal{C}_{\text{LVIS}}$	['human arm', 'human beard', 'human body', 'human ear', 'human eye', 'woman', 'man' 'human face', 'human foot', 'human hair', 'human hand', 'human head', 'human leg', 'human mouth', 'human nose']
$\mathcal{C}_{\text{OI}}$	['human']
$\mathcal{C}_{\text{JUS}}$	[]

Table 7: Ground truth labels of the images in Japanese Uncertainty Scenes dataset.

Filename	**List of ground truth class labels**
20180728_204527.jpg	['food']
20180729_152751.jpg	['dessert']
20180730_132300.jpg	['sushi']
20180801_203824.jpg	['okonomiyaki', 'food']
20180808_220402.jpg	['sushi']
20180812_145111.jpg	['egg fried rice', 'fried chicken', 'gyoza', 'japanese food']
20180814_181327.jpg	['person', 'food']
DSC01703.jpg	['drawing', 'animal']
DSC01754.jpg	['octopus', 'person']
DSC01796.jpg	['paper lanterns']
DSC01851.jpg	['statue']
DSC01874.jpg	['tombstone']
DSC02396.jpg	['fish']
DSC02711.jpg	['bird', 'building']
DSC02941.jpg	['lantern', 'building', 'paper lanterns']
DSC02960.jpg	['lamp', 'paper lanterns']
DSC03113.jpg	['gate', 'torii']
DSC03256.jpg	['gate', 'torii']
DSC03391.jpg	['bamboo tree']
DSC03397.jpg	['bus', 'bridge']
DSC04631.jpg	['wooden plaque', 'ema']
DSC04742.jpg	['tree', 'building']
DSC04746.jpg	['building']
DSC04796.jpg	['bridge']
DSC04858.jpg	['boat', 'mountain']
DSC05168.jpg	['tree', 'painting']
DSC05403.jpg	['person', 'tree']
DSC05406.jpg	['person', 'tree']
DSC05439.jpg	['warrior']
DSC05535.jpg	['japanese food']
DSC06071.jpg	['tree']
DSC07641.jpg	['coach', 'train']
P9250145.jpg	['tokyo tower']
P9250156.jpg	['building']
P9280467.jpg	['light poles', 'signs']
P9301022.jpg	['building', 'tree']
PA011063.jpg	['volcano']
PA011245.jpg	['statue', 'animal']
PA041459.jpg	['building', 'tree']