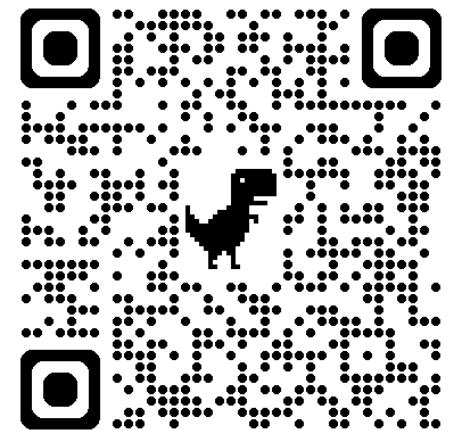


GPT-4oにおける物体検出スコアの測定

ソースコードやスライドは以下のGithubに公開しています
https://github.com/Tatsuya736482/J-CRe3_gpt_4o_Public.git



マルチモーダル参照解析

動画または画像　テキスト



図2 「こちらに捨てておきますね。」という発話に対する phrase grounding モデルの解析例。黄色の物体矩形が正解データを表し、それ以外はシステムの出力である。システム出力にはシステムの予測確率が記載されている。

テキスト中の名刺や述語
と画像データ
の参照関係

出典:実世界における総合的参照解析を目的とした
マルチモーダル対話データセットの構築

マルチモーダル参照解析

- **テキスト間照応解析**

テキスト中の単語や句の間に存在する照応・共参照関係を解析するタスク

- **物体検出**

画像中から参照されている物体が存在する領域を特定するタスク

- **テキスト・物体間参照解析**

テキスト間照応解析における照応・共参照の対象を物体検出によって特定された物体領域から選択するタスク

出典:実世界における総合的参照解析を目的としたマルチモーダル対話データセットの構築

マルチモーダル参照解析

- **テキスト間照応解析**

テキスト中の単語や句の間に存在する照応・共参照関係を解析するタスク

- **物体検出**

画像中から参照されている物体が存在する領域を特定するタスク

- **テキスト・物体間参照解析**

テキスト間照応解析における照応・共参照の対象を物体検出によって特定された物体領域から選択するタスク

出典:実世界における総合的参照解析を目的としたマルチモーダル対話データセットの構築

J-CRe3データセット

- **テキスト間照応解析**

テキスト中の単語や句の間に存在する照応・共参照関係を解析するタスク
→テキスト間照応アノテーション

(テキストに述語項構造・共参照・橋渡し照応関係を付与)

- **物体検出**

画像中から参照されている物体が存在する領域を特定するタスク
→物体領域アノテーション

(動画から抽出された全フレームについて、物体に物体矩形を付与)

- **テキスト・物体間参照解析**

テキスト間照応解析における照応・共参照の対象を物体検出によって特定された物体領域から選択するタスク

→テキスト・物体間参照アノテーション

(テキスト中の名詞句および述語と、画像中の物体矩形のすべての組み合わせについて参照関係を付与)

出典:実世界における総合的参照解析を目的としたマルチモーダル対話データセットの構築

GPT-4oによる実験(物体検出)

方法

使用言語: python 3.10.14

使用モデル: gpt-4o

使用したデータ: J-CRe3/20230426-57195279-1/'001'-'083'

プロンプト: 質問と画像を与え、フォーマットをjson形式で指定した(後述)

GPT-4oによる実験(物体検出)

プロンプト:質問と画像を与え、フォーマットをjson形式で指定した
質問:

"Given an image, identify the objects as many as possible even with low confidence scores, and for each detected object, extract its name, corresponding bounding box coordinates, and the associated probability scores"

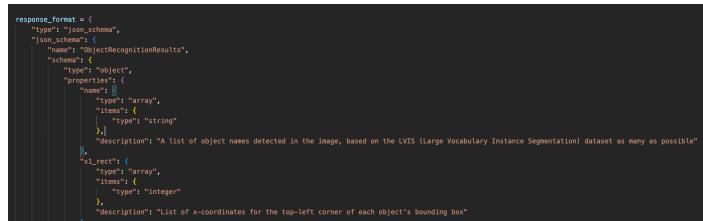
GPT-4oによる実験(物体検出)

プロンプト:質問と画像を与え、フォーマットをjson形式で指定した
フォーマット:(json形式で型と説明を書き、出力フォーマットを指定できる)

```
response_format = {
    "type": "json_schema",
    "json_schema": {
        "name": "ObjectRecognitionResults",
        "schema": {
            "type": "object",
            "properties": {
                "name": {
                    "type": "array",
                    "items": {
                        "type": "string"
                    },
                    "description": "A list of object names detected in the image, based on the LVIS (Large Vocabulary Instance Segmentation) dataset as many as possible"
                },
                "x1_rect": {
                    "type": "array",
                    "items": {
                        "type": "integer"
                    },
                    "description": "List of x-coordinates for the top-left corner of each object's bounding box"
                }
            }
        }
    }
}
```

GPT-4oによる実験(物体検出)

プロンプト:質問と画像を与え、フォーマットをjson形式で指定した



“name”型:array(string)

: "A list of object names detected in the image, based on the LVIS (Large Vocabulary Instance Segmentation) dataset, including as many as possible, even with low confidence scores."

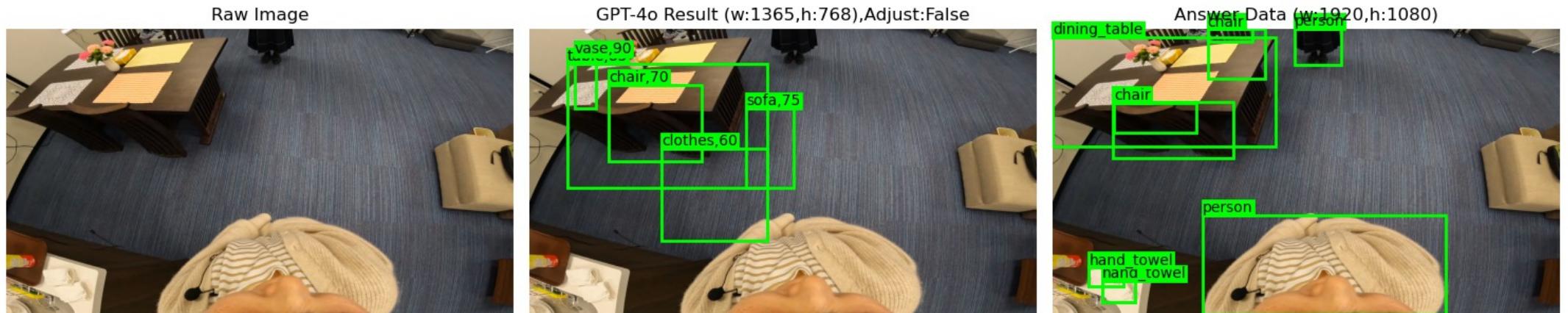
“x1_rect”型:array(integer)

: "List of x-coordinates for the top-left corner of each object's bounding box"

“probability”型:array(integer)

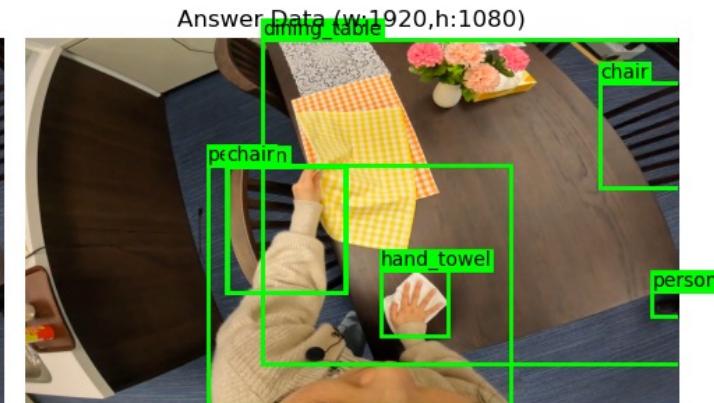
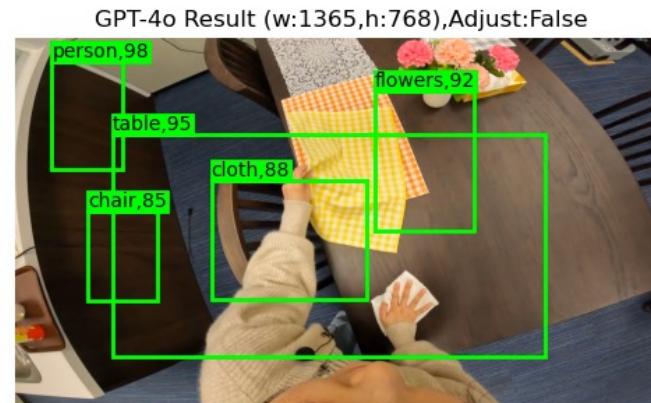
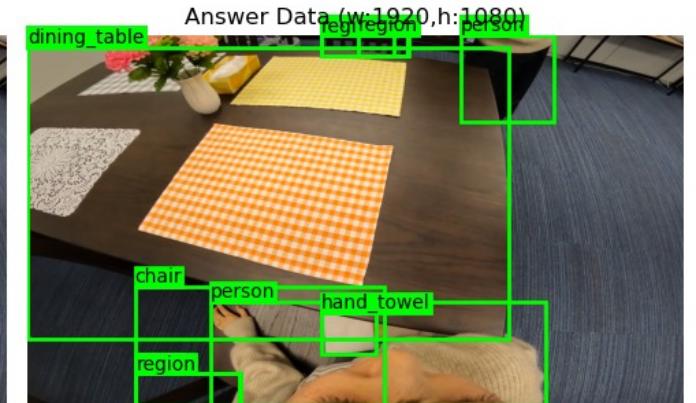
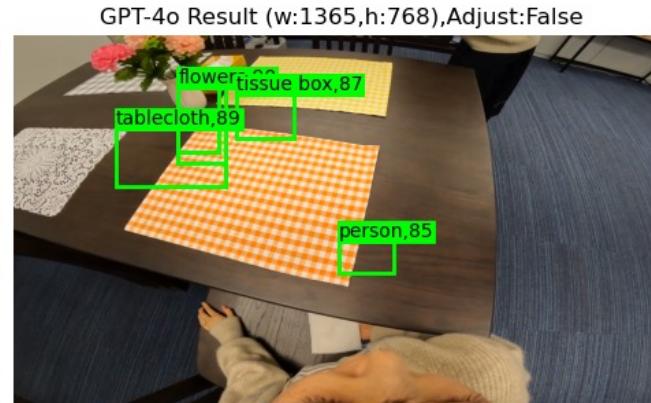
: "An array of probability values, each representing the likelihood of the corresponding object being correctly detected."

GPT-4oによる実験(物体検出)



table, chairs, sofaなど画像中に存在するものが输出されている
→ 物体認識自体は出来ている様子
バウンディングボックスの出力は的外れ(位置がずれている)
→ 座標情報の出力は出来ない?

GPT-4oによる実験(物体検出)

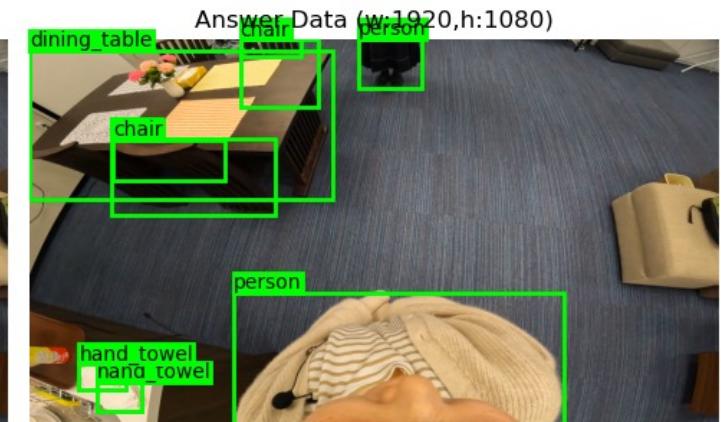
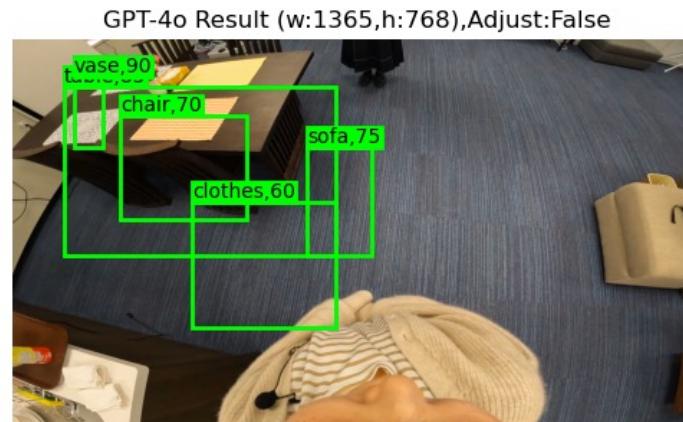


GPT-4oによる実験(物体検出)

仮説:

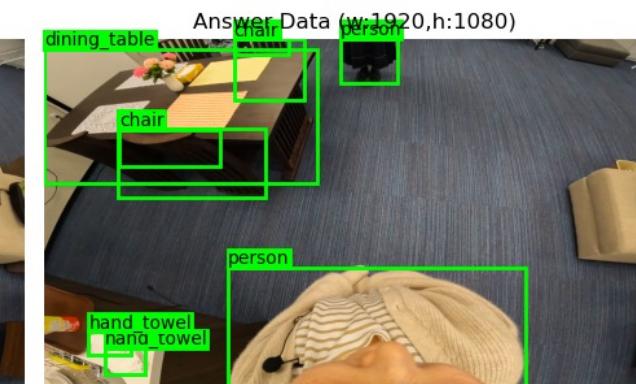
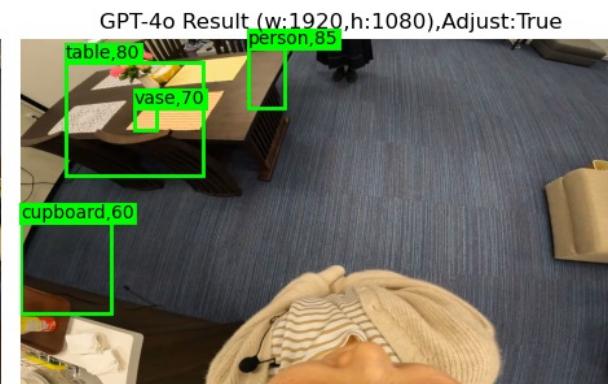
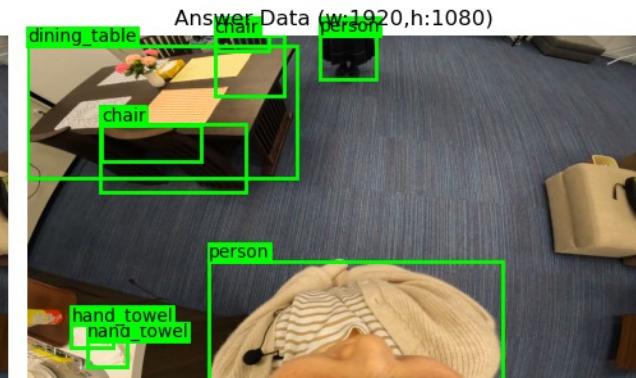
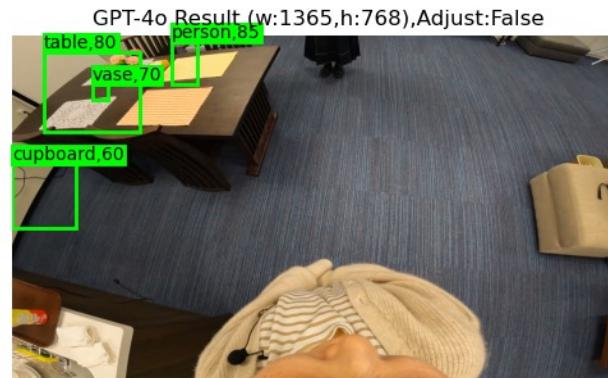
「gpt-4oのプロセスによって画像のサイズが変わってしまい、バウンディングボックスの出力と元画像のサイズが合っていない」ためなのではないか

→プロンプトとして画像サイズも付け加えたところ縦横比は変わらないが、画像サイズが小さくなっていることが分かった。



GPT-4oによる実験(物体検出)

そこで、画像サイズが元画像に合うようにバウンディングボックスを調整すると以下のようにになった。



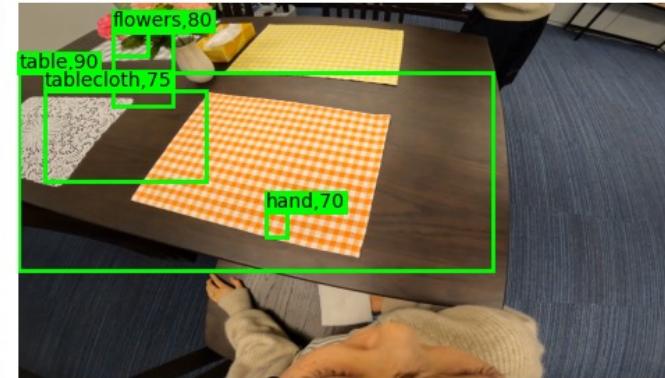
しかし、それでもバウンディングボックスのずれは直らなかった。

GPT-4oによる実験(物体検出)

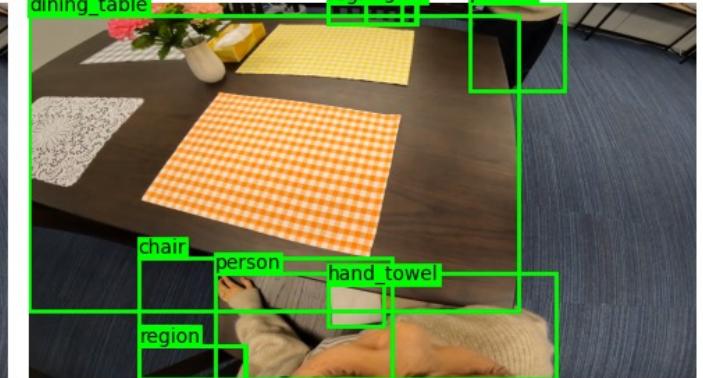
Raw Image



GPT-4o Result (w:1365,h:768),Adjust:False



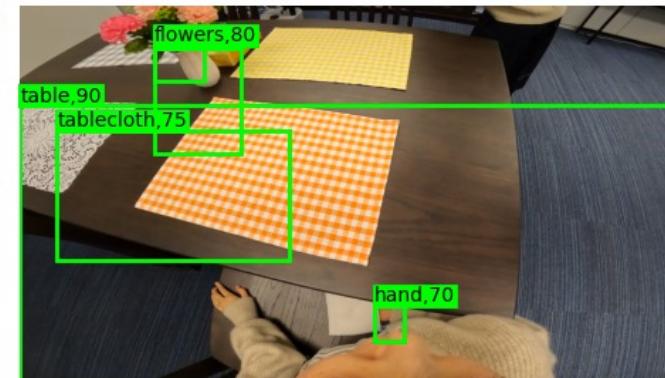
Answer Data (w:1920,h:1080)



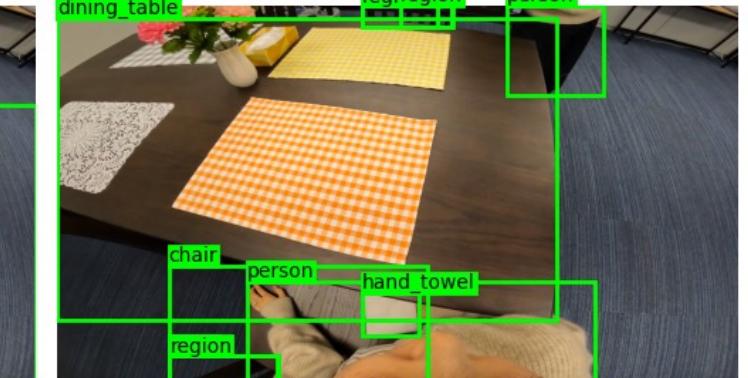
Raw Image



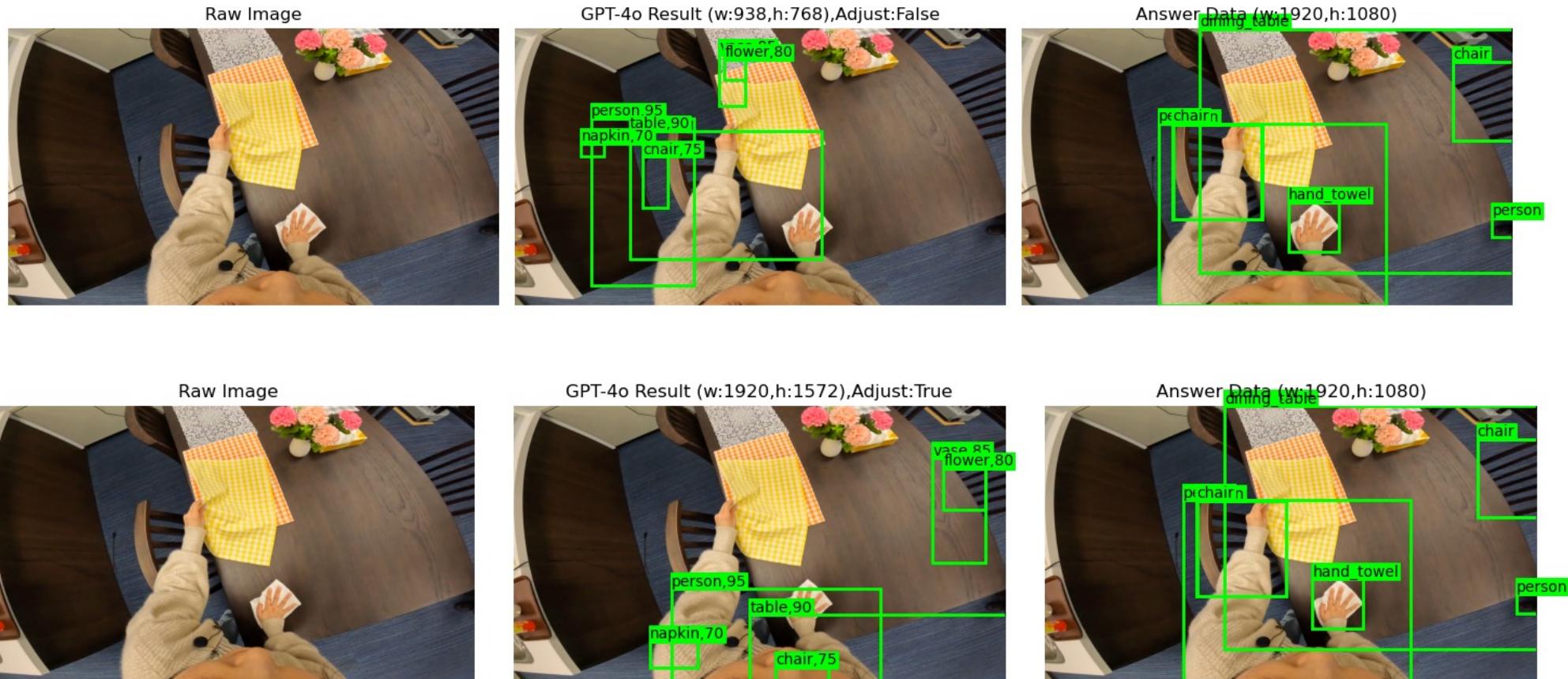
GPT-4o Result (w:1920,h:1080),Adjust:True



Answer Data (w:1920,h:1080)

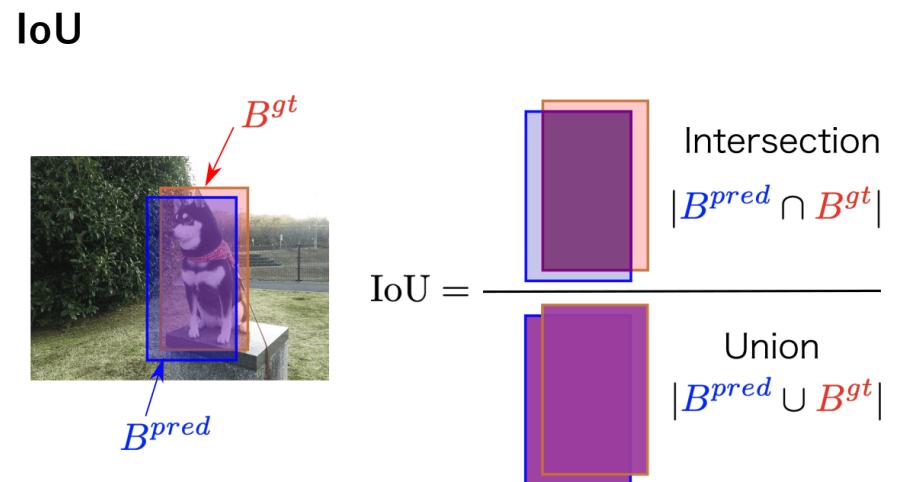
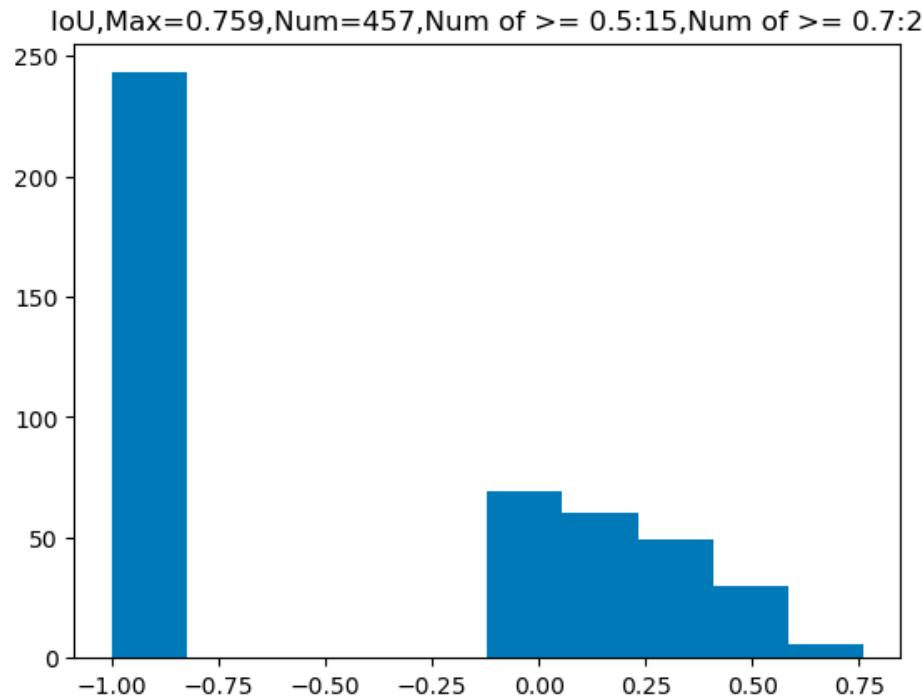


GPT-4oによる実験(物体検出)



GPT-4oによる実験(物体検出)

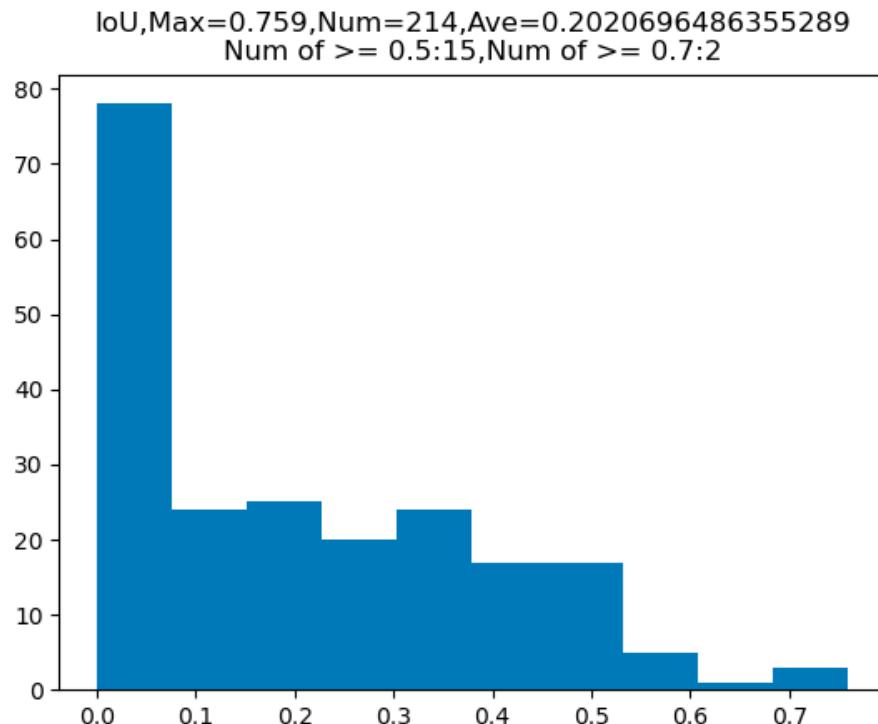
83個の画像についてIoUのデータをヒストグラム表示すると以下のようになる。尚正解データにない物体は-1の値を取るように設定した



出典:<https://cvml-expertguide.net/terms/dl/object-detection/iou/>

GPT-4oによる実験(物体検出)

83個の画像についてIoUのデータをヒストグラム表示すると以下のようになる。尚正解データにない物体は-1の値を取るように設定した



- ・データセットは発言に関係のない物体の正解データを含んでいないので、IoU=-1となっている部分は誤認識なかどうかは不明
- ・-1をとるもの除去すると214個なので、0.5以上の値をとるサンプルは約7%

GPT-4oによる実験(物体検出)

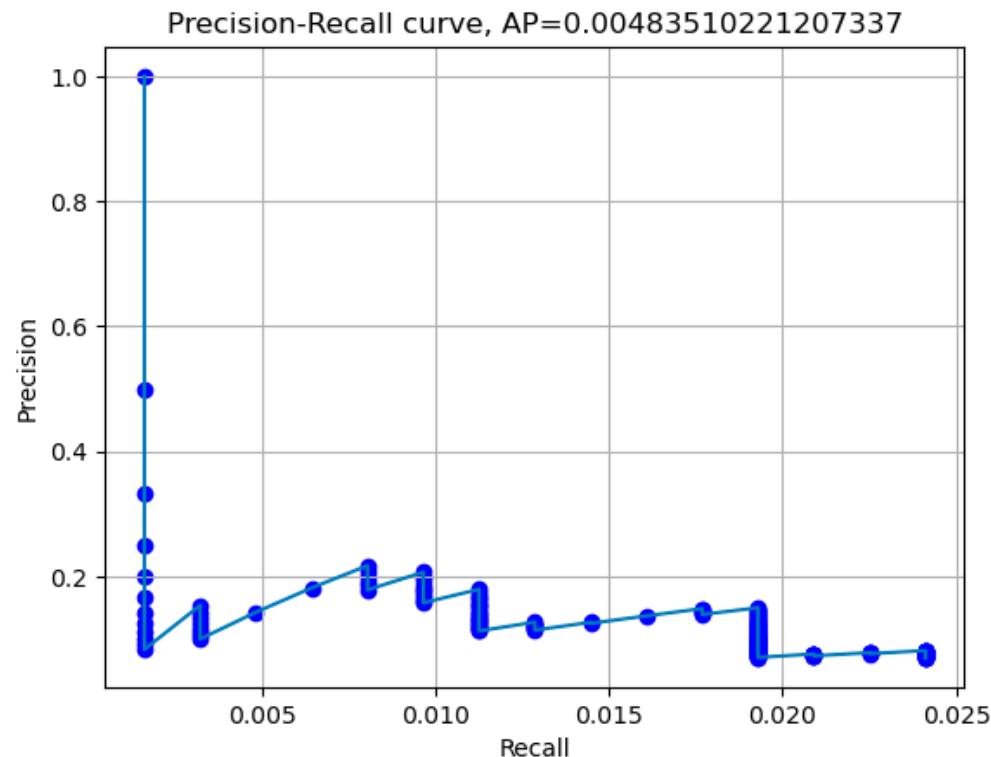
Precision(精度) = $TP / (TP+FP)$
 = 正解数 / サンプル数
 (モデルが物体と検出したものの中で正解の割合)

Recall(再現率) = $TP / (TP+FN)$
 = 正解数/正解データの数
 (正解の中からモデルが認識した割合)

AP (Average Precision) は、Precision-Recallカーブを積分した部分で、
全ての再現率に関する平均

GPT-4oによる実験(物体検出)

正解データが見つからなかったデータは除外してRecall, Precision, APを計測
信頼度順に並べてPrecision-Recall カーブを作成



と非常に低い値になった。

Prompting の工夫によるスコアの改善

調べたところMicrosoftの論文でSet-of-Mark (SoM) promptingという方法が見つかった。これは以下のように全ての物体に番号を付ける処理をした後でGPTに入力するというプロンプトの手法である

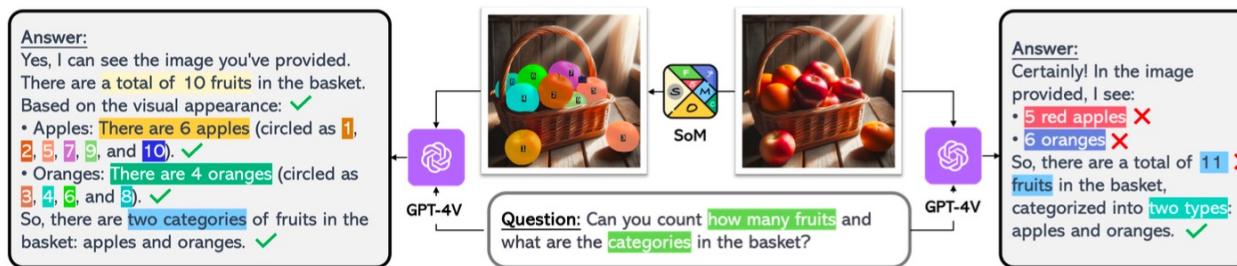


Figure 2: Comparing standard GPT-4V and its combination with *Set-of-Mark (SoM) Prompting*. it clearly shows that our proposed prompting method helps GPT-4V to see more precisely and finally induce the correct answer. We highlight the differences between our method and the standard one. (The image is generated by Dalle-3 and is better viewed in color.)

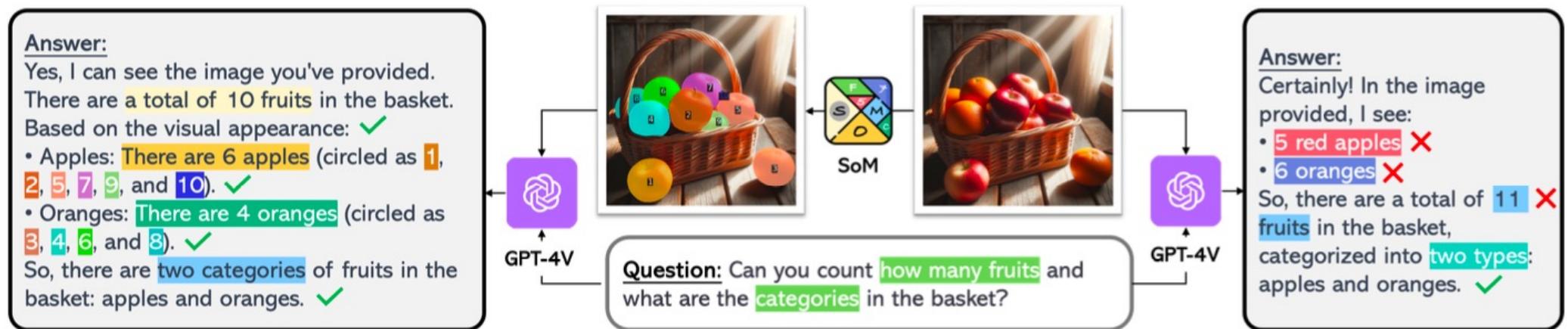
出典: Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V
<https://github.com/microsoft/SoM>

Prompting の工夫によるスコアの改善



出典: Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V
<https://github.com/microsoft/SoM>

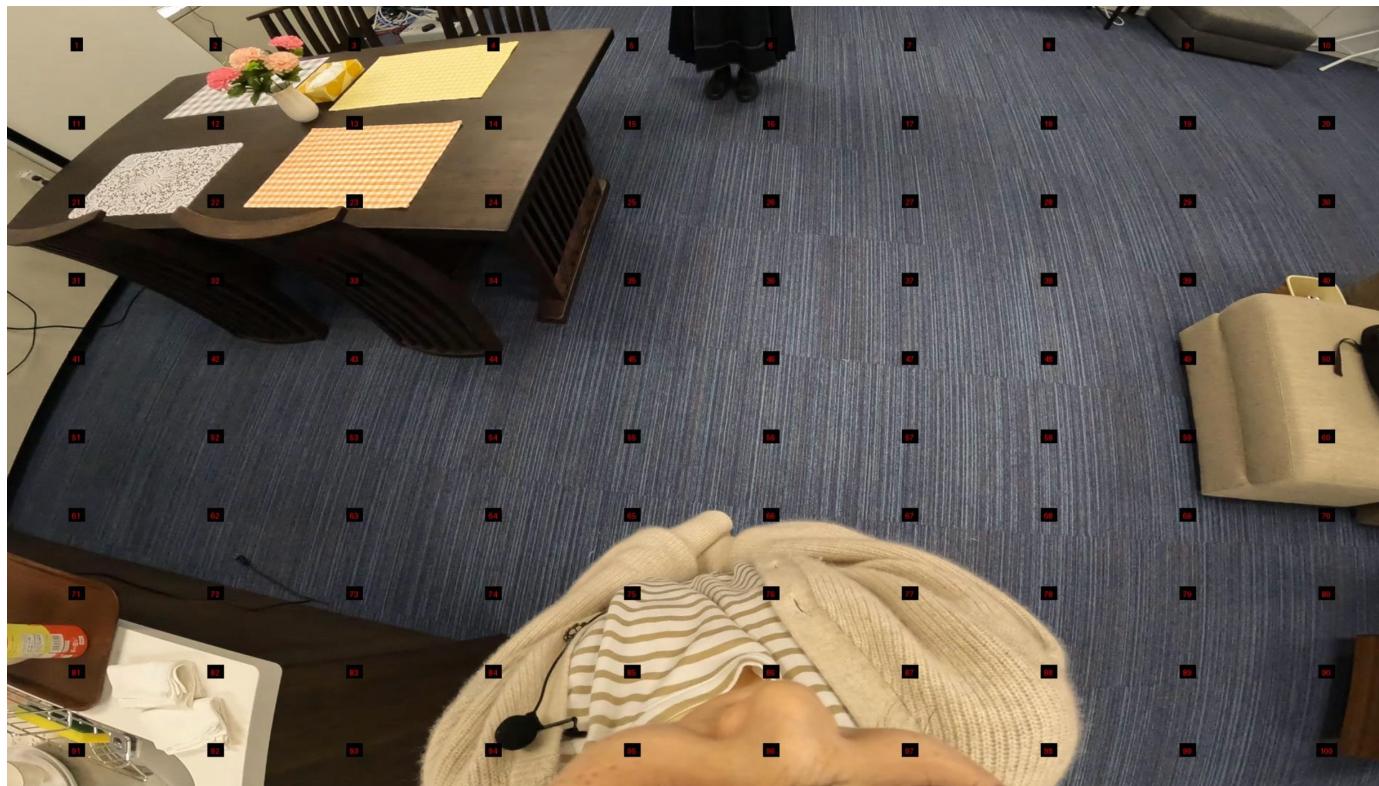
Prompting の工夫によるスコアの改善



出典: Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V
<https://github.com/microsoft/SoM>

Prompting の工夫によるスコアの改善

簡易版として、画像をグリッドに分けてその中央に数字ラベルを貼り付けてSOM promptingを実装した。



Prompting の工夫によるスコアの改善

簡易版として、画像をグリッドに分けてその中央に数字ラベルを貼り付けてSOM promptingを実装した。

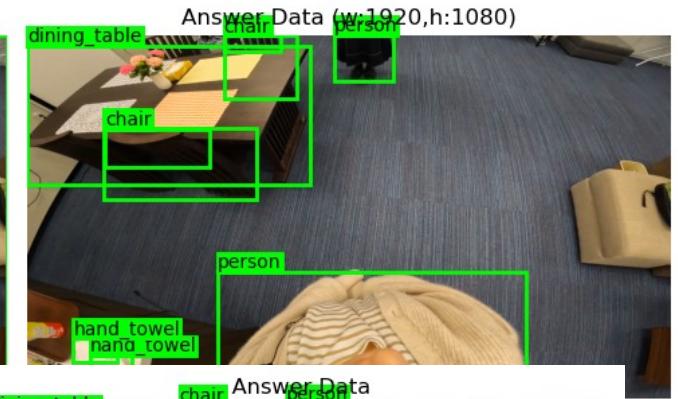
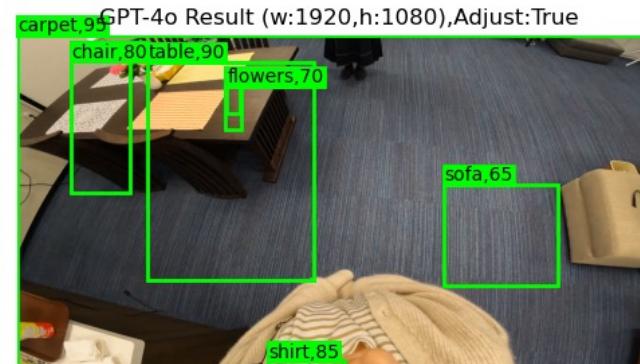
質問:

Given an image, identify as many objects as possible. For each detected object, provide its name, **grid cell numbers** (bounding box), and associated probability scores. Note that the image is divided into a 10x10 grid. Grid cells are numbered starting from the top-left corner as 1, increasing from left to right. Upon reaching the rightmost cell, numbering continues from the leftmost cell of the next row down. This pattern continues until the bottom of the grid is reached.

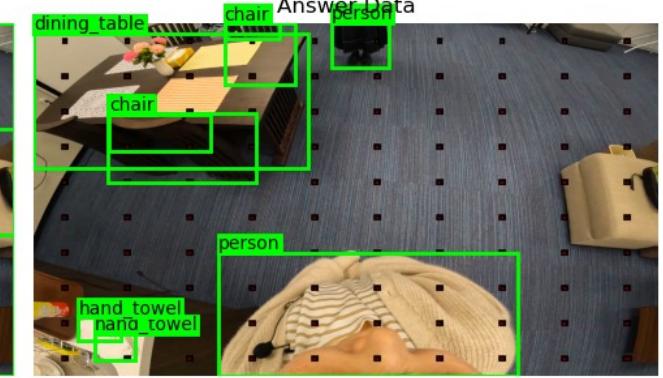
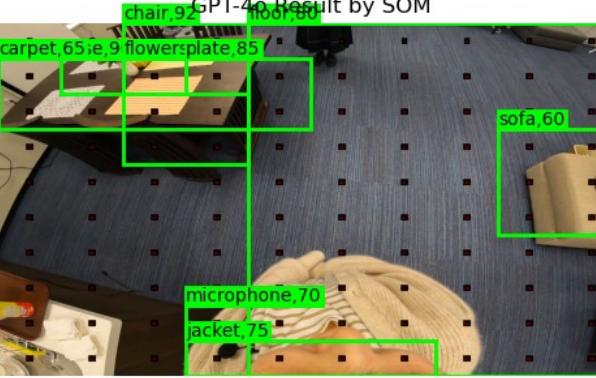
Prompting の工夫によるスコアの改善

少し精度が上がったように感じる

S
O
M
な
し

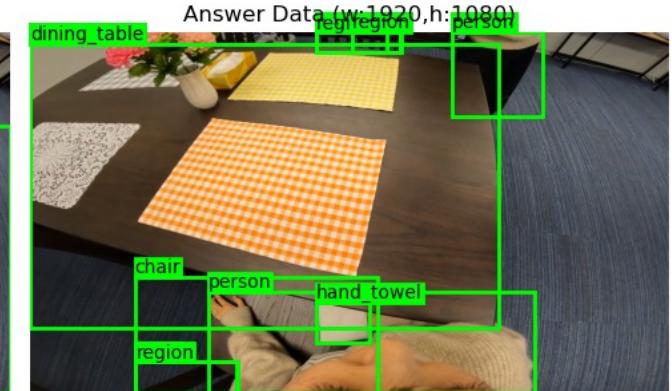
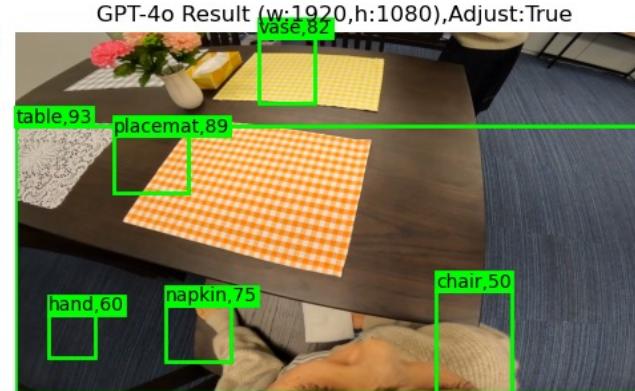


S
O
M
あ
り

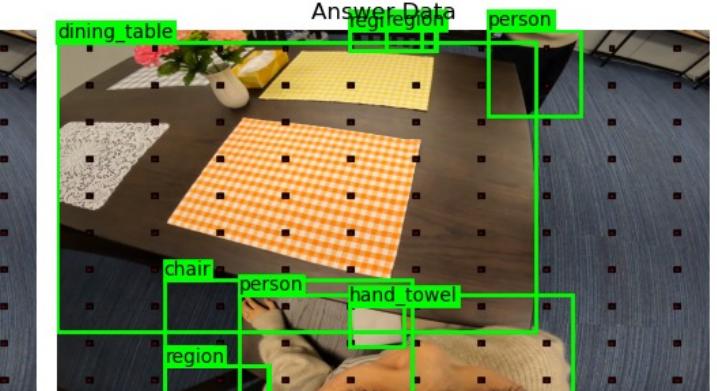
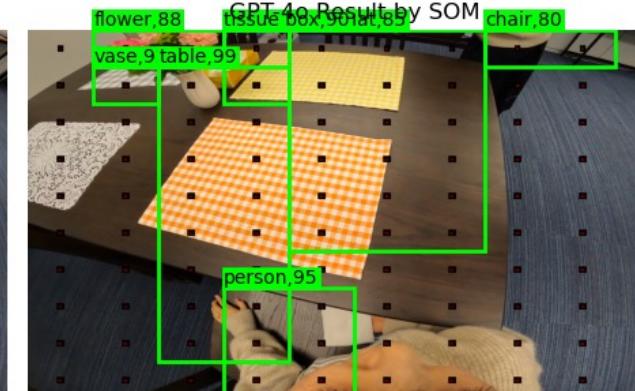


Prompting の工夫によるスコアの改善

S
O
M
な
し

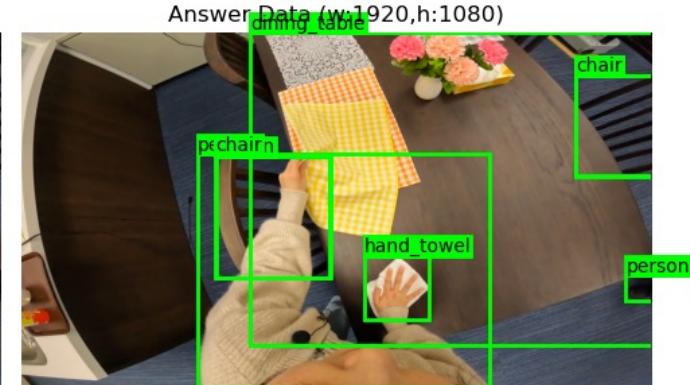
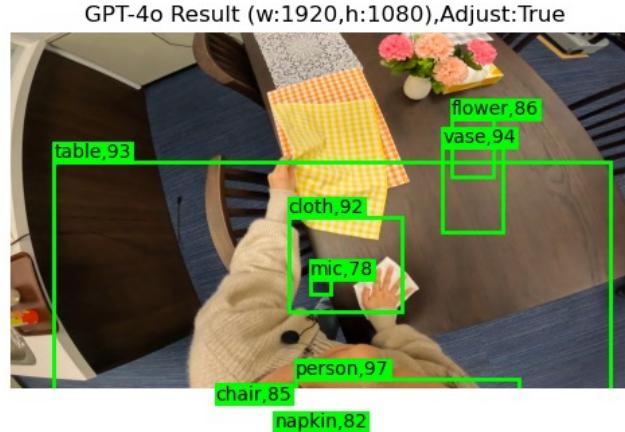


S
O
M
あ
り

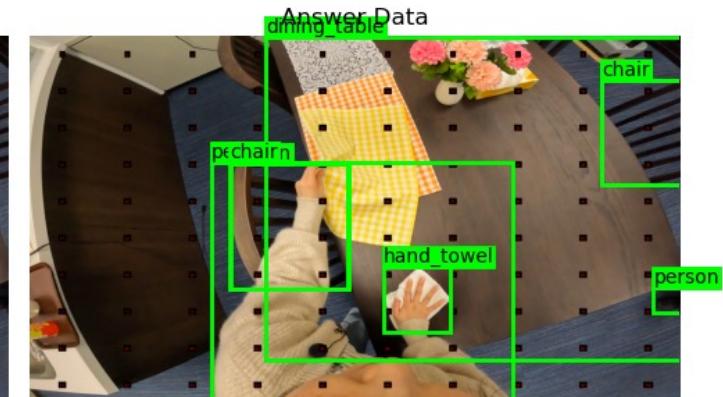
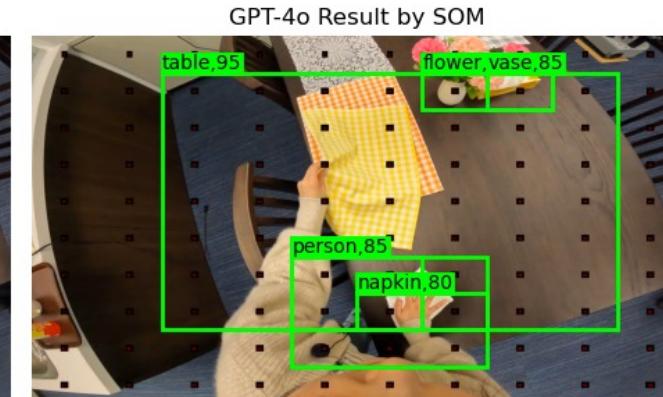


Prompting の工夫によるスコアの改善

S
O
M
なし

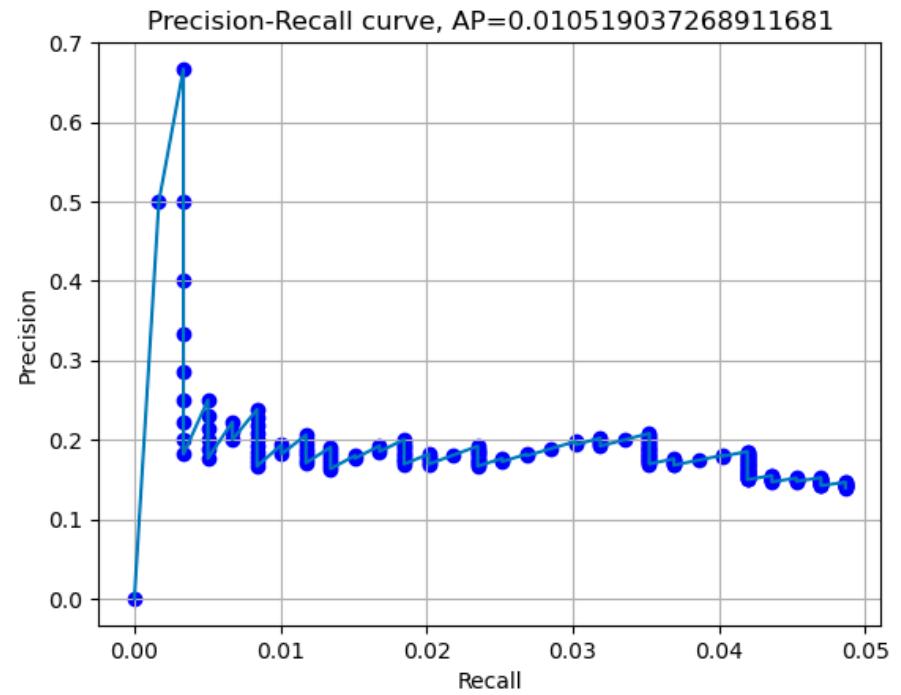
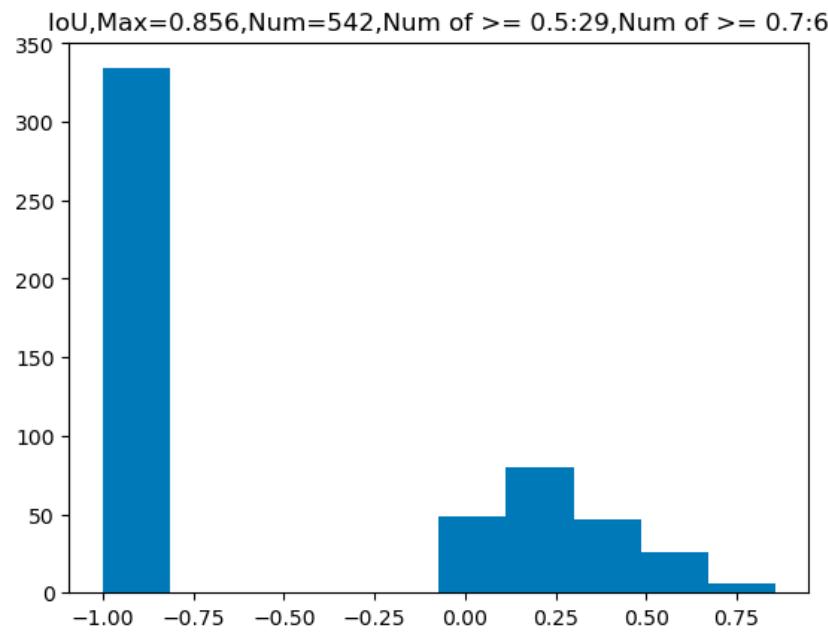


S
O
M
あり



Prompting の工夫によるスコアの改善

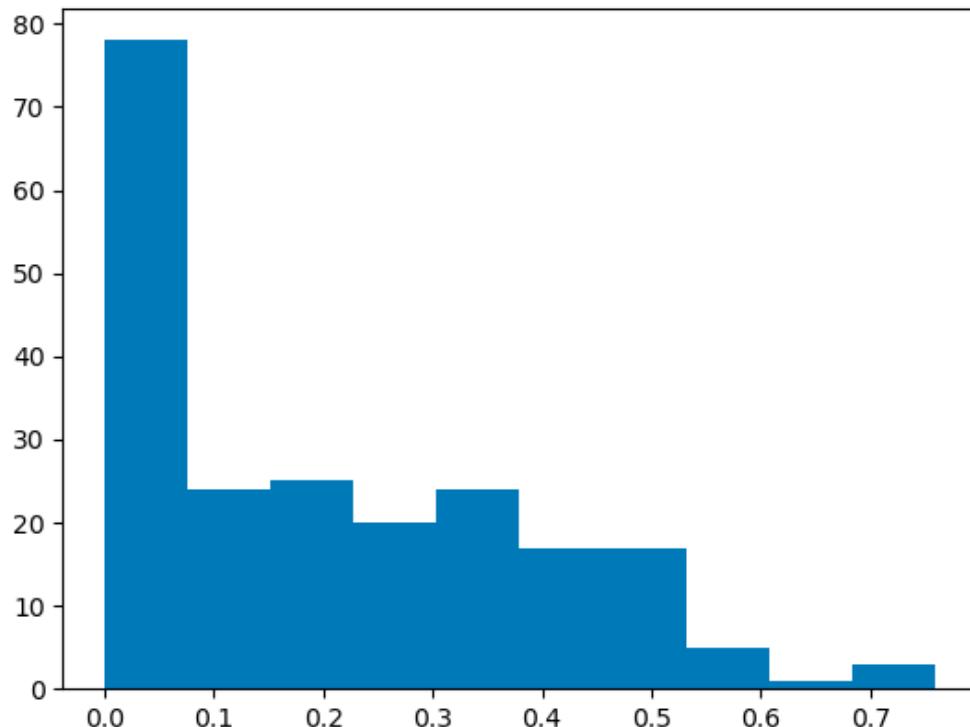
同様の手法でスコアを計測すると以下のようにになった。



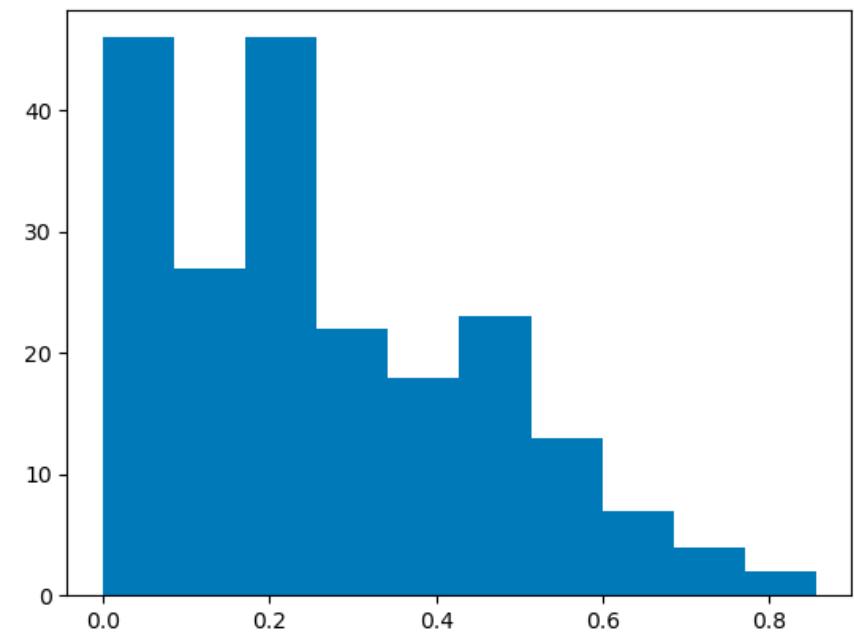
Prompting の工夫によるスコアの改善

同様の手法でスコアを計測すると以下のようになった。

IoU,Max=0.759,Num=214,Ave=0.2020696486355289
Num of >= 0.5:15,Num of >= 0.7:2

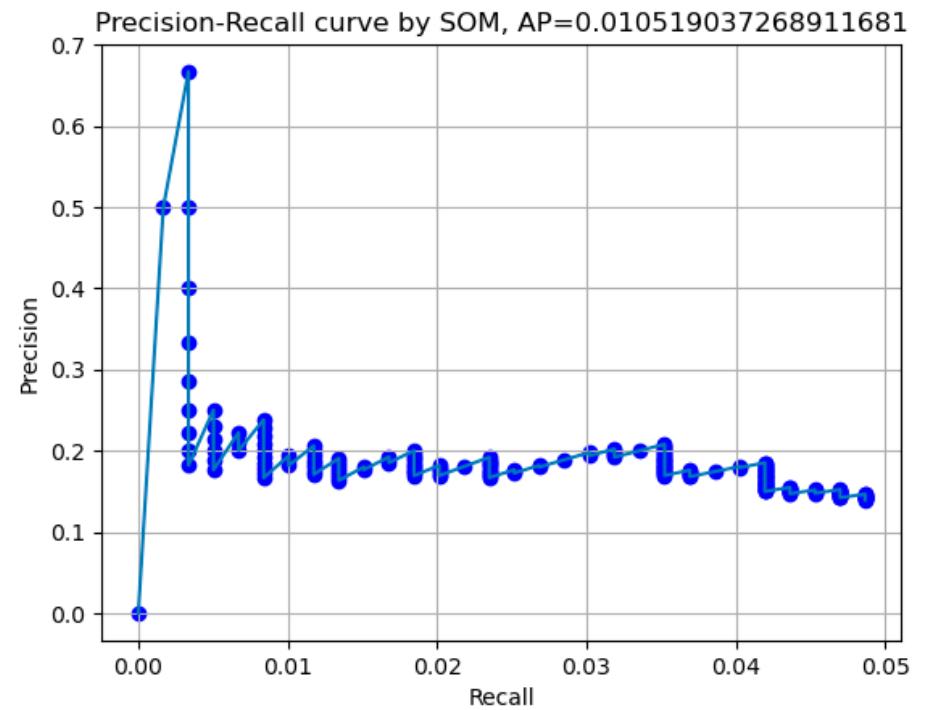
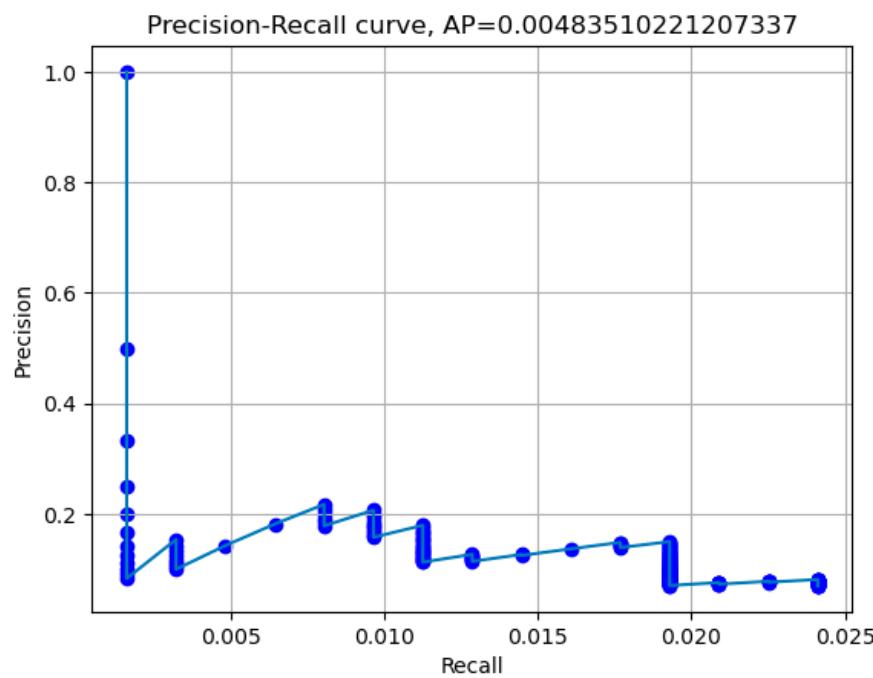


IoU by SOM,Max=0.856,Num=208,Ave=0.2643263864080909
Num of >= 0.5:29,Num of >= 0.7:6



Prompting の工夫によるスコアの改善

同様の手法でスコアを計測すると以下のようにになった。



参考文献

実世界における総合的参照解析を目的としたマルチモーダル対話データセットの構築

<https://github.com/riken-grp/JCRe3/blob/main/README.md>

IoU(Intersection over Union): 物体検出における評価指標・ロス関数

<https://cvml-expertguide.net/terms/dl/object-detection/iou/>

[物体検出] AP(Average Precision)を理解するための諸ステップと計算コード

https://output-zakki.com/apaverage_precision/#toc6

Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V

<https://github.com/microsoft/SoM>

GPT-4oを使った訓練無しでの物体検出(BBox)の精度はセグメンテーションの活用で改善できる

https://zenn.dev/saitom_tech/articles/gpt4o_bbox_improvement_with_segmentation