A. Overview

In this supplementary material, we present more details and results.

- We provide the prompts for ChatGPT to generate questions in **numerical direct** group and **boolean** group.
- We show some qualitative results of end-to-end models.
- We show some qualitative results of ViperGPT on *C*-VQA-Synthetic.
- We present more detailed statistics and the top words for nouns and verbs of *C*-VQA.

B. Prompt for ChatGPT

In the process of annotating *C*-VQA-Real, we prompt ChatGPT to generate most new counterfactual modified questions for **numerical direct group** and **boolean group**. to maximize the correctness of ChatGPT-generated questions, we leverage chain-of-thought [38] strategy and insert in-context-examples [33] into the prompt. We adopt different prompt patterns for the two groups, and the whole prompts are shown below.

B.1. Numerical Direct Group

The counterfactuals for questions in numerical direct group are simple, so we prompt ChatGPT to produce the counterfactual suppositions straight. Then the new answer can be obtained through simple calculations.

```
You will change some numerical questions.
Your task is to perform the following actions:
1 - Read the original numerical question and answer
2 - Increase or decrease the number of items directly.
3 - Work out how this would change the answer to the question.
4 - Write a new question that asks how many items would there be if the number of items was
    increased or decreased according to the step 2.
    Change the original questions to new questions of unreal conditions with counterfactual
        presuppositions, using if clauses. Do not change the meaning of questions in new
        questions.
5 - Write the new answer to the new question.

Answer each initial question with the following format:
Original question:<original question>
Original answer:<original answer>
Step1:Add or remove <number> <item> to the original question.
Step2:<how the answer would change>
New question:<new question>
New answer:<new answer with a single number>

Here are some examples:
-----
Original question:How many birds are there?
Original answer:3
Step1:Add 3 birds
Step2:The answer would be 3+3=6
New question:How many birds would there be if 3 birds came?
New answer:6
-----
Original question:How many people in the picture?
Original answer:2
Step1:add 2 women
Step2:The answer would be 2+2=4
New question:How many people would be in the picture if there were 2 more women?
New answer:4
-----
Original question:How many zebras are here?
Original answer:2
Step1:1 zebra left and 2 zebras came
```

```
Step2:The answer would be 2-1+2=3
New question:How many zebras would there be if 1 zebra left and 2 zebras came?
New answer:1
-----
Original question:How many bikes are outside?
Original answer:2
Step1:double the bikes
Step2:The answer would be 2*2=4
New question:How many bikes would there be if the number of bikes doubled?
New answer:4
-----
Original question:How many sinks?
Original answer:2
Step1:add 2 sinks
Step2:The answer would be 2+2=4
New question:How many sinks if two more sinks were added?
New answer:4
-----
Original question:How many oranges are there?
Original answer:2
Step1:eat all oranges
Step2:The answer would be 2-2=0
New question:How many oranges would there be if all oranges were eaten?
New answer:0
-----
Original question:How many animals are here?
Original answer:2
Step1:another zebra comes
Step2:The answer would be 2+1=3
New question:How many animals would there be if another zebra came?
New answer:3
-----
Original question:How many birds are there?
Original answer:3
Step1:2 birds fly away
Step2:The answer would be 3-2=1
New question:How many birds would there be if 2 birds flew away?
New answer:1
-----

Now change the following questions step by step:
```

B.2. Boolean Group

To ensure the model fully understands the counterfactual suppositions, we propose that the new answers should be different from the original ones. However, applied with the same prompt strategy as numerical direct group, ChatGPT often fails to flip the original answer for questions in boolean group. Therefore we alter the CoT strategy as follows:

- Flip the original answer and describe what the situation is now.
- Design a counterfactual supposition that can make this situation true.

```
You will change some questions.
Your task is to perform the following actions:
1 - Read the original yes/no question and answer
2 - FLip the original Answer
3 - Work out how to make the answer true.
4 - Write a new question that asks if the answer would be true if the action you worked out
    in step 3 was performed.
    Change the original questions to new questions of unreal conditions with counterfactual
```

```
        presuppositions, using if clauses. Do not change the meaning of questions in new
        questions.
5 - Write the new answer to the new question.

Answer each initial question with the following format:
Original question:<original question>
Original answer:<original answer>
Step1:The new answer should be <yes/no>, so ...
Step2:How to make ...:...
New question:<new question>
New answer:<new answer>

Here are some examples:
-----
Original question:Are the goggles covering her eyes?
Original answer:yes
Step1:The new answer should be no, so the goggles are not covering her eyes.
Step2:How to make the goggles not cover her eyes: take off the glasses.
New question:Would the goggles be covering her eyes if she took off the glasses?
New answer:no
-----
Original question:Is there a hotdog on this car?
Original answer:yes
Step1:The new answer should be no, so there is no hotdog on this car.
Step2:How to make there be no hotdog on this car: remove all food.
New question:Would there be a hotdog on this car if all food was removed?
New answer:no
-----
Original question:Are these vegetables cooked?
Original answer:yes
Step1:The new answer should be no, so these vegetables are not cooked.
Step2:How to make these vegetables not be cooked: make them raw.
New question:Would these vegetables be cooked if they were raw?
New answer:no
-----
Original question:Is he happy?
Original answer:no
Step1:The new answer should be yes, so he is happy.
Step2:How to make him happy: make him laugh.
New question:Would he be happy if he was laughing?
New answer:yes
-----
Original question:Is this woman doing something active?
Original answer:no
Step1:The new answer should be yes, so she is doing something active.
Step2:How to make her do something active: make her dance.
New question:Would this woman be doing something active if she was dancing?
New answer:yes
-----
Original question:Is the ground wet?
Original answer:no
Step1:The new answer should be yes, so the ground is wet.
Step2:How to make the ground wet: make it rain.
New question:Would the ground be wet if it was raining?
New answer:yes
-----
Original question:Is the sky clear?
Original answer:yes
```

```
Step1:The new answer should be no, so the sky is not clear.
Step2:How to make the sky not clear: make it cloudy.
New question:Would the sky be clear if it was cloudy?
New answer:no
-----
Original question:Is the plane flying?
Original answer:no
Step1:The new answer should be yes, so the plane is flying.
Step2:How to make the plane fly: make it take off.
New question:Would the plane be flying if it took off?
New answer:yes
-----

Now change the following questions step by step:
```

C. Qualitative Result of End-to-end Models

When counterfactuals are added, most models fail to provide correct answers and examples are provided in Fig. 6. We also notice that there exists some weird data in the result table. For example, LLaVA-13B (Vicuna-13B) [26] gets 31.2% correct for both original and counterfactual questions in numerical direct group. We inspect its result and find out that it is often the case that LLaVA-13B (Vicuna-13B) answers the counterfactual questions correctly but answers the original questions incorrectly. Several instances are shown in Fig. 7.

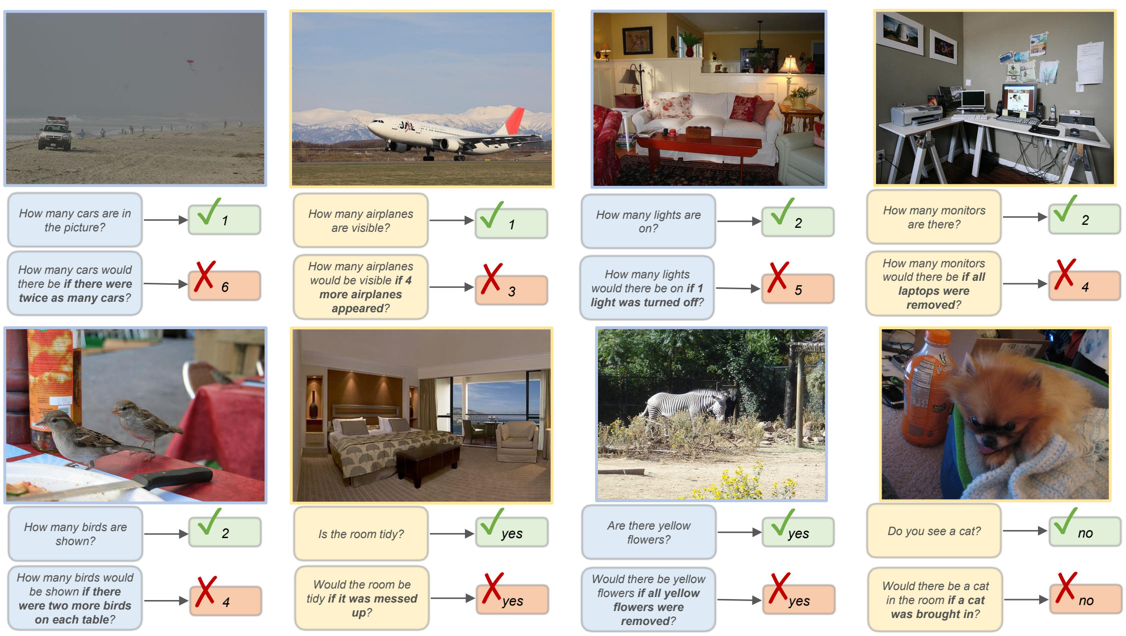


Figure 6. Common failure cases of end-to-end models on *C*-VQA, the added counterfactual presupposition are in **bold**.

D. Qualitative Result of Neuro-symbolic Models

When evaluating ViperGPT [9], We inspect the codes generated by ChatGPT. The codes are sometimes wrong in that ChatGPT misunderstands or even entirely ignores the counterfactuals.

ViperGPT fails to handle *C*-VQA-Synthetic in that it cannot check whether a flower is in a polygon. Despite this, the codes generated for the *C*-VQA-Synthetic is inspiring. We notice that it can often produce correct code even when counterfactual presuppositions are added. Some example codes are provided in Fig. 9.

Figure 7. **Examples of LLaVA-13B.** The model answers the counterfactual questions correctly but fails at the original questions.

Type	Counts
Type-Token Ratio	0.1251
Verb-Token Ratio (total # verb-types)	0.1479
Verb-Token Ratio (total # types)	0.0287
Noun-Token Ratio (total # noun-types)	0.1587
Noun-Token Ratio (total # types)	0.0783
Direct/Indirect/Boolean (%)	37/27/36

Figure 8. **More detailed statistics of *C*-VQA**. The table shows the type-token ratio and the distribution of counterfactual questions. The pictures are word clouds of nouns and verbs in *C*-VQA.

E. Additional Detailed Statistics

The type-token ratios(TTR) of nouns and verbs are provided in Fig. 8. The TTR of *C*-VQA is higher than other common datasets in that the words in presuppositional statements are highly duplicated with that in main clauses. The word clouds in Fig. 8 reveal the top words for nouns and verbs in *C*-VQA.

F. Limitations, future works, and broader impact

Although our work presents the first of its kind evaluation of counterfactual reasoning abilities of multi-modal large language models, the number of images in *C*-VQA is a limiting factor. However, with the 3K counterfactual questions on real images and the additional 3K images on synthetic data, we are able to show a drastic drop in performance when current state-of-the-art multi-modal models are evaluated on the counterfactual questions. One future work we are planning is to gather annotations for more images and cover more models for the evaluation.

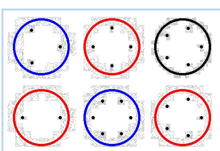
As counterfactual reasoning is considered as the corner stone of human intelligence, we hope our proposed benchmark could help evaluate the progress towards artificial general intelligence or building the next generation of AI assistant.

How many trees would be pictured **if the leftmost tree were cut down**?

```
def execute_command(image):
    image_patch = ImagePatch(image)
    tree_patches = image_patch.find("tree")
    tree_patches.sort(key=lambda x: x.left)
    leftmost_tree = tree_patches[0]
    leftmost_tree_cropped = image_patch.crop(leftmost_tree.right,
        image_patch.lower, image_patch.right, image_patch.upper)
    remaining_tree_patches = leftmost_tree_cropped.find("tree")
    return len(remaining_tree_patches)
```

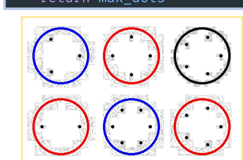
How many players would be in the picture **if all the players wore red**?

```
def execute_command(image):
    image_patch = ImagePatch(image)
    player_patches = image_patch.find("player")
    red_player_patches = [patch for patch in player_patches
                 if patch.verify_property("player", "red")]
    return str(len(red_player_patches))
```



How many dots do a circle contain at most? Select the correct answer:
A:7 B:9 C:4 D:6

```
def execute_command(image):
    image_patch = ImagePatch(image)
    circle_patches = image_patch.find("circle")
    max_dots = 0
    for circle_patch in circle_patches:
        dots = circle_patch.
            simple_query("How many dots does this circle contain?")
        dots = coerce_to_numeric(dots)
        if dots > max_dots:
            max_dots = dots
    return max_dots
```



How many dots would a circle contain at most **if one of the circles with most dots were removed**? Select the correct answer: A:8 B:4 C:10 D:5

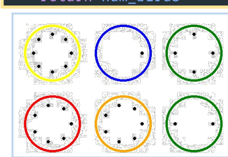
```
def execute_command(image):
    image_patch = ImagePatch(image)
    circle_patches = image_patch.find("circle")
    circle_patches.sort(key=lambda x: len(x.find("dot")),
                        reverse=True)
    if len(circle_patches) > 1:
        circle_patches.pop(0)
    max_dots = max(len(circle.find("dot"))
                   for circle in circle_patches)
    return max_dots
```

How many goats would be here **if each giraffe was replaced by a goat**?

```
def execute_command(image):
    image_patch = ImagePatch(image)
    giraffe_patches = image_patch.find("giraffe")
    goat_patches = image_patch.find("goat")
    num_giraffes = len(giraffe_patches)
    num_goats = len(goat_patches)
    total_goats = num_goats + (num_giraffes * num_goats)
    return str(total_goats)
```

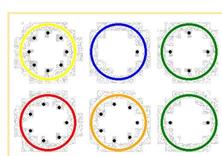
How many birds would there be **if there were loud sounds of guns**?

```
def execute_command(image):
    image_patch = ImagePatch(image)
    bird_patches = image_patch.find("bird")
    num_birds = len(bird_patches)
    if image_patch.llm_query("Are there loud sounds of guns?") == "yes":
        num_birds *= 0
    return num_birds
```



How many dots are there in all the circles together? Select the correct answer: A:29 B:25 C:31 D:34

```
def execute_command(image):
    image_patch = ImagePatch(image)
    circle_patches = image_patch.find("circle")
    total_dots = 0
    for circle_patch in circle_patches:
        dots = circle_patch.simple_query("How many dots?")
        total_dots += int(coerce_to_numeric(dots))
    return total_dots
```



How many dots would there be in all the circles together **if 24 dots were removed from the circles**? Select the correct answer:
A:10 B:8 C:5 D:3

```
def execute_command(image):
    image_patch = ImagePatch(image)
    circle_patches = image_patch.find("circle")
    total_dots = 0
    for circle_patch in circle_patches:
        dots = circle_patch.
            simple_query("How many dots are in this circle?")
        total_dots += int(dots)
    total_dots -= 24
    return total_dots
```

Figure 9. **The codes generated by ViperGPT**. Here the codes for *C*-VQA-Real are logically wrong, and the codes for *C*-VQA-Synthetic are logically correct.

Acknowledgement

Yongshuo Zong was supported by the United Kingdom Research and Innovation (grant EP/S02431X/1), UKRI Centre for Doctoral Training in Biomedical AI at the University of Edinburgh, School of Informatics. For the purpose of open access, the author has applied a creative commons attribution (CC BY) licence to any author accepted manuscript version arising.