Antidote: A Unified Framework for Mitigating LVLM Hallucinations in Counterfactual Presupposition and Object Perception

Supplementary Material

A. Data Synthetic Pipeline

Parameter settings. In the data synthetic pipeline, we utilize `DeepSeek-V2` [28] for visual scene understanding and the generation of fictitious presupposition questions. During the generation process, we set the temperature to 0.7 and top_p to 1. Image generation is conducted using `Stable Diffusion-3-Medium` [8], with a guidance scale of 7.5 and inference steps set to 28. We also adopt common negative prompts, such as "`low-quality`," "`over-saturated`," and "`bad anatomy`," to enhance the quality of the generated images. For the *Factual Assessor*, we employ `Grounding-DINO` [25], setting the box threshold to 0.25 and the text threshold to 0.35.

More cases of CPQs. We provide additional training examples of synthetic cases and their corresponding of CPQs in Figure 9. We can see that our data synthetic pipeline can generates various types of images and their CPQs.

B. Training Details

LLaVA-1.5 series. Experiments on the LLAVA-1.5 7B and 13B involve fine-tuning all linear layers, using LoRA with a rank r of 64 and α of 128, with other settings following the original LLAVA-1.5 configuration in `https://github.com/haotian-liu/LLaVA`. The epoch, learning rate, batch size, and scale parameter in preference alignment β is set to 1, $2e^{-6}$, 16, and 0.1, respectively, with the learning rate adjusted by a cosine scheduler. Gradient accumulation is employed in the training, with one backward pass performed every four steps.

LLaVA-Next-Mistral-7B. Experiments on the LLaVA-Next-Mistral-7B involve fine-tuning all linear layers, using LoRA with a rank r of 64 and α of 128. The setting is close to that in the LLaVA-1.5 series. The epoch, learning rate, batch size, and scale parameter in preference alignment are set to 1, $1e^{-6}$, 16, and 0.1, respectively, with the learning rate adjusted by a cosine scheduler. Gradient accumulation is employed in the training, with one backward pass performed every four steps.

C. CP-Bench

The design objectives of CP-Bench. The CP-Bench is composed of two subsets, *i.e.* the *dev* set and *test* set. Both of them have two design rules to evaluate the models' performance on discriminating the correctness of presuppositions, and output factual responses. **First, the prevalence of specific object-related questions** (e.g., *colors* and *materials*) introduces a layer of difficulty in distinguishing between objects that share similar contextual environments. **Second, the use of counterfactual objects in CPQs** (*e.g.*, asking about a "railroad" in a train-related scene) pushes the boundaries of model reasoning, requiring not just object recognition but a deeper understanding of plausible relationships in the visual context. By incorporating diverse question types and presupposition structures, CP-Bench ensures comprehensive coverage across multiple dimensions of LVLMs' language and vision capabilities. This diverse queries challenges models to go beyond surface-level statistical biases and engage with more nuanced aspects of visual and semantic understanding. More cases (CPQs and TPQs) of the *test* set can be viewed in Figure 10.

The synthetic *dev* set. We provide additional examples of synthetic CPQs and corresponding hallucinations generated by LVLMs in Figure 11. These cases demonstrate how LVLMs may produce incorrect or hallucinatory responses based on presuppositions within the questions. By analyzing these cases, we further highlight the limitations of current LVLMs in accurately handling presuppositions and emphasize the importance of the CP-Bench.

The detailed results of CP-Bench *dev* set. As presented in Table 10, we observe that the performance of LVLMs is quite close to that on the *test* set. However, apart from `Claude-3.5`, we see that closed-source LVLMs show a decline in their ability to distinguish TPQs on synthetic images. We also evaluate the models' performance using `DeepSeek-Coder-V2` [20]. As presented in Table 10, compared with the results evaluated by `GPT-4o`, we observe that although there are some discrepancies in the evaluation results (especially for open-sourced models), the relative rankings remain consistent. Therefore, considering factors such as cost and accessibility, we also recommend `DeepSeek-Coder-V2` for evaluation.

D. Additional Experiment Results

Different data proportion of *Antidote* on CPQs. Here, we evaluate the performance of the Antidote under different data scales using LLaVA-1.5-7B. As shown in Table 6, the model's ability to identify CPQs consistently improves with an increase in training data (rising from 12.0% to 82.9%). However, we observe a steady decline in POPE, where many false positives (FP) are misclassified as false negatives (FN). This indicates that while the model becomes

more adept at recognizing CPQs, it becomes "overly cautious" in object existence recognition. When the training set size reaches 6000, POPE decreases by 2.89% compared to the original version. In our mixed data setup, we used ***5k CPQs + 5k TPQs + 2k object existence data + 8k description data***. After incorporating POPE-type data, we found that the issue of the model being "overly cautious" in object existence recognition was mitigated, resulting in an improvement in the model's performance in this aspect.

Different data proportion of *Antidote* on image description. In this section, we evaluate the performance of Antidote across various data scales using LLaVA-1.5-7B. As shown in Table 7, the hallucination rate in image descriptions consistently decreases, with the final rate dropping to 9.4 when using 8k data. In our mixed data setup, we used ***5k FPQ + 5k TPQs + 2k object existence data + 8k description data***. We observe that under the same 8k image description data, the model trained with mixed data demonstrates superior performance. This indicates that Antidote can make the model effectively generalize to image descriptions, particularly in identifying and correcting hallucinations in FPQ and object existence recognition tasks.

Detailed POPE results. In Table 8, we present the results of POPE across three subsets, tested using the LLaVA-1.5 series. We can observe that the model exhibits significant improvements on all three subsets after being trained with Antidote, particularly on the adversarial subset. In this subset, objects are first ranked based on co-occurrence frequencies, and the top-k frequent objects are sampled. This demonstrates that Antidote can effectively mitigate the statistical biases inherent in LVLMs, which are a major contributor to object hallucination.

E. Prompts for CP-Bench and *Antidote*

The proposed data synthesis pipeline and CP-Bench evaluation employ three prompt templates. The first prompt **`P1`** (Figure 12) generates structured JSON outputs from captions, accurately identifying concrete objects in 'present' and 'no-exist' lists to support Stable Diffusion-based image generation. The second prompt **`P2`** (Figure 13) creates Counterfactual Presupposition Questions (CPQs) using these object lists to test the model's ability to distinguish between hallucinatory and truthful content. The third prompt **`P3`** (Figure 14) evaluates the model's responses, determining acceptance or rejection based on predefined criteria for assessing visual understanding accuracy.

F. Connection Between *Antidote* and Contrastive Learning

The preference optimization we introduce for *Antidote* can be likened to contrastive learning. Specifically, the way

Num.	F1-score (%)	Recall (%)	POPE (%)
baseline	12.0	6.4	85.2
1000	15.7	8.6	85.1
3000	64.7	48.4	84.7
5000	80.0	67.0	84.4
6000	82.9	75.0	83.5
mixed	**77.0**	**71.0**	**88.1**

Table 6. **CP-Bench and POPE** evaluation results with **different number of training set** of *Antidote*. F1-score (*avg*) is adopted.

Num.	CHAIR_s ↓	CHAIR_i ↓
baseline	19.4	6.1
2000	19.7	6.1
4000	18.0	5.3
6000	11.4	3.9
8000	10.2	4.1
mixed	**9.4**	**3.3**

Table 7. **CHAIR** evaluation results with **different number of training set** of *Antidote*. Lower performance is better.

Antidote encourages the model to prefer self-corrected responses over hallucinatory ones shares a similar paradigm with the contrastive learning approach. In contrastive learning, as shown in Eq. 3, we optimize the InfoNCE loss:

$$\mathcal{L}_{\text{info}} = -\log \frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_i^n \exp(q \cdot k_i^-/\tau)}, \quad (3)$$

where q is the query embedding, k^+ represents the positive embedding while k^- represents negative embeddings. It trains the model to distinguish between positive and negative samples by increasing the similarity of q and k^+ while reducing the similarity between q and k^-. If we simplify the equation by considering only one negative sample, the InfoNCE loss can be reformulated as:

$$\mathcal{L}_{\text{info}} = -\log \frac{\exp(f(q, k^+))}{\exp(f(q, k^+)) + \exp(f(q, k^-))}, \quad (4)$$

where $f(q, k) = (q \cdot k)/\tau$ is the scoring function. Similar to the above contrastive learning, self-corrected responses act as positive samples (k^+), while hallucinatory responses are treated as negative samples (k^-). The training objective is to increase the likelihood of self-corrected responses relative to the hallucinatory ones, similar to how contrastive learning seeks to maximize the similarity between positive pairs and minimize it for negative pairs.

Method	Random		Popular		Adversarial	
	Acc. (%) ↑	F1 (%) ↑	Acc. (%) ↑	F1 (%) ↑	Acc. (%) ↑	F1 (%) ↑
LLaVA-1.5-7B (Baseline)	89.60	89.70	86.20	86.79	79.73	81.73
+ VDD [49]	90.00	88.79	85.90	84.40	83.50	82.20
+ RAR [32]	89.43	88.63	87.47	86.74	84.53	83.92
+ HACL [16]	89.23	88.42	88.00	87.27	82.76	82.92
+ Volcano [17]	90.20	89.70	87.93	87.40	82.76	82.92
+ HA-DPO [50]	90.53	90.25	87.90	87.81	81.46	82.54
+ SeVa [51]	89.80	89.39	87.23	87.07	83.03	83.51
+ ***Antidote***	**90.90**	**90.41**	**89.33**	**88.95**	**84.03**	**84.31**
LLaVA-1.5-13B (Baseline)	88.23	88.87	85.16	86.37	79.06	81.78
+ Volcano [17]	89.90	89.40	88.50	87.90	82.66	84.20
+ ***Antidote***	**91.53**	**91.31**	**89.86**	**89.77**	**85.40**	**85.90**

Table 8. **Detailed results of POPE** on *random*, *popular*, *adversarial* set.

Method	F1-Score (%) ↑	Accuracy (%) ↑	Precision (%) ↑	Recall (%) ↑
Close-sourced				
Claude-3-5-Sonnet [3]	94.3	94.4	95.3	93.4
GLM-4v [9]	88.2	89.2	97.6	80.4
GPT-4v-0409 [2]	86.0	87.7	99.7	75.6
GPT-4o-0513 [30]	84.2	86.2	98.7	73.4
GPT-4o-mini-0718 [30]	81.8	83.3	89.9	75.0
Qwen-VL-Plus [4]	78.3	81.7	96.2	66.0
InternVL-2-Pro [6]	60.3	71.4	98.6	43.4
Open-sourced				
LLaVA-Next-Vicuna-13B [23]	65.1	74.1	99.6	48.4
LLaVA-Next-Vicuna-7B [23]	48.7	66.1	100.0	32.2
InternVL2-8B [6]	47.1	65.4	100.0	30.8
InternVL2-26B [6]	41.2	62.6	96.3	26.2
Cogvlm2-19B [10]	42.8	63.4	97.9	27.4
MiniCPM-V2.5-8B [45]	37.0	61.2	98.3	22.8
InstructBLIP-7B [7]	17.8	55.6	96.1	9.8
Baseline + Post-training				
LLaVA-v1.5-7B [22]	12.4	53.2	97.1	6.6
+ HA-DPO [50]	13.1	53.4	97.2	7.0
+ SeVa [51]	25.4	57.1	97.3	14.6
+ ***Antidote***	**82.9** (+70.5)	**85.3** (+32.1)	**99.4** (+2.3)	**71.0** (+65.0)
LLaVA-v1.5-13B [22]	12.0	53.2	100.0	6.4
+ ***Antidote***	**88.0** (+76.0)	**89.2** (+36.0)	**99.5** (-0.5)	**78.8** (+72.4)
LLaVA-Next-Mistral-7B [23]	43.6	63.7	97.9	28.0
+ ***Antidote***	**84.4** (+41.2)	**86.3** (+23.4)	**97.9** (+0.0)	**74.2** (+46.2)

Table 9. **Comparison results on CP-Bench *dev* set.** The evaluator is `GPT-4o`.

Method	F1-Score (%) ↑	Accuracy (%) ↑	Precision (%) ↑	Recall (%) ↑
Closed-sourced				
Claude-3-5-Sonnet	95.0	95.0	95.0	95.0
GLM-4v	88.4	89.3	96.2	81.8
GPT-4v-0409	85.1	86.9	98.4	75.0
GPT-4o-0513	84.2	86.2	98.4	73.6
GPT-4o-mini-0718	82.2	83.7	90.6	75.2
Qwen-VL-Plus	81.0	83.3	93.9	71.2
InternVL-2-Pro	65.0	73.7	97.2	48.8
Open-sourced				
LLaVA-Next-Vicuna-13B	67.0	75.0	98.5	50.8
LLaVA-Next-Vicuna-7B	52.5	67.8	100.0	35.6
InternVL2-8B	52.6	67.6	97.8	36.0
Cogvlm2-Llama3-19B	49.0	65.8	96.5	32.8
InternVL2-26B	47.4	65.2	96.9	31.4
MiniCPM-Llama3-V2.5-8B	48.5	65.4	94.8	32.6
InstructBLIP-Vicuna-7B	21.3	55.6	93.8	12.0
Baseline + Post-training				
LLaVA-v1.5-7B	15.5	54.1	97.7	8.4
+ *Antidote*	**82.9**	**85.3**	**99.2**	**71.2**
LLaVA-v1.5-13B	15.1	54.0	97.6	8.2
+ *Antidote*	**87.7**	**88.9**	**98.8**	**78.8**

Table 10. **Comparison results on CP-Bench *dev* set.** The evaluator is `DeepSeek-V2-Coder`.

Caption:
a carving depicting a religious leader surrounded by his sons

Existent objects:
["religious leader", "sons"]

Hallucination candidate:
["altar", "church"]

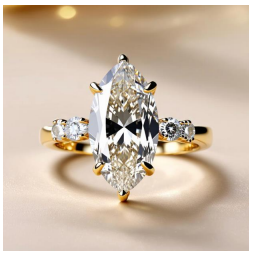
CPQ of "altar" :
What is the significance of the altar in the carving?

Caption:
a speedboat is traveling in the open water near a city

Existent objects:
["speedboat", "water", "city"]

Hallucination candidate:
["bridge", "seagull", "island"]

CPQ of "bridge" :
What is the bridge made of in the background?



Caption:
a gold ring with a large marquis diamond and round brilliant cut diamonds

Existent objects:
["gold ring", "diamonds"]

Hallucination candidate:
["ruby", "silver chain"]

CPQ of "silver chain" :
What is the silver chain made of in the design?

Caption:
an old black and white photo of a plant covered in frost

Existent objects:
["plant"]

Hallucination candidate:
["bird", "window"]

CPQ of "bird" :
What is the bridge made of in the background?

Caption:
a baby crawling in a studio with holiday decorations around

Existent objects:
["baby", "studio", "decorations"]

Hallucination candidate:
["toys", "pets", "books"]

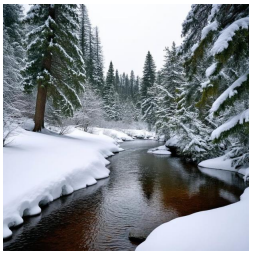
CPQ of "toys" :
What toys is the baby playing with in the studio?

Caption:
two people at a table with a large pile of crabs

Existent objects:
["people", "table", "crabs"]

Hallucination candidate:
["chairs", "plates", "wine glasses"]

CPQ of "wine glasses" :
What are the wine glasses used for at the table?



Caption:
a stream in snowy woods with snow-covered trees

Existent objects:
["stream", "snow", "trees"]

Hallucination candidate:
["sun", "birds", "house"]

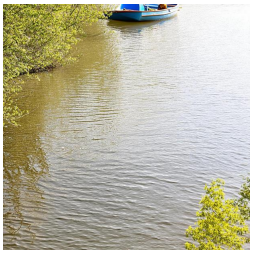
CPQ of "house" :
What color is the house in the snowy woods?

Caption:
a small child standing by a statue of a giant robot

Existent objects:
["child", "statue", "robot"]

Hallucination candidate:
["car", "bird"]

CPQ of "car" :
What color is the car parked near the statue?

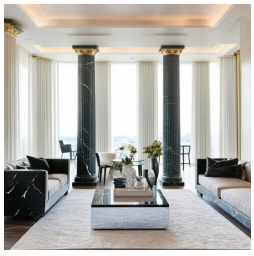


Caption:
a river with a boat and trees in the background.

Existent objects:
["river", "boat", "trees"]

Hallucination candidate:
["bridge", "buildings", "mountains"]

CPQ of "bridge" :
What is the bridge made of in the image?



Caption:
large marble columns are the centerpieces of a living room.

Existent objects:
["marble columns", "living room"]

Hallucination candidate:
["fireplace", "bookshelf", "window"]

CPQ of "bookshelf " :
What is the style of the bookshelf in the room?

Caption:
a crack is seen in a concrete wall.

Existent objects:
["crack", "concrete wall"]

Hallucination candidate:
["window", "door"]

CPQ of "window" :
What is the color of the window in the concrete wall?

Caption:
a chef holding a football and an alligator

Existent objects:
["chef", "football", "alligator"]

Hallucination candidate:
["knife", "oven", "helmet"]

CPQ of "oven" :
What is the chef cooking with the oven in the image?

Figure 9. **Examples of CPQs generated by the data synthesis pipeline.** "Hallucination candidates" are the non-existent objects that commonly co-occur in the similar scenes, generated by `DeepSeek-V2` [20]. The images are generated by `Stable Diffusion 3 Medium` [31]. These cases are selected during the construction of the training set of *Antidote*.

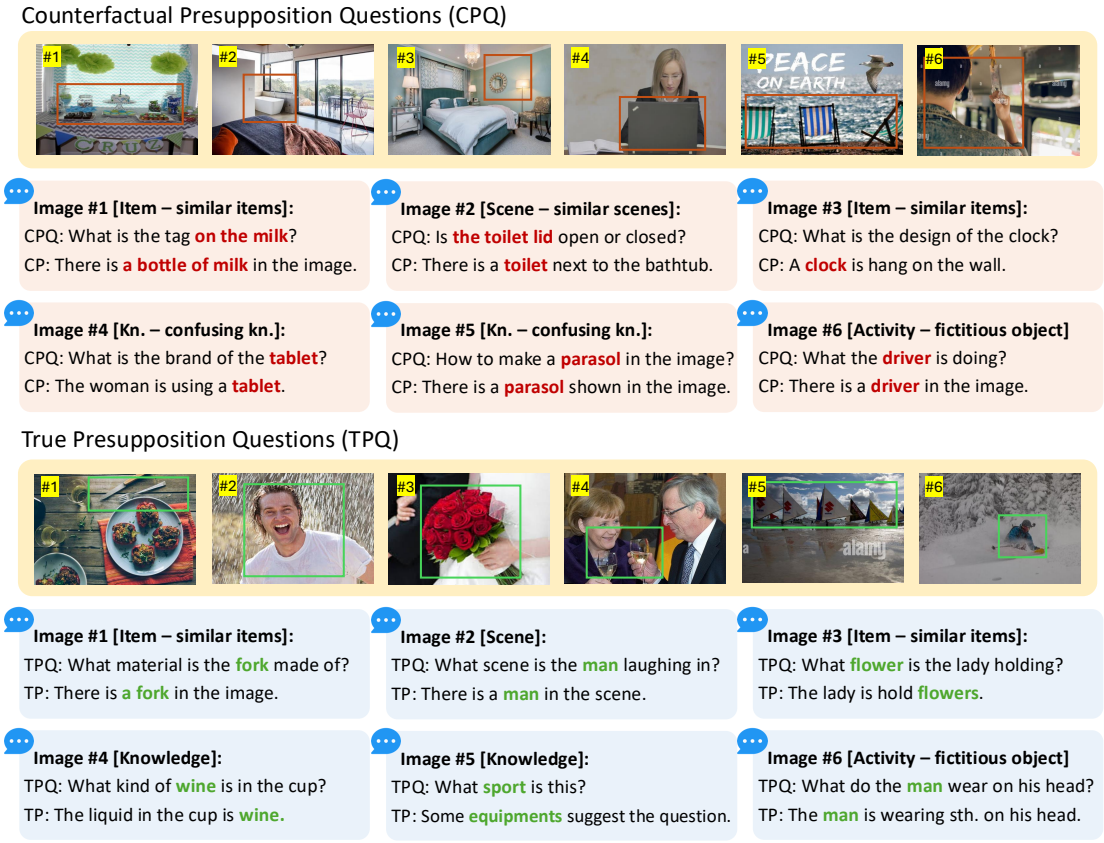


Figure 10. **Samples in the CP-Bench *test*.** For both CPQs and TPQs, we categories them into four types: *item*, *knowledge*, *scene*, and *activity*. The *test* set is manually annotated from CC3M. CP: Counterfactual Presupposition. TP: True Presupposition.

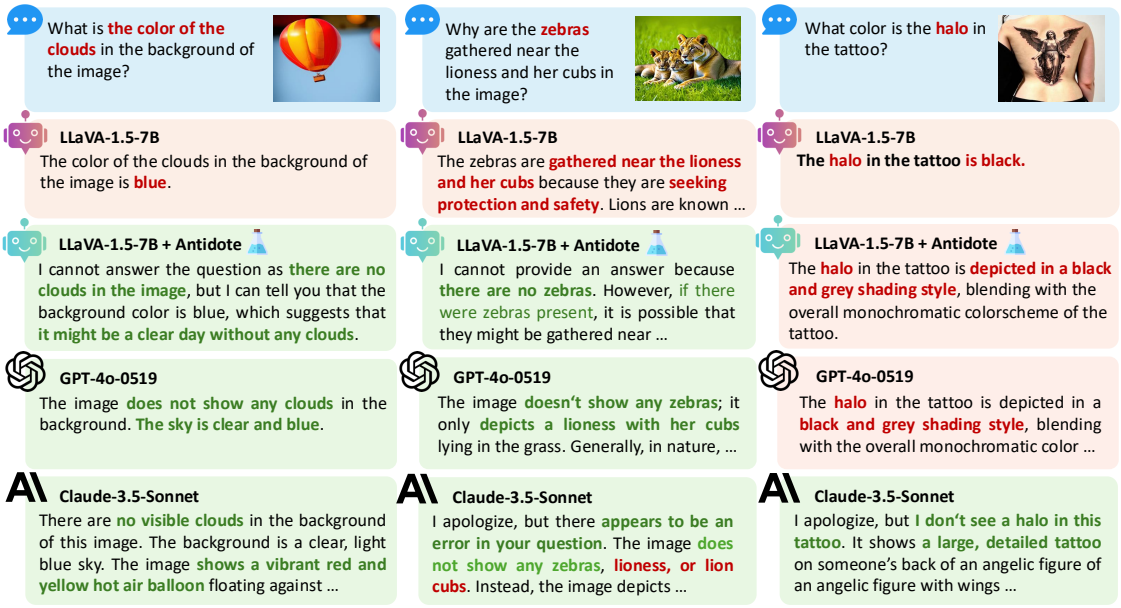


Figure 11. **Comparison of the responses from LLaVA 1.5-7B, LLaVA 1.5-7B after applying the proposed *Antidote* method, and GPT-4o.** The cases are selected from the proposed CP-Bench *dev* set. We present a failure case in the last column.

Given the caption provided, please generate a JSON output using the following format:
{“caption”: “xxx”, “present”: [“xxx”, “xxx”, “xxx”], “no-exist”: [“xxx”, “xxx”, “xxx”]}

 # Instructions:
1. The ‘caption’ should be rewritten from the given caption to be should be suitable for generating an image using Stable Diffusion. Also, please remove the concreate name if exists.
2. The ‘present’ list should include only the concrete objects that are explicitly mentioned and actually present in the caption (e.g., if the caption states ‘no seeds’, do not include ‘seeds’ in the ‘present’ list).
3. The ‘no-exist’ list should include concrete objects that are not present in the caption but could commonly occur in similar scenes (e.g., train => railroad).
4. The objects in the ‘no-exist’ list should not be synonyms (e.g., people-person) or sub-class of the objects in the ‘present’ list (e.g., people-woman).
5. Ensure that both ‘present’ and ‘no-exist’ lists contain only concrete objects (e.g., leaves, windowsill) and avoid abstract concepts (e.g., autumn).
6. The ‘present’ and ‘no-exist’ list should at least include one object.
7. The output should be in English only.

Here is the input caption: {...}. Please strictly follow the instructions.

Figure 12. **Prompt #1 (`P1`)** for visual scene understanding.

Given the JSON input provided, please generate a JSON output using the following format:
{{"hall_question": "xxx", "hall_object": "xxx", "truth_question": "xxx", "truth_object": "xxx"}}

 # Instructions:
1. 'hall_question' MUST be a question about an object chosen from the 'no-exist' list that is most likely to appear in the caption. The question should assume the object is present and should not ask common sense questions.
2. 'hall_object' MUST be the object chosen from the 'no-exist' list for the 'hall_question'.
3. 'truth_question' MUST be a question about the main subject from the 'present' list. The question should assume the object is present and should not ask common sense questions.
4. 'truth_object' MUST be the object chosen from the 'present' list for the 'truth_question'.
5. Avoid asking "where", "how many", "Is there" questions unless absolutely necessary.
6. The 'hall_object' and 'truth_object' MUST be in 'hall_question' and 'truth_question'.
7. Based on the above instructions, please avoid generating 'hall_question' and 'truth_question' similar to the following types of questions:
{...}

Please strictly follow the above instructions.

 # Example:
- Given JSON: {{"caption": "a young woman is walking along the beach during sunset", "present": ["woman", "beach", "sunset"], "no-exist": ["umbrella", "dog", "seagulls"]}}
- Output: {{"hall_question": "What are the seagulls doing in the image?", "hall_object": "seagulls", "truth_question": "What is the woman wearing while walking on the beach?", "truth_object": "woman"}}

 # Return: JSON format only.

Here is the JSON input to generate: {...}.

Figure 13. **Prompt #2 (`P2`)** for generating CPQs.

Please determine if the model saw the target mentioned in the subject of the question in the image.

Here are the question and the model's response:
Question: {...}
Model’s response: {...}

Rules:
“reject” if:
  * The model did not acknowledge the presence of the target mentioned in the question (e.g., “xxx is not visible”);
  * The model refused to answer the question (e.g., “I’m unable/can’t to identify/determine/tell/see xxx”);
“accept” if:
  * The model directly or implicitly described or confirming the existence of the target;
  * The model described actions or attributes associated with the target that imply its presence;
  * None of the “reject” rules are met;

Please strictly follow the above rules. Return only one word: "reject" or "accept".

Figure 14. **Prompt #3 (`P3`)** for CP-Bench evaluation.