PunchBench: Benchmarking MLLMs in Multimodal Punchline Comprehension

Anonymous ACL submission

Abstract

Multimodal punchlines, which involve humor or sarcasm conveyed in image-caption pairs, are a popular way of communication on online multimedia platforms. With the rapid development of multimodal large language models (MLLMs), it is essential to assess their ability to effectively comprehend these punchlines. However, existing benchmarks on punchline comprehension suffer from three major limitations: 1) language shortcuts that allow models to solely rely on text, 2) lack of question diversity, and 3) narrow focus on a specific domain of multimodal content (e.g., cartoon). To address these limitations, we introduce a multimodal **Punch**line comprehension **Bench**mark, named **PunchBench**, which is tailored for accurate and comprehensive evaluation of punchline comprehension. To enhance the evaluation accuracy, we generate synonymous and antonymous captions by modifying original captions, which mitigates the impact of shortcuts in the captions. To provide a comprehensive evaluation, PunchBench incorporates diverse question formats and image-captions from various domains. On this basis, we conduct extensive evaluations and reveal a significant gap between state-of-the-art MLLMs and humans in punchline comprehension. To improve punchline comprehension, we propose Simple-to-Complex Chain-of-Question (SC-CoQ) strategy, enabling the models to incrementally address complicated questions by first mastering simple ones. SC-CoQ effectively enhances the performance of various MLLMs on PunchBench, surpassing in-context learning and chain-of-thought.

1 Introduction

Recent research on Multimodal Large Language Models (MLLMs) (Wang et al., 2024; OpenAI, 2024) has made rapid progress in vision-language tasks such as visual question answering (Antol et al., 2015), dense image captioning (Johnson et al., 2016) and optical character recognition (Islam et al., 2017). Despite the advanced capabilities of modern MLLMs in comprehending factual information from visual content, whether they can effectively grasp punchlines within the multimodal context remains an open question.

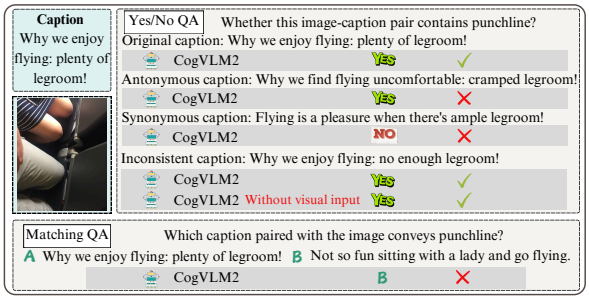


Figure 1: An example of multimodal punchline comprehension. We illustrate the response of CogVLM2 when provided with different captions and question formats.

As illustrated in Figure 1, multimodal punchlines are typically presented as image-caption pairs (Cai et al., 2019), where humor or sarcasm is elicited through a striking contrast or alignment between visual and textual elements. Understanding these punchlines is important yet challenging for the development of MLLMs. **On the one hand**, multimodal punchlines are an essential way of communication on online multimedia platforms. Improving comprehension of punchlines is crucial for many real-world applications, including Human-AI interaction (Hempelmann and Petrenko, 2015) and sentiment analysis (Mahdaouy et al., 2021). **On the other hand**, unlike conventional visual question answering and captioning tasks, multimodal punchline understanding necessitates a nuanced perception of visual content, a strong grasp of language prior knowledge, as well as a deep understanding of the interplay between visual and textual information (Jing et al., 2023).

There are some prior studies on multimodal punchline comprehension, attempting to evalu-

ate sarcasm explanation (Desai et al., 2022) and humor comprehension (Hessel et al., 2023), respectively. However, despite the valuable benchmarks presented by these studies, they suffer from three major limitations that hinder an accurate and comprehensive assessment of multimodal punchline comprehension. **First, existing benchmarks overlook the potential shortcuts in the captions.** As shown in the *Yes/No QA* task from Figure 1, CogVLM2 (Hong et al., 2024) can correctly identify that the original caption conveys a punchline regarding the image but fails when some words in the original caption are replaced with antonymous or synonymous ones. Additionally, the model can correctly answer *Yes/No QA* solely based on an inconsistent caption without visual input. This suggests that the model may exploit biased words (e.g., "enjoy," "plenty of") or text-only inconsistencies (e.g., "enjoy flying" versus "not enough legroom") to arrive at the correct answer rather than genuinely understanding the multimodal punchline. **Second, most previous benchmarks are constrained to a single question format** (Cai et al., 2019; Desai et al., 2022), limiting their ability to assess the robustness of MLLMs across various user question formats. As depicted in Figure 1, the model can answer the *Yes/No QA* correctly but struggle with the *Matching QA*, highlighting performance variations across question formats. **Third, prior works** (Qiao et al., 2023; Hessel et al., 2023) **solely focus on humor or sarcasm within a narrow domain** (*e.g.,* cartoon). This limits their applicability to broader real-world scenarios that convey punchlines, and hence causes insufficient evaluations.

In light of the above limitations, we introduce a novel multimodal **Punch**line comprehension **Bench**mark, **PunchBench** for short, designed to provide an accurate and comprehensive evaluation of this task. To enhance **evaluation accuracy**, we modify captions to mitigate the impact of potential shortcuts. Specifically, we apply context consistency adaptation to eliminate inconsistent captions, and then use word substitution and inversion to generate synonymous and antonymous captions with the help of ChatGPT (OpenAI, 2022). Regarding **evaluation comprehensiveness**, PunchBench features diversity across multiple dimensions. For punchline types, it includes both humor and sarcasm. For task types, it involves two levels of punchline understanding: shallow-level punchline perception and deep-level punchline reasoning. Each task employs diverse question formats: *Yes/No QA*, *Matching QA*, *Multi-option QA* and *Generation QA*. Furthermore, PunchBench spans a wide range of multimodal content domains, including posts, cartoons, comments, and memes. In total, PunchBench comprises $6,000$ image-caption pairs and $54,000$ question-answer pairs, allowing a comprehensive evaluation.

Leveraging PunchBench, we evaluate a range of state-of-the-art MLLMs. The results reveal a significant gap between MLLMs and humans in punchline comprehension. Additionally, the performance of MLLMs varies across different question formats, and shows notable degradation when faced with synonymous or antonymous captions. These observations emphasize the importance of incorporating diverse question formats, synonymous and antonymous captions in the evaluation process.

To improve the punchline understanding ability of MLLMs, we propose a strategy called **Simple-to-Complex Chain-of-Question** (SC-CoQ), inspired by the simple-to-complex progression for solving complicated problems. SC-CoQ structures questions from simple to complex within and across tasks, enabling the models to incrementally develop the capability to address complex questions by first mastering simple ones. Compared to in-context learning (Brown et al., 2020) and chain-of-thought (Wei et al., 2022) methods, SC-CoQ demonstrates superior performance, further validating its effectiveness in promoting punchline comprehension.

In a nutshell, our contributions can be summarized as follows.

- We introduce PunchBench, which, to the best of our knowledge, is the first benchmark for accurate and comprehensive evaluation of multimodal punchline comprehension.
- Extensive evaluations on PunchBench reveal a significant gap between MLLMs and humans in punchline comprehension, and highlights the performance variations across question formats in each task.
- We propose Simple-to-Complex Chain-of-Question (SC-CoQ), which follows a progression from simple to complex questions to effectively improve punchline comprehension.

2 Related Works

2.1 Multimodal Large Language Models

Large Language Models (LLMs) for pure text like ChatGPT (OpenAI, 2022), GPT-4 (OpenAI et al., 2024), and LLaMA (Touvron et al., 2023) have proved impressive comprehension capabilities of text. Following this success and to expand it on multimodal tasks, many efforts (Li et al., 2023; Liu et al., 2023a) have been made to integrate visual comprehension capability into LLMs, and lead to a blowout of Multimodal Large Language Models (MLLMs), both closed-source models (*e.g.,* GPT-4V (OpenAI, 2023a) and GPT-4o (OpenAI, 2024)) and open-source models (*e.g.,* LLaVA series (Liu et al., 2023a, 2024a,b), CogVLM series (Wang et al., 2023; Hong et al., 2024), Qwen-VL family (Bai et al., 2023; Wang et al., 2024) and GLM-4V (GLM et al., 2024)). They demonstrate unprecedented and surprising multimodal understanding capabilities in vision-language tasks such as visual question answering (Antol et al., 2015), dense image captioning (Johnson et al., 2016) and optical character recognition (Islam et al., 2017).

2.2 Punchline Comprehension

Despite significant progress of MLLMs in understanding factual information from visual content (Long et al., 2023; Jian et al., 2024), the punchline comprehension capabilities (Cai et al., 2019; Ouyang et al., 2024) of MLLMs still lack sufficient evaluations. Prior works (Desai et al., 2022; Kumar et al., 2022; Hessel et al., 2023) related to multimodal punchline comprehension have concentrated on sarcasm or humor. For example, Desai et al. curated the MORE dataset for multimodal sarcasm explanation, which aims to explain the ironic semantics of multimodal post. Furthermore, previous benchmarks overlooked potential shortcuts in captions that MLLMs may exploit to answer questions, undermining true comprehension of punchlines. Noticing these concerns, our benchmark is introduced to provide an accurate and comprehensive evaluation of multimodal punchline comprehension.

3 PunchBench

As illustrated in Figure 2, our PunchBench is constructed in four steps: Source Data Collection & Annotation (§ 3.1), Synonymous & Antonymous Caption Generation (§ 3.2), Instruction Construction (§ 3.3), Quality Checking (§ 3.4). In this section, we elaborate on the construction process as well as the data statistics (§ 3.5).

3.1 Source Data Collection & Annotation

The image-caption pairs in our dataset are obtained from two sources. 1) Prior datasets. Recognizing the wealth of resources in prior datasets that contribute to punchline comprehension, we select three relevant datasets, *i.e.,* MTSD (Castro et al., 2019), MORE (Kumar et al., 2022) and HUB (Hessel et al., 2023). Then, we meticulously filter the high-quality image-caption pairs using a hybrid approach that combines both manual and automatic filtering, as detailed in Appendix A.1. 2) Multimedia platforms. To ensure up-to-date of our dataset, we gather image-caption pairs from the social media platforms, such as X, Instagram, and YouTube. Additionally, we include image-caption pairs from the cartoon websites like CartoonMovement and CartoonStock. The information about these multimeida platforms is provided in Appendix F.

After obtaining the raw set of image-caption pairs, we implement a crowd voting process, which is outlined in Appendix A.1, to identify a label indicating whether the image-caption pair contains punchline. Ultimately, we compile a collection of $6,000$ image-caption pairs spanning diverse scenarios (*e.g.,* cartoon, post, comment and meme), half of which are identified as containing punchline. To explain why the particular pair contains punchline, we employ three human annotators to hand craft reasoning sentence for it, which is detailed in Appendix A.1. Finally, we acquire $6,000$ image-caption pairs, corresponding labels and reasoning sentences. To emphasize the superiority of PunchBench, we provide a comparison between our PunchBench and prior datasets in Table 4.

3.2 Synonymous & Antonymous Caption Generation

As aforementioned, MLLMs may exploit shortcuts in the captions, such as word bias and context inconsistency, to answer the question without truly understanding the image-caption pair. To prevent these shortcuts, we generate **synonymous caption** and **antonymous caption** for each image-caption pair through following methods. 1) *Word substitution and inversion.* Assisted by `gpt-3.5-turbo-0125`, we substitute the sentiment, action, object and other words with synonymous words to generate synonymous caption,

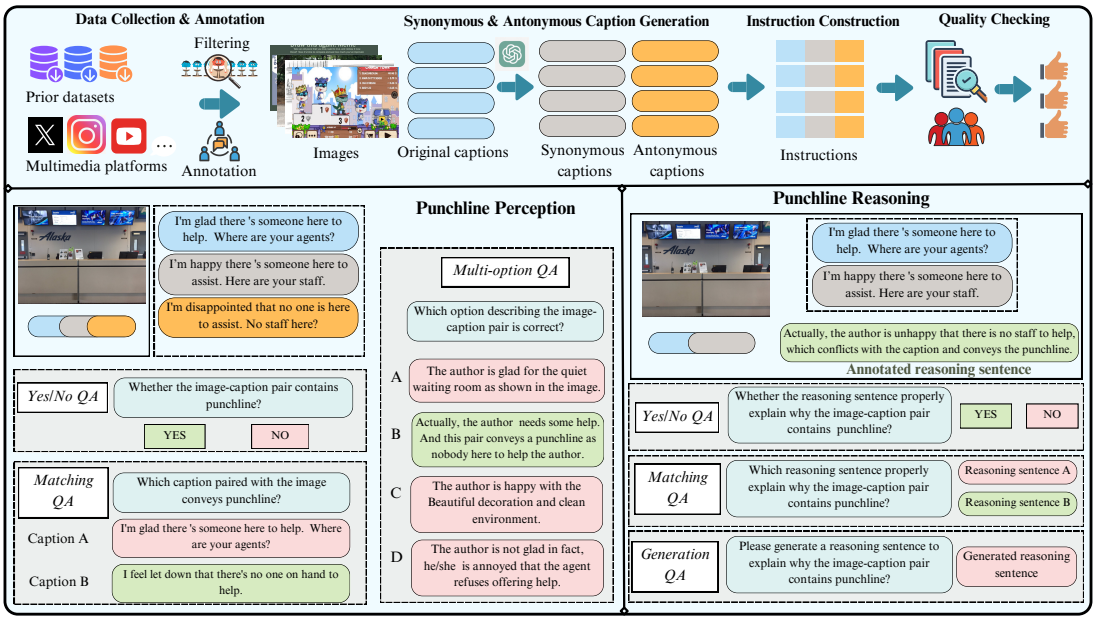


Figure 2: **Upper**: Data collection workflow for PunchBench. We first collect image-caption pairs from prior datasets and multimedia platforms with meticulous filtering, conduct human annotation to obtain the corresponding labels and reasoning sentences for the pairs. And we then utilize `gpt-3.5-turbo-0125` to generate synonymous and antonymous captions corresponding to the original captions. Based on these image-caption pairs, we construct corresponding instructions for punchline perception and reasoning. Finally, we perform quality checking to ensure the reliability of our PunchBench. **Lower**: Data examples for Punchline Perception and Punchline Reasoning.

and we invert the semantics by replacing these words with their antonyms to obtain antonymous caption. 2) *Context consistency adaption*. To adapt the consistency of captions containing semantically conflicting components, *e.g.,* "I am so glad today! What a disgusting rainy day!", we first leverage `gpt-3.5-turbo-0125` to identify and isolate the two conflicting parts, "I am so glad today" contradicts "What a disgusting rainy day". And we then employ word substitution and inversion for the two parts to generate synonymous and antonymous caption. We supplement additional implementation details in Appendix A.2.

3.3 Instruction Construction

Based on the collected image-caption pairs and corresponding annotations, we now construct instructions for two types of tasks: **Punchline Perception**, which assesses whether an MLLM can identify the existence of punchline in image-caption pairs, and **Punchline Reasoning**, which requires the model to understand the reason why a particular image-caption pair contains punchline. Figure 2 illustrates some examples of the instructions. Before delving into the details, we first clarify some notations.

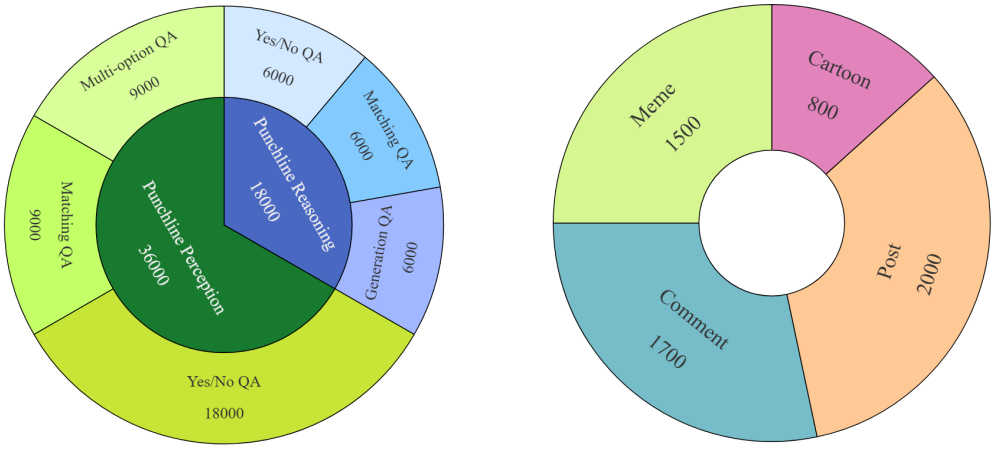
Notations. Each image-caption pair $P_i^x = <I_i, C_i^x>$ consists of an image I_i and a caption C_i^x, where $x \in \{o, s, a\}$ denotes the original (C^o), synonymous (C^s) and antonymous (C^a) caption. And each pair is assigned a label $L_i^x \in \{0, 1\}$, where 1 indicates that the pair contains punchline while 0 is opposite. Notably, P_i^s shares the same label as P_i^o, while P_i^a serves as the contrast. We detail instruction construction process as follows, temporally omitting the subscript i that indexes the samples for simplicity.

3.3.1 Punchline Perception

Yes/No QA. The model is required to answer whether the given image-caption pair P^x contains punchline. The instruction is derived based on various instruction templates, with the answer "Yes" or "No" being determined by the label L^x. To attain a balance, the number of negative answers is equal to that of positive answers.

Matching QA. The model is asked to select between two captions, recognizing which one effectively conveys punchline with the given image. For pair P^x containing punchline, we utilize `gpt-4o-2024-05-13`[1] to generate a distractor caption C^d for the image I. The distractor caption C^d just describes the content of image I without con-

[1] `https://platform.openai.com/docs/models`.



(a) Distribution of Question Formats

(b) Distribution of Image-Caption Pairs

Figure 3: The overall data statistics of our PunchBench.

veying the punchline. Finally, the image-caption pair P^x, as well as C_d are subsequently integrated into several templates to obtain the instructions. To prevent bias associated with the position of captions, we randomize the order in which the two captions are displayed for each instruction.

Multi-option QA. The model aims to discern the correct one from four options *i.e.,* O_1, O_2, O_3, O_4 describing the image-caption pair P^x. The four options are generated by `gpt-3.5-turbo-0125` based on the caption C^x and former distractor caption C^d, with only one being correct. These options, along with P^x are incorporated into the instruction templates. The sequence of the four options are shuffled to avoid the positional bias.

3.3.2 Punchline Reasoning

We utilize the $3,000$ pairs P^o containing punchline, their synonymous captions C^s and annotated reasoning sentences R^a to construct instructions for punchline reasoning.

Yes/No QA. Presented with an image-caption pair and a reasoning sentence, the model is asked to identify whether the reasoning sentence succeeds in explaining why the pair contains punchline. Specifically, we first resort to `gpt-3.5-turbo-0125` to generate distractor reasoning sentence R^d based on our annotated reasoning sentence R^a. And we then randomly assign half of the image-caption pairs to annotated reasoning sentences R^o, while the other part is linked to the distractor ones R^d, incorporate them into instruction templates. The answer to instruction using R^a is "Yes" and using R^d is "No". Finally, we have an equal number of positive and negative instructions.

Matching QA. Given an image-caption pair and two reasoning sentences, *i.e.,* R^a (correct) and R^d (distractor), only one of which appropriately interprets the punchline in the pair, the model is required to select the correct reasoning sentence. Specifically, R^a and R^d are paired with P^o or P^s in several templates to construct the instructions, with the order of R^d and R^a being randomly shuffled.

Generation QA. In this task, the image-caption pair is utilized in various instruction templates to prompt the model to generate a reasoning sentence to explain the punchline, with R^a serving as the reference answer.

The above instructions undergo a thorough review and refinement process by human annotators. The instruction templates and more details of this construction process are supplied in Appendix A.3.

3.4 Quality Checking

To ensure the quality of PunchBench, we randomly sample 100 instructions for each question format, excluding *Generation QA*, for quality checking process. Three human annotators are employed to answer the questions guided by the sampled instructions. Human annotators have an extra option "CBA" that means "Cannot Be Answered" for each question. Among 500 instructions, only 1 is labeled by "CBA", which verifies the high quality of the instructions. Moreover, they answer the questions with high accuracy as results reported in Table 1, which further demonstrates the superior quality of

our dataset.

3.5 Dataset Statistics

We illustrate Figure 3 to exhibit the dataset statistics of our PunchBench. PunchBench consists of $6,000$ image-caption pairs, spanning cartoon, post, comment and meme. Each image has three types of captions: original, synonymous, and antonymous captions. Our question formats include *Yes/No QA*, *Matching QA*, and *Multi-option QA* for punchline perception, and *Yes/No QA*, *Matching QA*, and *Generation QA* for punchline reasoning. Above all, our PunchBench covers a diverse question formats and domains, which can provide a comprehensive evaluation. We also compare our PunchBench with previous Benchmarks in Appendix A.4.

4 Simple-to-Complex Chain-of-Question

In our initial evaluation (the "Zero-shot" results in Table 1), we observe that different question formats present varying levels of difficulty for the MLLMs. The general trend for punchline perception is *Yes/No QA* < *Matching QA* < *Multi-option QA*, and for punchline reasoning, it is *Yes/No QA* < *Matching QA* < *Generation QA*, where < indicates easier than. Inspired by these observations, we propose a Simple-to-Complex Chain-of-Question (SC-CoQ) strategy, which prompts MLLMs to answer the simpler questions before solving the most complex questions. Specifically, we introduce two variations of SC-CoQ, Intra-task and Inter-task:

Intra-task SC-CoQ integrates the various formats of questions within the same task to improve performance on the most challenging question (*i.e., Multi-option QA and Generation QA*). We sequence the questions in a specific order mirroring simple to complex, *i.e.,* <*Yes/No QA*, *Matching QA*, *Multi-option QA* or *Generation QA*>.

Inter-task SC-CoQ incorporates similar question formats (*i.e., Yes/No QA* and *Matching QA*) across different tasks to enhance punchline comprehension. For *Yes/No QA*, we sequentially link the questions from the two tasks, *i.e.,* <*Yes/No* QA_m, *Yes/No* QA_n> or <*Yes/No* QA_n, *Yes/No* QA_m>, where m refers to punchline perception task and n denotes punchline reasoning task. For *Matching QA*, this chain utilizes both *Yes/No QA* and *Matching QA* to reinforce punchline comprehension across tasks, *i.e.,* <*Yes/No* QA_m, *Yes/No* QA_n, *Matching* QA_m, *Matching* QA_n> or <*Yes/No* QA_n, *Yes/No* QA_m, *Matching* QA_n, *Matching* QA_m>. More details of SC-CoQ and specific prompting examples can be found in Appendix B.

5 Experiments

5.1 Baselines

We include both MLLMs and human baseline for evaluation as follows.

Evaluated MLLMs. We evaluate four open-source MLLMs (*i.e.,* LLaVA (Liu et al., 2024b), GLM-4V (GLM et al., 2024), Qwen2-VL (Wang et al., 2024), and CogVLM2 (Hong et al., 2024)) and two closed-source MLLMs (*i.e.,* GPT-4V (OpenAI, 2023a) and GPT-4o (OpenAI, 2024)). And we adopt zero-shot, 3-shot (in-context learning) and Chain-of-Thought (CoT) as the baselines for prompting MLLMs. A detailed description of these models, their parameter settings, and introduction for in-context learning (Brown et al., 2020) and CoT (Wei et al., 2022) are provided in Appendix C.

Human Baseline. To make a comparison with human performance on punchline comprehension, we introduce a human baseline. Specifically, 1) for punchline perception, we first randomly select 100 instructions for each question format except *Generation QA*, and we then recruit human annotators (three undergraduates outside of the work) to answer the questions guided by the instructions. Notably, the manually annotated reasoning sentences serve as the performance of human baseline for the *Generation QA*.

5.2 Evaluation Metric

For *Yes/No QA*, *Matching QA* and *Multi-option QA*, we utilize accuracy as the metric. A response is deemed correct when the candidate option (*e.g., Yes/No*, *Option A/Option B*, or *A/B/C/D*) mentioned in the response matches the ground truth option. The accuracy is then calculated as the ratio of correct responses to the total number of questions. For *Generation QA*, where the responses from MLLMs are free-form, we resort to `gpt-3.5-turbo-0125`[2] to assess whether the response matches the semantics of the annotated reasoning sentence with a binary judgment "Yes" or "No". Responses marked by "Yes" are considered correct and their ratio serves as the accuracy metric. To ensure the reliability of automatic evaluation, we analyze the correlation between automatic and human assessments. The details provided in the Appendix D.3

[2] https://chatgpt.com/.

Model	#Params	Yes/No QA				Matching QA				Multi-choice QA			
		Zero-shot	CoT	3 shot	SC-CoQ	Zero-shot	CoT	3 shot	SC-CoQ	Zero-shot	CoT	3 shot	SC-CoQ
LLaVA	7B	62.7	61.5	63.5	64.8⋆	54.2	54.9	55.8	57.1⋆	36.4	37.5	37.2	39.1⋆
GLM-4V	9B	61.4	61.8	62.2	63.7⋆	55.3	53.1	56.9	57.7⋆	38.2	38.8	39.5	40.6⋆
Qwen2-VL	7B	70.1	71.9	72.4	73.2⋆	58.0	58.4	59.2	61.3⋆	41.7	43.0	42.4	44.1⋆
CogVLM2	19B	68.2	67.6	69.5	71.3⋆	57.3	58.9	58.6	60.8⋆	43.4	44.2	44.7	46.3⋆
GPT-4V	-	75.0	74.2	76.2	78.1⋆	62.1	63.2	63.9	65.0⋆	48.1	50.5	50.3	51.9⋆
GPT-4o	-	77.5	**78.6**	**79.2**	**80.7**⋆	64.2	**66.3**	**65.4**	**67.9**⋆	50.8	**51.4**	**52.0**	**53.1**⋆
Human	-	**98.3**	-	-	-	**97.7**	-	-	-	**90.7**	-	-	-

(a) Punchline Perception

Model	#Params	Yes/No QA				Matching QA				Generation QA			
		Zero-shot	CoT	3 shot	SC-CoQ	Zero-shot	CoT	3 shot	SC-CoQ	Zero-shot	CoT	3 shot	SC-CoQ
LLaVA	7B	60.1	61.7	61.3	62.6⋆	50.7	51.3	51.9	53.0⋆	35.2	37.1	36.6	38.7⋆
GLM-4V	9B	59.7	60.8	61.3	62.9⋆	53.1	52.2	54.8	55.9⋆	37.1	38.5	38.2	39.8⋆
Qwen2-VL	7B	64.5	65.3	66.0	67.4⋆	55.7	56.1	57.2	58.4⋆	40.6	41.5	41.9	43.7⋆
CogVLM2	19B	66.3	67.2	68.0	69.6⋆	54.2	54.9	55.4	56.3⋆	41.8	42.7	42.5	43.4⋆
GPT-4V	-	73.9	74.7	75.4	76.5⋆	57.1	59.0	58.2	60.6⋆	44.7	46.4	45.9	47.5⋆
GPT-4o	-	75.1	**75.9**	**76.2**	**77.4**⋆	59.2	**61.5**	**61.2**	**62.8**⋆	47.2	**47.6**	**48.7**	**50.1**⋆
Human	-	**96.0**	-		-	**93.0**	-		-	**100.0**	-		-

(b) Punchline Reasoning

Table 1: Evaluation results on PunchBench. The best results among the MLLMs are in **boldface**, while the second best are underlined. ⋆ denotes the best results among the prompting methods. The results are the average of four replicates. And the P-value between SC-CoQ performance and other prompting method results is consistently less than 0.01.

demonstrate that the automatic metrics align well with human judgments.

5.3 Main Results

The evaluation results of punchline perception and reasoning are presented in Table 1, and we conclude the following findings from five aspects.

Overall Performance. The evaluated MLLMs exhibit limited capability of punchline comprehension, with the accuracy across different question formats for both punchline perception and reasoning falling below 80% in zero-shot setting. As can be seen, the closed-source models consistently surpass the open-source models, where GPT-4o achieves the leading performance among the evaluated MLLMs. Regrettably, GPT-4o still lags substantially behind human-level performance, revealing a substantial gap in punchline comprehension between MLLMs and humans.

Cross-task Performance. Comparing performance of MLLMs cross the two tasks, we can see that punchline reasoning poses greater challenges than punchline perception, since MLLMs perform worse in punchline reasoning. This disparity is expected, as punchline reasoning demands a deeper understanding to explain why a particular pair contains punchline, rather than simply identifying its presence. Consequently, punchline reasoning proves to be a more complex task for MLLMs compared to punchline perception.

Cross-question Performance. Comparing the results cross question formats within each task, we can observe that there exists a significant variation in performance. The reasons can be two folds. On the one hand, the complexity of the question formats varies inherently. From simplest to most complex, the question formats can be ranked as follows: *Yes/No QA*, *Matching QA*, *Multi-option QA*/*Generation QA*. MLLMs show a noticeable decline in performance as the complexity of the questions increases. On the other hand, individual models have varying innate strengths and weaknesses across different question formats. For instance, LLaVA exceeds GLM-4V in *Yes/No QA* but falls behind GLM-4V in *Matching QA* for punchline perception task.

Effectiveness of SC-CoQ. Compared to the zero-shot setting, both 3-shot and SC-CoQ methods consistently improve performance across all question formats. While CoT method slightly degrades performance in *Yes/No QA* for punchline perception, it enhances performance in other question formats. Notably, SC-CoQ outperforms both 3-shot and CoT approaches corss various question formats, highlighting its superiority. The effectiveness of SC-CoQ is further validated in Section 5.4, where its performance improvements in synonymous and antonymous caption settings are analyzed.

5.4 Effect of Synonymous and Antonymous Captions

To explore the effect of synonymous and antonymous captions, we compare the performance of CogVLM2 cross the original, synonymous and

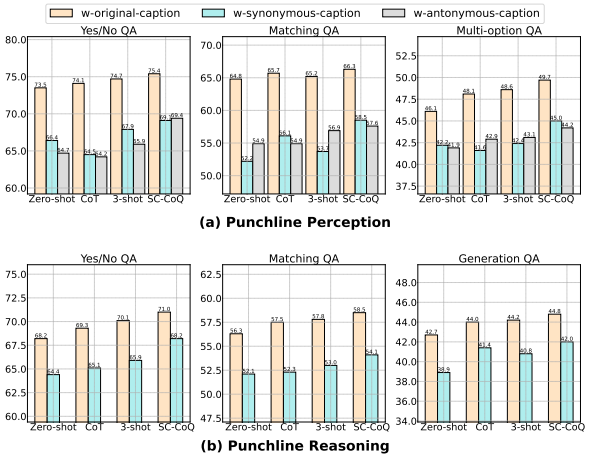


Figure 4: Performance comparison cross original, synonymous and antonymous captions in zero-shot, 3-shot, CoT and our SC-CoQ.

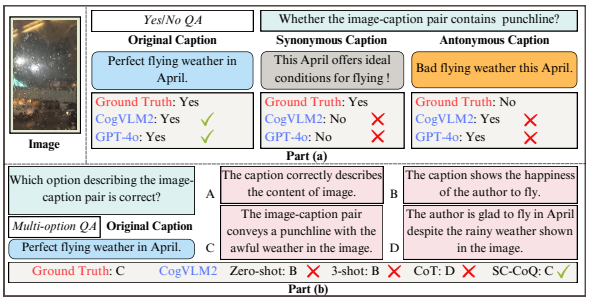


Figure 5: Example responses from CogVLM2 and GPT-4o to the *Yes/No QA* with zero-shot prompts. Responses from CogVLM2 to *Multi-option QA* with different prompting methods are also presented.

antonymous captions, as illustrated in Figure 4. And the performance comparison for other models are provided in Appendix D.3. We analyze the results from two perspectives: 1) There is a notable drop in model performance across different question formats when replacing the original caption with synonymous or antonymous captions. It suggests that synonymous and antonymous captions effectively successfully eliminate shortcuts found in the original captions and hence challenge models to achieve a thorough comprehension of image-caption pair, which leads to a more comprehensive assessment for punchline comprehension capabilities. 2) When using 3-shot and CoT methods, model performance with synonymous and antonymous captions lags behind that with the original captions. However, the models show significant improvement across original, synonymous and antonymous captions when applying SC-CoQ. It proves that SC-CoQ can enhance the models' ability to effectively capture the semantics of image-caption pairs and hence achieve better punchline comprehension.

5.5 Qualitative Analysis

To provide an intuitive display, we illustrate some testing samples in Figure 5 for qualitative analysis. Part (a) showcases the responses from two representative models CogVLM2 and GPT-4o in the *Yes/No QA*. Both of them answer correctly when given the original caption, but fail when the original caption is replaced by the synonymous or antonymous caption. This indicates the biases existing in the captions and hence the models may not truly understand the inherent semantics of the image-caption pair to attain the answer. And it underscores the significance of introducing synonymous and antonymous captions in assessing punchline comprehension. Part (b) exhibits the responses of CogVLM2 with zero-shot, 3-shot, CoT and SC-CoQ for *Multi-option QA*. Notably, with the guidance of SC-CoQ, CogVLM2 successfully answers the question, whereas it fails under the other settings (*i.e.,* zero-shot, 3-shot, and CoT). It highlights the effectiveness of SC-CoQ in enhancing punchline comprehension. More qualitative results for other question formats can be found in Appendix D.4.

6 Conclusions

We introduce PunchBench, a benchmark designed to evaluate the ability of MLLMs to comprehend multimodal punchlines. PunchBench distinguish itself from existing benchmarks in two key ways: First, it incorporates synonymous and antonymous captions to mitigate the risk of models relying on shortcuts in the original captions, achieving a more accurate assessment of their capabilities. Second, PunchBench includes a diverse range of punchline types, evaluation tasks, question formats, and multimodal content domains, ensuring a comprehensive evaluation. Our evaluation results highlight a significant gap between the performance of state-of-the-art MLLMs and human capabilities in understanding multimodal punchlines. To address this, we design the Simple-to-Complex Chain-of-Question (SC-CoQ), which effectively enhances the punchline comprehension ability of MLLMs and outperforms widely-used inference-time techniques such as in-context learning and chain-of-thought.

Limitations

In this work, we focus on multimodal punchline comprehension for the image-caption pairs, which only consist of static content. According to the evaluation results, MLLMs struggle with the punchline comprehension and fall behind humans. Extending this challenge to videos, where punchlines are often embedded in dynamic flows of information, poses even greater complexity. Unlike static images, videos require models to process temporal dynamics and integrate contextual cues across frames, demanding more advanced comprehension capabilities. Given the added challenges of punchline comprehension in video content, such as comedy, this area presents a meaningful avenue for further exploration. In future work, we aim to evaluate MLLMs' ability to understand punchlines within videos, advancing their capability to process and interpret dynamic multimodal content.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. In *ICCV*, pages 2425–2433. IEEE Computer Society.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multimodal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2506–2515. Association for Computational Linguistics.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an _Obviously_ perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4619–4629, Florence, Italy. Association for Computational Linguistics.

Poorav Desai, Tanmoy Chakraborty, and Md. Shad Akhtar. 2022. Nice perfume. how long did you marinate in it? multimodal sarcasm explanation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, pages 10563–10571. AAAI.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools.

Kilem L. Gwet. 2014. Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters. In *4th edition edition*, pages 1–38. Advanced Analytics, LLC.

Christian F. Hempelmann and Max Petrenko. 2015. An AI for humorously reframing interaction narratives with human users. In *Distributed, Ambient, and Pervasive Interactions - Third International Conference, DAPI 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings*, volume 9189 of *Lecture Notes in Computer Science*, pages 651–658. Springer.

Jack Hessel, Ana Marasovic, Jena D. Hwang, Lillian Lee, Jeff Da, Rowan Zellers, Robert Mankoff, and Yejin Choi. 2023. Do androids laugh at electric sheep? humor "understanding" benchmarks from the new yorker caption contest. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 688–714. Association for Computational Linguistics.

Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. 2024. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*.

Noman Islam, Zeeshan Islam, and Nazia Noor. 2017. A survey on optical character recognition system. *ArXiv*, abs/1710.05703.

Pu Jian, Donglei Yu, and Jiajun Zhang. 2024. Large language models know what is key visual entity: An llm-assisted multimodal retrieval for VQA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 10939–10956. Association for Computational Linguistics.

Liqiang Jing, Xuemeng Song, Kun Ouyang, Mengzhao Jia, and Liqiang Nie. 2023. Multi-source semantic graph-based multimodal sarcasm explanation generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11349–11361. Association for Computational Linguistics.

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. 2016. Densecap: Fully convolutional localization networks for dense captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4565–4574. IEEE Computer Society.

Shivani Kumar, Atharva Kulkarni, Md. Shad Akhtar, and Tanmoy Chakraborty. 2022. When did you become so smart, oh wise one?! sarcasm explanation in multi-modal multi-party dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5956–5968. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 26286–26296. IEEE.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning.

Yanxin Long, Youpeng Wen, Jianhua Han, Hang Xu, Pengzhen Ren, Wei Zhang, Shen Zhao, and Xiaodan Liang. 2023. Capdet: Unifying dense captioning and open-world detection pretraining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 15233–15243. IEEE.

Abdelkader El Mahdaouy, Abdellah El Mekki, Kabil Essefar, Nabil El Mamoun, Ismail Berrada, and Ahmed Khoumsi. 2021. Deep multi-task model for sarcasm detection and sentiment analysis in arabic language. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop, WANLP 2021, Kyiv, Ukraine (Virtual), April 9, 2021*, pages 334–339. Association for Computational Linguistics.

OpenAI. 2022. Introducing chatgpt. *CoRR*.

OpenAI. 2023a. Gpt-4v(ision) system card.

OpenAI. 2024. Gpt-4o.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun

Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Kun Ouyang, Liqiang Jing, Xuemeng Song, Meng Liu, Yupeng Hu, and Liqiang Nie. 2024. Sentiment-enhanced graph-based sarcasm explanation in dialogue. *CoRR*, abs/2402.03658.

Yang Qiao, Liqiang Jing, Xuemeng Song, Xiaolin Chen, Lei Zhu, and Liqiang Nie. 2023. Mutual-enhanced incongruity learning network for multi-modal sarcasm detection. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 9507–9515. AAAI Press.

Noam Shazeer. 2020. GLU variants improve transformer. *CoRR*, abs/2002.05202.

The Mistral AI Team. 2023. Mistral-7b-instruct-v0.2.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2023. Cogvlm: Visual expert for pretrained language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *CoRR*, abs/2407.10671.

A More Details for PunchBench

Here we provide more details for the dataset construction for both punchline perception and reasoning.

A.1 Source Data Collection & Annotation

We detail the data collection process.

Data Collection. 1) **Data filtering**. To reduce time-consuming and labor-cost, we introduce MLLM-based filtering method to answer the above questions to help filter the image-caption pairs. To prevent biases from MLLM, we randomly select a model from the set of evaluated MLLMs as the judge. It then is required to assess the quality of image-caption pairs by responding the following questions. Q1: "Whether it contains possible ethics conflict?" If No, go to the next question. Q2: "Whether the content of image is clearly visible?" If Yes, go to the next question. Q3: "Whether the caption is well-written from the aspects of fluency, length and readability?" If Yes, this image-caption pair passes the filtering process. To make sure the filtering quality, we randomly sample 500 image-caption pairs and then employ three undergraduates outside of this work to answer the above questions. Only 1 pairs of 500 fail to pass the manual filtering process, which verifies the reliability of automatic filtering process. 2) **Crowd voting**. To determine whether a collected image-caption pair contains a punchline, we conducted a crowd voting process using a questionnaire. Participants were asked, "Does the given image-caption pair make you laugh?" and could choose between "Yes" and "No." Each questionnaire was considered valid if it received more than 10 votes. If one option garnered over 80% of the votes, it was assigned as the label for the corresponding pair. Notably, for the pairs collected from the prior datasets, we adopted the original labels. Specifically, if the pair is identified as humorous or sarcastic in previous datasets, we regarded it as containing punchline.

Data Annotation. To acquire reasoning sentences for particular pairs containing punchline, we employ three human annotators to write reasoning sentence based on the content of image and caption. Specifically, we provide the annotated sarcasm or humor explanations for the pairs existing in the previous datasets, which can be referred to write reasoning sentence. Reasoning sentence must cover the key components in image and caption that convey punchline, and the annotators should state how the interplay between visual content and textual information conveys punchline.

A.2 Synonymous & Antonymous Caption

We illustrate Figure 16 to present the prompts to guide `gpt-3.5-turbo-0125` to generate synonymous and antonymous captions. And we provide more implementation details for context consistency adaption as follows. After identifying and isolating the two conflicting parts of inconsistent caption, we adopt word substitution and inversion to derive synonymous and antonymous captions. Specifically, we conduct word substitution for the former part and utilize word inversion for the latter part, if the generated caption maintain the punchline, we regard it as the synonymous caption. And we then conduct word substitution for the latter part and utilize word inversion for the former part, if the generated caption loses the punchline, we regard it as the antonymous caption.

A.3 Instruction Construction

Instruction Template. We provide various instruction templates for each question format, as follows. For punchline perception, the templates for *Yes/NO QA* are shown in Figure 21. The prompts for distractor captions generation and instruction templates for *Matching QA* are exhibited in Figure 22. The prompts for distractor options generation and instruction templates for *Multi-option QA* are exhibited in Figure 23. For punchline reasoning, the prompts for distractor reasoning sentence generation and instruction templates for *Yes/No QA* are exhibited in Figure 24. The instruction templates for *Matching QA* are exhibited in Figure 25. The instruction templates for *Generation QA* are exhibited in Figure 26.

A.4 Benchmark Comparison

We compare our PunchBench with the prior benchmarks related to multimodal punchline comprehension in Table 4. PunchBench shows superiority in domain, task, question format, punchline type.

B More Details for SC-CoQ

For the simplest question format *Yes/No QA*, we construct Inter-task SC-CoQ, *i.e.,* <*Yes/No* QA_m, *Yes/No* QA_n>, <*Yes/No* QA_n, *Yes/No* QA_m>. m denotes punchline perception and n means punchline reasoning. Specifically, For a specific *Yes/No* QA_m in punchline perception task, *Yes/No* QA_n> is filled

by a randomly sampled Yes/No QA from punchline reasoning task. For a specific *Yes/No* QA_n in punchline reasoning task, *Yes/No* QA_m> is implemented by the Yes/No QA from punchline reasoning task which shares the same image-caption pair. Notably, we integrate the response to the former question before the final question in the chain, as shown in n Figure 17. Similarly, for *Matching QA*, we adopt the same process. Then we can obtain SC-CoQ for *Yes/No QA* and *Matching QA*. Additionally, we exhibit some prompt examples of *Matching QA* using SC-CoQ in Figure 18 and Figure 19. For *Multi-option QA* and *Generation QA*, we implement <*Yes/No QA*, *Matching QA*, *Multi-option QA* or *Generation QA*> for a specific image-caption pair. The prompt examples of *Multi-option QA* are shown in Figure 20.

C More Details for Evaluation

Introduction for the MLLMs.

- **LLaVA** (Liu et al., 2024b). We use llava-v1.6-mistral-7b in our experiment. It reuses the pre-trained connector of LLaVA-1.5 (Liu et al., 2023b) and adopts Mistral (Team, 2023) as the base LLM.
- **GLM-4V** (GLM et al., 2024). It consists of GLMTransformer with 40 GLM Blocks and an EVA2CLIP Model with 63 Transformer Layers, along with a GLU mechanism.
- **Qwen2-VL** (Wang et al., 2024). Qwen2-VL employs a $675M$ parameter ViT across various-sized LLMs, ensuring that the computational load of the ViT remains constant regardless of the scale of the LLM. In terms of language processing, we have opted for the more powerful Qwen2 (Yang et al., 2024).
- **CogVLM2** (Hong et al., 2024). It is a stronger version of CogVLM, which is an extension of Vicuna, incorporating ViT (Dosovitskiy et al., 2021) as the vision encoder, a two-layer MLP (Shazeer, 2020) as adapter, and introducing Visual expert module.
- **GPT-4V** (OpenAI, 2023a) and **GPT-4o** (OpenAI, 2024). They are the leading MLLMs proposed by OpenAI.

Inference settings of the MLLMs. We present the inference settings, including decoding strategy and parameters of MLLMs in Table 2.

A	Strategy	Parameters
LLaVA	Random	T=0.7
GLM-4V	Top-k	k=3
Qwen2-VL	Top-p	p=0.7
CogVLM2	Random	T=0.7
GPT-4V	Greedy	-
GPT-4o	Greedy	-

Table 2: Decoding strategy and parameters for the evaluated MLLMs.

Introduction for in-conext learning and chain-of-thought. 1) In-context learning (ICL) (Brown et al., 2020). ICL enables models to perform tasks without explicit parameter updates by conditioning on a sequence of input-output examples, often referred to as a prompt. The model implicitly learns the task by observing these examples within the context, leveraging its pre-trained knowledge to generate predictions for new inputs. In this work, we adopt 3-shot prompt as one of the baselines. 2) Chain-of-Thought (CoT) (Wei et al., 2022). CoT prompting encourages models to generate intermediate reasoning steps in natural language, leading to more accurate and interpretable outputs for complex problems. By including step-by-step explanations in the prompt, CoT facilitates the decomposition of multi-step tasks, such as arithmetic, logical reasoning, or commonsense inference, into manageable sub-tasks. This approach significantly improves performance on reasoning-heavy benchmarks and highlights the potential of leveraging language models for tasks requiring structured thought processes.

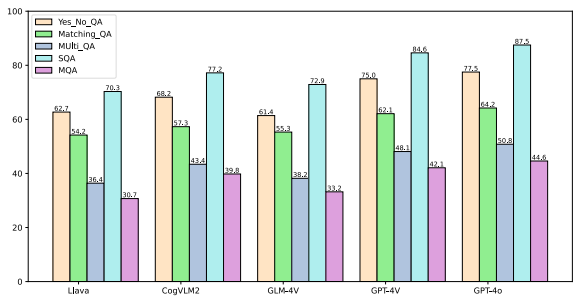


Figure 6: Performance across synonymous and antonymous captions.

D Evaluation and Analysis

D.1 Performance Variations

We compare the results cross the original, synonymous, and antonymous captions for all the evaluated MLLMs. The results for LLaVA, GLM-4V, Qwen2-VL, GPT-4V and GPT-4o cross different captions are exhibited in Figure 8, Figure 9, Figure 10, Figure 11, and Figure 12. As can be seen, synonymous and antonymous captions effectively eliminate shortcuts in the original captions, challenging models to fully comprehend the image-caption pairs. This leads to a more comprehensive evaluation of punchline comprehension capabilities. When using 3-shot and CoT methods, model performance with synonymous and antonymous captions lags behind that with original captions. However, when applying SC-CoQ, models show significant improvement across all caption types. This demonstrates that SC-CoQ enhances the models' ability to grasp the semantics of image-caption pairs, leading to better punchline comprehension.

D.2 Human Evaluation

To validate the reliability of automatic evaluation for *Generation QA*, we conduct human evaluation through pairwise test. Specifically, we first randomly sample 100 pairs of reasoning sentences from two candidate models. And we then involve three independent annotators (undergraduate students uninvolved in this work) to compare reasoning sentences generated by two models (A and B) for the same image-caption pair. The annotators are supposed to choose one of three options: *i.e.,* "A Wins", "A Draws B" and "B Wins". Finally, the winner is determined by the "Win" votes. If both models receive an equal number of "Win" votes, the final result is recorded as "A Draws B". In addition, we calculate Gwet's γ (Gwet, 2014) to represent inter-annotator agreement. The results for human evaluation of the generated reasoning sentences from evaluated models are shown in Table 3.

D.3 Correlation between Automatic and Human Evaluation

Human evaluation results, which are presented in Appendix D.2, show substantial agreement among annotators since Gwet's γ (Gwet, 2014) is consistently larger than 70%. And we exhibit the correlation between Automatic and Human evaluation in Figure 7 to emphasize the reliability of automatic evaluation for *Generation QA*. As observed, the models or methods that rank higher in human evaluation also show better accuracy in automatic evaluation. And our SC-CoQ achieves the best performance in both automatic and human evaluation. It not only verifies the credibility of the automatic evaluation results, but also further demonstrates the advantages of our SC-CoQ.

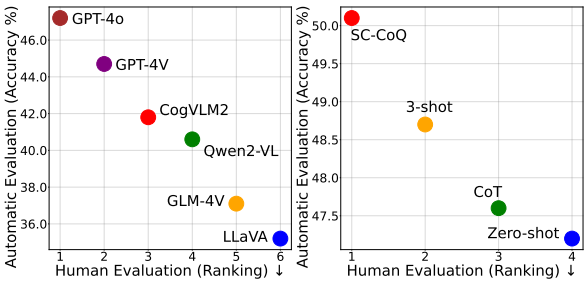


Figure 7: We show the relation between accuracy of automatic evaluation and ranking of human evaluation for evaluated MLLMs and different prompting methods.

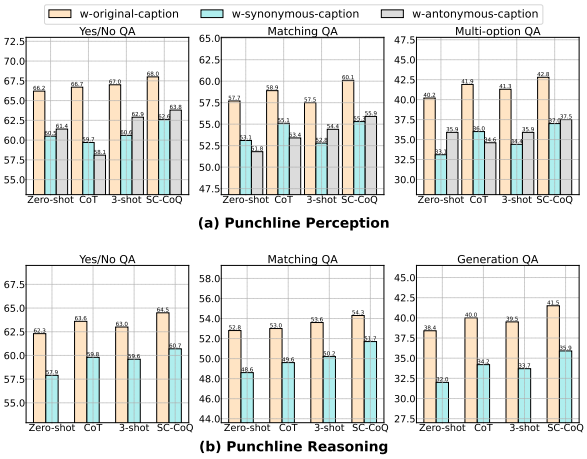


Figure 8: Performance comparison for LLaVA across original, synonymous and antonymous captions in zero-shot, 3-shot, CoT and our SC-CoQ.

D.4 More Qualitative Results

We provide result examples for *Matching QA* of punchline perception in Figure 13. As we can see, when using SC-CoQ, the model correctly answers the question, while failing when utilizing other prompting methods. For punchline reasoning task, we supply result examples for *Yes/No QA* and *Matching QA* in Figure 14. In addition, we present result examples for *Generation QA* in Figure 15.

A	B	A Wins (%)	A Draws B (%)	B Wins (%)	G-γ (%)
GLM-4V	**Llava**	57.0	18.0	25.0	82.6
Qwen2-VL	**GLM-4V**	67.0	23.0	10.0	77.4
CogVLM2	**Qwen2-VL**	41.0	37.0	22.0	80.4
GPT-4V	**CogVLM2**	59.0	20.0	21.0	78.1
GPT-4o	**GPT-4V**	47.0	30.0	23.0	74.6
GPT-4o (CoT)	**GPT-4o (Zero-shot)**	31.0	48.0	21.0	71.2
GPT-4o (3-shot)	**GPT-4o (CoT)**	39.0	38.0	23.0	76.3
GPT-4o (SC-CoQ)	**GPT-4o (3-shot)**	46.0	32.0	22.0	82.7

Table 3: Human estimation for *Generation QA*. Inter-annotator agreement is emphasized by Gwet's γ (Gwet, 2014), which is consistently larger than 70.0%, indicating substantial agreement.

Benchmarks	Domain	Task	Question Format	Punchline Type	#Num of Image-caption Pairs	#Num of Question-answer Pairs
MTSD (Cai et al., 2019)	Post	Sarcasm Classification	Single	Sarcasm	19,816	19,816
MORE (Desai et al., 2022)	Post	Sarcasm Explanation	Single	Sarcasm	3,510	3,510
HUB (Hessel et al., 2023)	Cartoon	Matching, Ranking and Explanation	Single	Humor	704	5,973
PunchBech	Cartoon, Post, Comment, Meme.	Punchline Perception, Punchline Reasoning	*Yes/No QA*, *Matching QA*, *Multi-option QA*, *Generation QA*.	Humor, Sarcasm	6,000	54,000

Table 4: Comparison between our PunchBench and previous benchmarks.

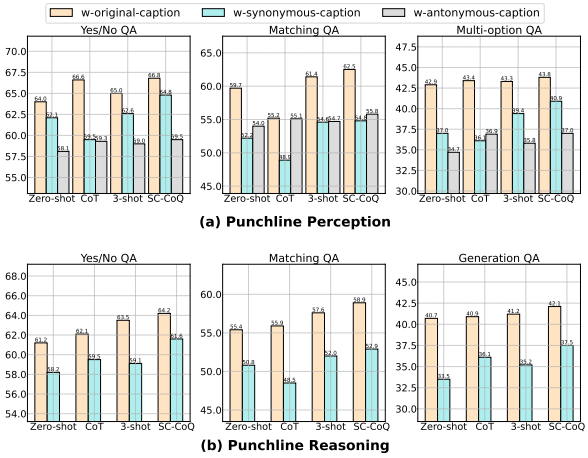


Figure 9: Performance comparison for GLM-4V across original, synonymous and antonymous captions in zero-shot, 3-shot, CoT and our SC-CoQ.

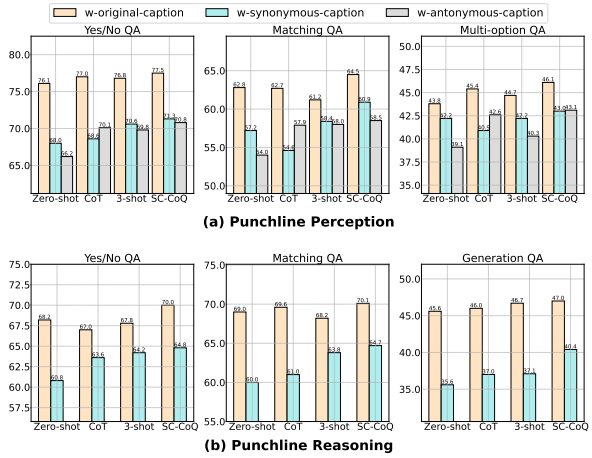


Figure 10: Performance comparison for Qwen2-VL across original, synonymous and antonymous captions in zero-shot, 3-shot, CoT and our SC-CoQ.

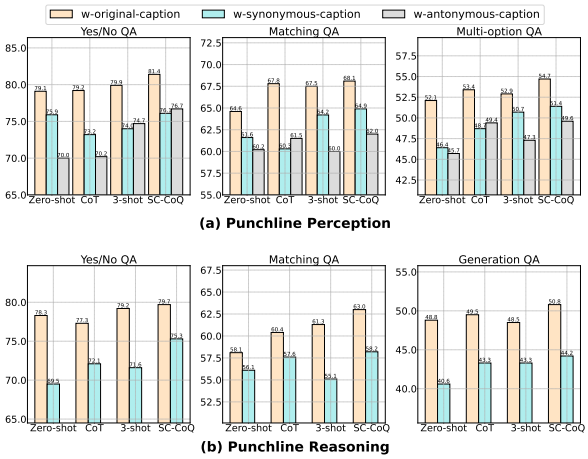


Figure 11: Performance comparison for GPT-4V across original, synonymous and antonymous captions in zero-shot, 3-shot, CoT and our SC-CoQ.

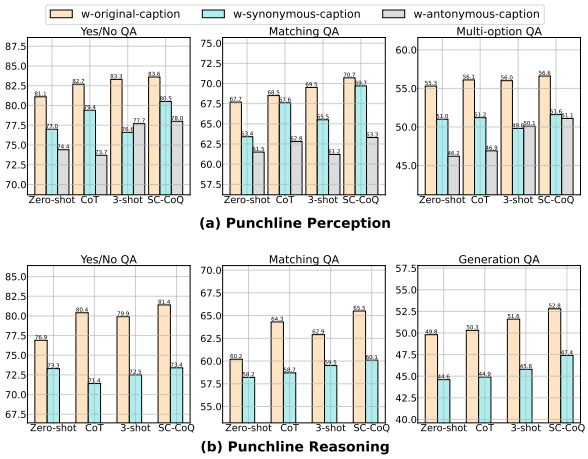


Figure 12: Performance comparison for GPT-4o across original, synonymous and antonymous captions in zero-shot, 3-shot, CoT and our SC-CoQ.

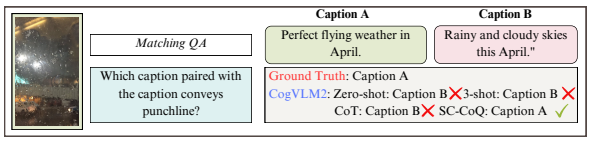


Figure 13: An example for qualitative analysis, where we show the responses from CogVLM2 to the *Matching QA* with different settings (*i.e.,* zero-shot, 3-shot, CoT and SC-CoQ).

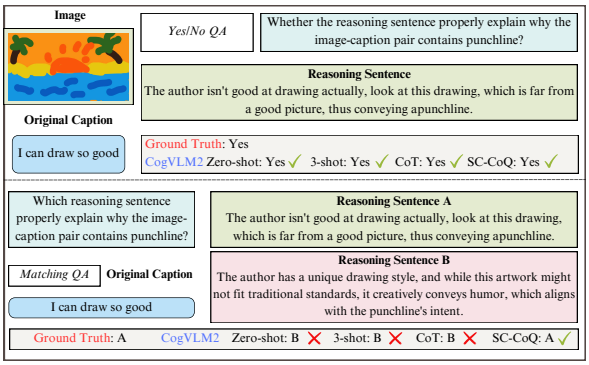


Figure 14: An example for qualitative analysis, where we show the responses from CogVLM2 to the *Yes/No QA* and *Matching QA* of punchline reasoning with different settings (*i.e.,* zero-shot, 3-shot, CoT and SC-CoQ).

E Documentation, Licensing, Potential risk and Intended Use of PunchBench

PunchBench encompasses $6,000$ image-caption pairs and $54,000$ question-answer pairs for multimodal punchline comprehension. We evaluate punchline comprehension in two levels: shallow-level punchline perception and deep-level punchline reasoning. We introduce three question formats for each task. We will release the dataset under CC BY-NC 4.0[3]. And for the preview use, the data source can be referred in the anonymous website[4] Notably, there may be some offensive information in the images, despite we have made efforts to exclude the potential offensive information in the collection and filtering process. Furthermore, PunchBench should only be used for research purpose only.

F Annotators Recruitment and Multimedia Platforms

For human baseline, we employed three undergraduates outside of the work as the annotators. For human evaluation, we asked another three undergraduate students to evaluate the quality of generated reasoning sentences. The information about the multimedia platforms we used is listed as follows. The social media platforms X[5], Instagram[6], and YouTube[7]. Additionally, we include image-caption pairs from the cartoon websites like CartoonMove-

[3] https://creativecommons.org/licenses/by-nc/4.0/.
[4] https://xhhxn9zbg2.wixsite.com/edge-1.
[5] https://x.com/.
[6] https://www.instagram.com/.
[7] https://www.youtube.com/.

She got it in the divorce.

driving home . thought the mountains . in the rain. with a vomiting kid . # happy vacation

CogVLM2	**3-shot**: The image-caption pair contains a punchline because it plays on the unexpected twist that Batman, a superhero, is not rich enough to afford a Batmobile, and instead, he has to rely on public transportation like the subway. **CoT**: The punchline of the image-caption pair arises from the surprising revelation that Batman, a famed crime-fighter, lacks the financial means to own a Batmobile, compelling him to utilize mundane methods of transport such as the subway. **SC-CoQ**: The image-caption pair contains a punchline because it tells of an unexpected twist: superhero Batman doesn't have enough money to afford the Batmobile, and instead has to rely on public transportation like the subway. Maybe it's because he's divorced.	**3-shot**: The image-caption pair contains a punchline because the situation described is not happy at all, it is quite unpleasant. **CoT**: The image-caption pair conveys the punchline due to the ironic contrast between the expected cheerfulness and the depicted scenario, which is far from delightful and rather discomforting. **SC-CoQ**: The image-caption pair contains a punchline because the rainy weather and vomiting kid mentioned imply he is not happy at all, very unpleasant..
GPT-4o	**3-shot**: The punchline of the image-caption pair stems from the absurdity of the situation. Divorce settlements usually involve the division of assets like property and belongings, yet here, the caption wryly indicates that the individual was awarded a lighthouse in the divorce proceedings, an unforeseen and ludicrously impractical "asset" to own. **CoT**: The image-caption pair contains a punchline because separation agreements pertain to the allocation of tangibles such as real estate and personal effects; however, here, the caption humorously suggests that the individual received a zoo as part of their settlement. **SC-CoQ**: The image-caption pair contains a punchline that juxtaposition of everyday life (two people waiting at a subway station) with the absurdity of one of them being dressed as Batman. The caption, "She got it in the divorce," sarcastically suggests that the man is left with a Batman costume as part of a divorce settlement, highlighting the unexpected and comedic nature of the situation.	**3-shot**: The image-caption pair contains a punchline because the caption is an example of sarcasm, as driving home through mountainous terrain in the rain with a sick child is a challenging and uncomfortable situation, which contrasts sharply with the notion of a 'happy vacation' typically associated with enjoyment and relaxation. **CoT**: The image-caption pair contains a punchline as the caption employs irony. Traveling back home amidst a downpour, traversing treacherous mountain paths with a child in poor health, presents a daunting and distressing ordeal, which is diametrically opposed to the conventional concept of a 'happy vacation' which is generally characterized by ease and rejuvenation. **SC-CoQ**: The image-caption pair contains a punchline because it describes a series of unpleasant events — driving in the rain through mountains with a vomiting child — yet concludes with the hashtag "#happy vacation," which contrasts the negative experience with an ostensibly positive sentiment, highlighting the irony.
Human	Batman would only take the subway if he didn't have access to the batmobile. In this situation, the onlooking man notices that, and batman insecurely retorts that he lost the iconic car in a divorce settlement. It's funny to think that batman has everyday concerns like divorce settlements.	The author writes happy vacation in the caption, but it's inconvenient driving home through the mountains in the rain with a vomiting kid, thus conveying a punchline

Figure 15: Two random samples of explanations generated by CogVLM2, GPT-4o, and human-written reasoning sentences. Notably, we present the generated reasoning sentences by CogVLM2 and GPT-4o prompted by 3-shot, CoT and SC-CoQ.

ment[8] and CartoonStock[9].

[8] https://www.cartoonmovement.com/.
[9] https://www.cartoonstock.com/.

Prompt 1	Caption: {caption}. Please substitute the sentiment, action, object words with synonymous words to generate a synonmous caption. The generated caption is:
Prompt 2	Caption: {caption}. Please substitute the sentiment, action, object words with antonymous words to generate a antonymous caption. The generated caption is:
Prompt 3	Caption: {caption}. Please identify whether semantics inconsistency exists in the caption. If yes, please isolate the two semantics conflicting parts in the caption. Only output the separated two parts without other additional texts.

Figure 16: Prompts used to guide `gpt-3.5-turbo-0125`, where Prompt1 guides the model to generate synonymous caption, Prompt2 guides it to derive antonymous caption, and Prompt3 guides it to identify the context inconsistency.

Yes/No QA	Whether the reasoning sentence properly explain why the image-caption pair contains punchline? Caption: {caption}. Reasoning sentence: {reasoning sentence}. Please answer with Yes or No. The answer is: {response}. Whether the image-caption pair contains punchline? Caption: {caption}. Please answer with Yes or No. The answer is:

Punchline Perception

Yes/No QA	Whether the image-caption pair contains punchline? Caption: {caption}. Please answer with Yes or No. The answer is: {response}. Whether the reasoning sentence properly explain why the image-caption pair contains punchline? Caption: {caption}. Reasoning sentence: {reasoning sentence}. Please answer with Yes or No. The answer is:

Punchline Reasoning

Figure 17: Prompt examples for *Yes/No QA* of punchline perception and reasoning using SC-CoQ.

Matching QA

Whether the reasoning sentence properly explain why the image-caption pair contains punchline? Caption: {caption}. Reasoning sentence: {reasoning sentence}.
Please answer with Yes or No. The answer is: {response}.
Whether the image-caption pair contains punchline? Caption: {caption}. Please answer with Yes or No. The answer is: {response}.
Which option properly explain why the image-caption pair contains punchline? Caption: {caption}. Option A: {reasoning sentence 1}. Option B: {reasoning sentence 2}.
Please answer with Option A or Option B. The answer is: {response}.
Which caption paired with the image conveys punchline? Caption A: {caption 1}. Caption B: {caption 2}.
Please answer with Caption A or Caption B. The answer is:

Figure 18: Prompt examples for *Matching QA* of punchline perception using SC-CoQ.

Matching QA

Whether the image-caption pair contains punchline? Caption: {caption}. Please answer with Yes or No. The answer is: {response}.
Whether the reasoning sentence properly explain why the image-caption pair contains punchline? Caption: {caption}. Reasoning sentence: {reasoning sentence}.
Please answer with Yes or No. The answer is: {response}.
Which caption paired with the image conveys punchline? Caption A: {caption 1}. Caption B: {caption 2}.
Please answer with Caption A or Caption B. The answer is: {response}.
Which option properly explain why the image-caption pair contains punchline? Caption: {caption}. Option A: {reasoning sentence 1}. Option B: {reasoning sentence 2}.
Please answer with Option A or Option B. The answer is:

Figure 19: Prompt examples for *Matching QA* of punchline reasoning using SC-CoQ.

Multi-option QA	Whether the image-caption pair contains punchline? Caption: {caption}. Please answer with Yes or No. The answer is: {response}. Which caption paired with the image conveys punchline? Caption A: {caption 1}. Caption B: {caption 2}. Please answer with Caption A or Caption B. The answer is: {response}. Which option describing the image-caption pair is correct? Options: A. {option1}. B. {option 2}. C. {option 3}. D. {option 4}.Please answer only with the option from {A, B, C, D}. The answer is:
Generation QA	Whether the reasoning sentence properly explain why the image-caption pair contains punchline? Caption: {caption}. Reasoning sentence: {reasoning sentence}. Please answer with Yes or No. The answer is:{response}. Which option properly explain why the image-caption pair contains punchline? Caption: {caption}. Option A: {reasoning sentence 1}. Option B: {reasoning sentence 2}. Please answer with Option A or Option B. The answer is:{response}. Please generate a reasoning sentence to explain why the image-caption pair contains punchline? Caption: {caption}. The reasoning sentence is:

Figure 20: Prompt examples for *Multi-option QA* and *Generation QA* using SC-CoQ.

Instruction Templates for Yes/No QA of Punchline Perception

Template 1: Whether the image-caption pair contains punchline? Caption: {caption}. Please answer with Yes or No. The answer is:

Template 2: Does the image-caption pair contain punchline? Caption: {caption}. Please respond by Yes or No. The response is:

Template 3: Is there any punchline in the image-caption pair? Caption: {caption}. Please output Yes or No. The output is:

Figure 21: Instruction templates for *Yes/No QA* of punchline perception.

Prompt used to guide GPT-4o to generate distractor caption

Prompt 1: Please generate a caption to describe the content of the input image. You cannot include any punchline (humor or sarcasm) in your caption.

Prompt 2: Please write a caption to sunmmarize the information of the input image. Please make sure no punchline (humor or sarcasm) in your caption.

Instruction Templates for Matching QA of Punchline Perception

Template 1: Which caption paired with the image conveys punchline? Caption A: {caption 1}. Caption B: {caption 2}. Please answer with Caption A or Caption B. The answer is:

Template 2: Which option conveys punchline when combined with the image? Option A: {caption 1}. Option B: {caption 2}. Please answer with Option A or Option B. The answer is:

Template 3: Which text conveys punchline when paired with the image? Text A: {caption 1}. Text B: {caption 2}. Please answer with Text A or Text B. The answer is:

Figure 22: Prompts used to guide GPT-4o to generate distractor caption and instruction templates for *Matching QA* of punchline perception.

Prompts used to guide ChatGPT to generate distractor options

Prompt 1: I will give you an image description and the corresponding caption. Description: {description}. Caption: {caption}. You should generate a distractor option describing the pair based on the description and caption.

Prompt 2: I will give you an image description and the corresponding caption. Description: {description}. Caption: {caption}. You should generate a correct option describing the pair based on the description and caption.

Instruction Templates for Multi-option QA of Punchline Perception

Template 1: Which option describing the image-caption pair is correct? Options: A. {option1}. B. {option 2}. C. {option 3}. D. {option 4}.Please answer only with the option from {A, B, C, D}. The answer is:

Template 2: Which description related to the image-caption pair is correct? Description: A. {option1}. B. {option 2}. C. {option 3}. D. {option 4}.Please only respond by A, B, C or D. The response is:

Template 3: Which statement describing the image-caption pair is correct? Statements: A. {option1}. B. {option 2}. C. {option 3}. D. {option 4}.Please only output A, B, C or D. The output is:

Figure 23: Prompts used to guide GPT-4o to generate distractor options and instruction templates for *Multi-option QA* of punchline perception.

Prompt used to guide ChatGPT to generate distractor reasoning sentence

Prompt 1: Please generate a new sentence to change the semantics of the following sentence. Sentence: {reasoning sentence}. The new sentence is:

Prompt 2: Please generate a sentence that has different semantics from the following sentence. Sentence: {reasoning sentence}. The generated sentence is:

Instruction Templates for Yes/No QA of Punchline Reasoning

Template 1: Whether the reasoning sentence properly explain why the image-caption pair contains punchline? Caption: {caption}. Reasoning sentence: {reasoning sentence}. Please answer with Yes or No. The answer is:

Template 2: Whether the reasoning sentence properly explain why the image-caption pair contains punchline? Caption: {caption}. Reasoning sentence: {reasoning sentence}. Please respond by Yes or No. The response is:

Template 3: Whether the reasoning sentence properly explain why the image-caption pair contains punchline? Caption: {caption}. Reasoning sentence: {reasoning sentence}. Please output Yes or No. The output is:

Figure 24: Prompts used to guide ChatGPT to generate distractor reasoning sentence and instruction templates for *Yes/No QA* of punchline reasoning.

Instruction Templates for Matching QA of Punchline Reasoning

Template 1: Which reasoning sentence properly explain why the image-caption pair contains punchline? Caption: {caption}. Reasoning sentence A: {reasoning sentence 1}. Reasoning sentence B: {reasoning sentence 2}. Please answer with Reasoning sentence A or Reasoning sentence B. The answer is:

Template 2: Which option properly explain why the image-caption pair contains punchline? Caption: {caption}. Option A: {reasoning sentence 1}. Option B: {reasoning sentence 2}. Please answer with Option A or Option B. The answer is:

Template 2: Which text properly explain why the image-caption pair contains punchline? Caption: {caption}. Text A: {reasoning sentence 1}. Text B: {reasoning sentence 2}. Please answer with Text A or Text B. The answer is:

Figure 25: Instruction templates for *Matching QA* of punchline reasoning.

Instruction Templates for Generation QA of Punchline Reasoning

Template 1: Please generate a reasoning sentence to explain why the image-caption pair contains punchline? Caption: {caption}. The reasoning sentence is:

Template 2: Please generate a reasoning sentence to interpret the reason why the image-caption pair contains punchline? Caption: {caption}. The reasoning sentence is:

Template 3: Please generate a reasoning sentence to explain why there is a punchline in the image-caption pair. Caption: {caption}. The reasoning sentence is:

Figure 26: Instruction templates for *Generation QA* of punchline reasoning.