# PUNMEMECN: A Benchmark to Explore Vision-Language Models' Understanding of Chinese Pun Memes

**Anonymous EMNLP submission**

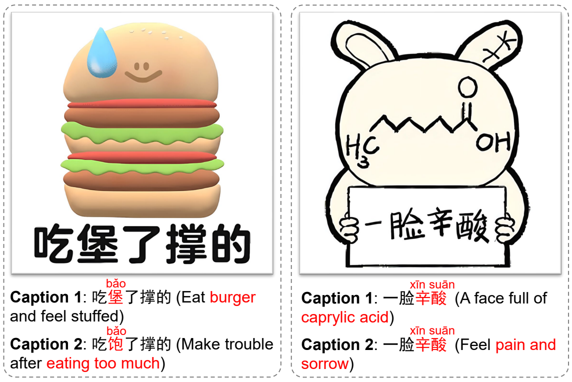
## Abstract

Pun memes, which combine wordplay with visual elements, represent a popular form of humor in Chinese online communications. Despite their prevalence, current Vision-Language Models (VLMs) lack systematic evaluation in understanding and applying these culturally-specific multimodal expressions. In this paper, we introduce PUNMEMECN, a novel benchmark designed to assess VLMs' capabilities in processing Chinese pun memes across three progressive tasks: pun meme detection, pun meme sentiment analysis, and chat-driven meme response. PUNMEMECN consists of 1,959 Chinese memes (653 pun memes and 1,306 non-pun memes) with comprehensive annotations of punchlines, sentiments, and explanations, alongside 2,008 multi-turn chat conversations incorporating these memes. Our experiments indicate that state-of-the-art VLMs struggle with Chinese pun memes, particularly with homophone wordplay, even with Chain-of-Thought prompting. Notably, punchlines in memes can effectively conceal potentially harmful content from AI detection. These findings underscore the challenges in cross-cultural multimodal understanding and highlight the need for culture-specific approaches to humor comprehension in AI systems.[1]

**Disclaimer**: *The samples presented in this paper may be considered offensive or vulgar to some readers.*

## 1 Introduction

Puns are a form of clever wordplay that exploit multiple meanings of words or phrases to create humor, irony, or other rhetorical effects (Attardo, 2009; Heller, 2014). With the development of the Internet and social networks, puns have evolved from pure text formats to multimodal expressions,



bǎo
**Caption 1**: 吃堡了撑的 (Eat burger and feel stuffed)

bǎo
**Caption 2**: 吃饱了撑的 (Make trouble after eating too much)

xīn suān
**Caption 1**: 一脸辛酸 (A face full of caprylic acid)

xīn suān
**Caption 2**: 一脸辛酸 (Feel pain and sorrow)

Figure 1: Two Chinese pun memes from our dataset, where double meanings (*Caption 1* and *Caption 2*) are conveyed by both image and text in the meme. We highlight the punchlines in red font, which often involve wordplay using homophony or polysemy. The letters above the Chinese characters are Pinyin, indicating the pronunciation of the Chinese text.

such as pun memes, which craft multiple meanings through both *caption* (text in the meme) and *image* (visual elements of the meme) (Lin et al., 2014). These memes are particularly popular in Chinese communities as they add layers of entertainment to online interactions (Mina, 2014; Hearn, 2020).

Consider the two Chinese pun memes in Figure 1 as examples. The caption in the left meme reads "吃堡了撑的", which literally translates to "eat burger and feel stuffed", connecting to the multi-layered burger in the image. However, "堡" (burger) and "饱" (full) share the same pronunciation "*bǎo*" in Chinese, evoking the common phrase "吃饱了撑的", which means "making trouble after eating too much" and often expresses dissatisfaction toward unnecessary trouble. The sweating expression in the image reinforces this emotional undertone. The pun in the right meme is comparatively simpler. The word "辛酸" in the caption can refer to both the caprylic acid molecule shown on the person's face and a feeling of hardship or sor-

---

[1] The resources of this paper will be released upon publication.

row, humorously conveying the sender's difficult situation.

Most previous research on puns has focused on English text puns (Miller et al., 2017; Yu et al., 2018). Before the advent of Large Language Models (LLMs), researchers typically designed specialized models or frameworks to address specific textual pun tasks, such as pun detection (Diao et al., 2018; Zhou et al., 2020), explanation (Sun et al., 2022), and generation (Luo et al., 2019; Mittal et al., 2022). More recently, with the rise of LLMs, the focus has shifted toward evaluating these models' comprehensive ability to understand textual puns (Xu et al., 2024; Chung et al., 2024). However, a significant gap remains in understanding how current AI models comprehend multimodal puns, especially pun memes that serve as popular and humorous expression units on Chinese social media. Studying pun memes not only reflects models' ability to process image-text ambiguity and punchlines but also contributes to future AI systems' thorough understanding of the vast and complex multimodal content on the Internet.

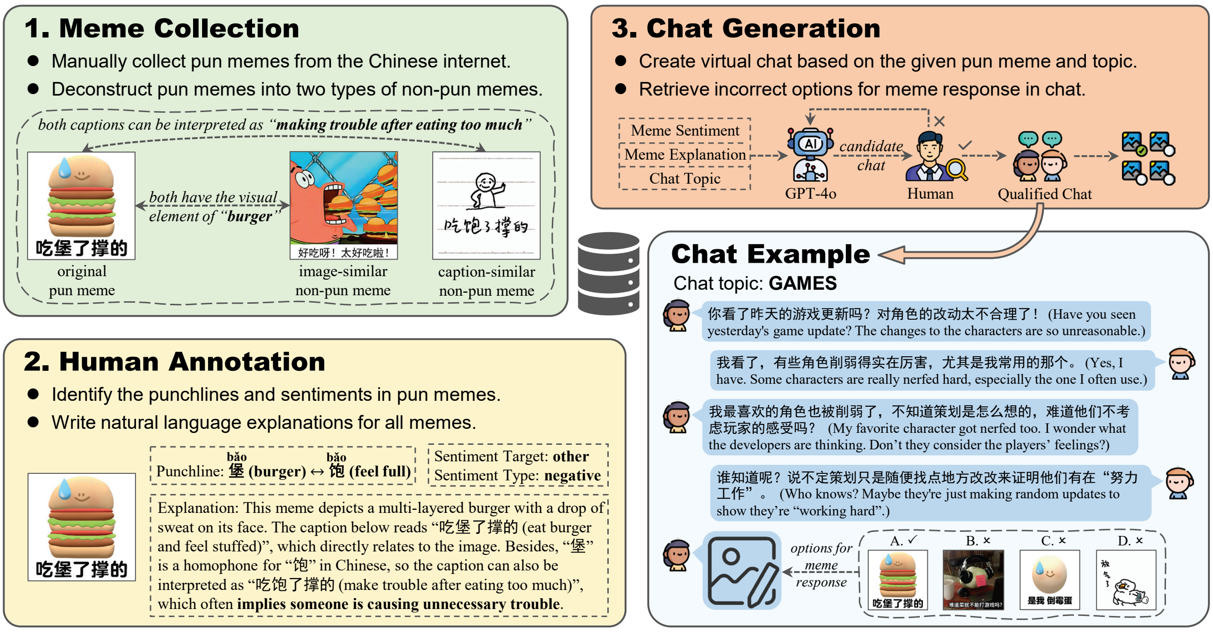
To address this gap, we propose PunMemeCN, a new benchmark designed to systematically evaluate Vision-Language Models' (VLMs) ability to comprehend and apply Chinese pun memes. We design three progressive tasks: pun meme detection, pun meme sentiment analysis, and chat-driven meme response. While the first two tasks focus on recognizing and analyzing individual pun memes, the third task examines models' potential to select appropriate pun memes for dialogue responses in online chat scenarios. To build the dataset for PunMemeCN, we collect diverse Chinese pun memes from the Internet and conduct detailed crowd-sourced annotations for their punchlines, sentiments, and other attributes. We also develop a semi-automated framework to generate multi-turn chats and retrieve challenging options for pun meme responses, increasing the difficulty of the third task.

Our extensive experiments with various VLMs under different prompting strategies demonstrate that most models struggle to distinguish between pun memes and non-pun memes in zero-shot settings. They also lag significantly behind human performance in sentiment analysis and chat responses involving pun memes. Although state-of-the-art models can better understand Chinese pun memes through Chain-of-Thought techniques, the homophone wordplay frequently found in these memes still creates substantial comprehension barriers. The main contributions of this paper are as follows:

- We propose PunMemeCN, a novel benchmark specifically designed to assess VLMs' ability to detect and analyze Chinese pun memes, and apply them in online chat scenarios. An efficient semi-automatic framework is introduced to generate simulated multi-turn chats and retrieve challenging distractors for meme response.
- We perform a comprehensive evaluation of mainstream VLMs. Our results reveal that most VLMs exhibit significant bias in pun meme detection while struggling with sentiment analysis and chat responses involving pun memes. This highlights the limitations of today's VLMs in understanding and applying Chinese pun memes.
- We conduct further experiments to explore feasible fine-tuning approaches for addressing pun detection bias, and analyze the comprehension barriers that punchlines in pun memes create for models, as well as their role in concealing potentially harmful content. These findings could benefit future research in this area.

## 2 Related Work

**Computational Pun Understanding** After early studies successfully model linguistic features of puns like ambiguity (Kao et al., 2016) and surprise (He et al., 2019), many subsequent works focus on training or fine-tuning specialized LMs to detect (Zou and Lu, 2019; Zhou et al., 2020), explain (Sun et al., 2022), or generate (Yu et al., 2020; Tian et al., 2022) puns. With the advent of general language models, recent works explore the proficiency of these large models to grasp puns. One leverages existing pun tasks to systematically evaluate LLMs' ability to understand puns (Xu et al., 2024), while another investigates how visual cues can help VLMs resolve textual pun ambiguities (Chung et al., 2024). However, most of these studies only consider English text puns and overlook other modalities of puns, such as traditional pun rebuses, which are studied by Zhang et al. (2024), and popular pun memes on the Internet, which are the focus of our research.



**1. Meme Collection**

- Manually collect pun memes from the Chinese internet.
- Deconstruct pun memes into two types of non-pun memes.

*both captions can be interpreted as “**making trouble after eating too much**”*

*both have the visual element of “**burger**”*

**2. Human Annotation**

- Identify the punchlines and sentiments in pun memes.
- Write natural language explanations for all memes.

Punchline: **堡 (burger)** ↔ **饱 (feel full)**

Sentiment Target: **other**
Sentiment Type: **negative**

Explanation: This meme depicts a multi-layered burger with a drop of sweat on its face. The caption below reads “吃堡了撑的 (eat burger and feel stuffed)”, which directly relates to the image. Besides, “堡” is a homophone for “饱” in Chinese, so the caption can also be interpreted as “吃饱了撑的 (make trouble after eating too much)”, which often **implies someone is causing unnecessary trouble**.

**3. Chat Generation**

- Create virtual chat based on the given pun meme and topic.
- Retrieve incorrect options for meme response in chat.

**Chat Example**

Chat topic: **GAMES**

你看了昨天的游戏更新吗？对角色的改动太不合理了！ (Have you seen yesterday's game update? The changes to the characters are so unreasonable.)

我看了，有些角色削弱得实在厉害，尤其是我常用的那个。 (Yes, I have. Some characters are really nerfed hard, especially the one I often use.)

我最喜欢的角色也被削弱了，不知道策划是怎么想的，难道他们不考虑玩家的感受吗？ (My favorite character got nerfed too. I wonder what the developers are thinking. Don’t they consider the players’ feelings?)

谁知道呢？说不定策划只是随便找点地方改改来证明他们有在“努力工作”。 (Who knows? Maybe they're just making random updates to show they’re “working hard”.)

Figure 2: Construction pipeline of the PunMemeCN benchmark. The punchlines and deeper meanings of the meme options in the chat example are as follows: *A)* "吃堡(饱)了撑的", which implies someone is causing unnecessary trouble; *B)* "难道菜(菜)就不能打游戏吗", which suggests someone is bad at gaming but still wants to play; *C)* "是我倒霉蛋(蛋)", which complains about being unlucky; *D)* "放气(弃)了", which means giving up on something.

**Meme Detection and Analysis** As an emerging cultural symbol, memes have become a popular research subject in communities (Hosseinmardi et al., 2016; Sharma et al., 2020; Tanaka et al., 2022). A mainstream area of research is harmful meme detection, where numerous researchers contribute datasets (Kiela et al., 2020; Pramanick et al., 2021) and models (Sharma et al., 2022; Ji et al., 2023) to identify different types of harmful memes. Some recent studies even use knowledge distilled from LLMs to further enhance the detector’s performance (Lin et al., 2024; Lu et al., 2024). Beyond harmful meme detection, meme research also includes but is not limited to, sentiment and emotion analysis (Mishra et al., 2023), visual metaphor interpretation (Hwang and Shwartz, 2023), and identification of persuasion techniques (Dimitrov et al., 2024). Our paper attempts to study the phenomenon of puns in memes, which other works have not systematically explored yet.

## 3 Overview of PunMemeCN

PunMemeCN is a comprehensive multimodal benchmark designed to systematically assess VLMs’ capabilities in recognizing, analyzing, and applying Chinese pun memes. It comprises 653 groups of Chinese memes, with each group containing one pun meme and two corresponding non-pun memes. All memes undergo meticulous manual annotation, including identification of potential punchlines, sentiment analysis, and detailed explanations. Additionally, PunMemeCN incorporates a semi-automatically constructed multi-turn chat dataset, where each chat concludes with a pun meme response. Figure 2 illustrates the overall construction and components of the benchmark.

### 3.1 Dataset Construction

**Meme Collection** We collect Chinese pun memes from major social media platforms including WeChat, Weibo, Tieba, and RedNote.[2] To maintain quality, we filter out duplicate memes and those with visible watermarks. Since pun memes derive their double meanings from both images and captions, we pair each pun meme with two related non-pun memes: one sharing similar visual elements and another with comparable caption content. These non-pun memes are obtained through keyword searches and image matching features provided by the aforementioned platforms. All collected memes are standardized by padding them into squares and adjusting to a uniform 448 × 448 resolution, ensuring all embedded captions remain clearly visible.

---

[2] The introduction to these Chinese social media platforms and methods for finding pun memes are placed in Appendix A.1.

| Chat Dataset | Perplexity↓ | BERTScore↑ | Distinct-1↑ | Distinct-2↑ |
|---|---|---|---|---|
| DuConv | 15.815 | 0.787 | 0.022 | 0.152 |
| NaturalConv | 18.397 | 0.839 | 0.015 | 0.183 |
| Our | **4.577** | **0.868** | **0.066** | **0.397** |

Table 1: Comparison of chat datasets on several indicators, including *1)* Perplexity, which reflects sentence fluency; *2)* BERTScore, which estimates the relevance between adjacent chat turns; *3)* Distinct-1 & Distinct-2, which measure the diversity of the chat content. The best results are **bolded**. Details of these indicators can be found in Appendix A.3.

**Human Annotation** We engage three annotators to perform comprehensive meme annotation.[3] Their annotation process consists of three primary tasks: *1)* Identifying potential punchlines in the memes, with particular attention to homophone wordplay; *2)* Analyzing the sentiment conveyed by each meme, categorizing both the sentiment target (as *self*, *both*, or *other*) and sentiment type (as *positive*, *neutral*, or *negative*); *3)* Composing natural language explanations that include literal descriptions of visual elements and interpretations of the memes' intended meanings. Each annotator independently labels all memes in the dataset. The final punchline and sentiment classifications are determined through majority voting, while we select the most comprehensive explanation from among the three provided. Our annotators demonstrate high reliability, achieving agreement rates of 95.6% for punchline identification and 74.2% for sentiment annotation.[4]

**Chat Generation** Since memes are frequently used in online conversations (Miltner, 2018), we create a multi-turn chat dataset incorporating pun memes. To avoid potential privacy concerns associated with collecting real-world chat data, we instruct GPT-4o (OpenAI, 2024) to generate simulated conversations between two individuals based on the sentiment and explanation of each pun meme. We specify that the final message in each conversation must include the meme. To ensure conversational diversity, we provide six distinct chat topics: sports, entertainment, technology, games, work, and daily life. GPT-4o is tasked with crafting contextually appropriate conversations that naturally culminate in the pun meme responses.[5] We employ an over-generation approach to produce numerous candidate conversations, which are subsequently reviewed and refined by our annotators, resulting in 2,008 high-quality Chinese chat entries. Table 1 compares our chat dataset with existing Chinese multi-turn chat datasets such as DuConv (Wu et al., 2019) and NaturalConv (Wang et al., 2021), highlighting the quality of our generated chat conversations. In addition to the correct meme responses, we also provide three incorrect meme responses as distractors for each chat entry, which is explained in detail in § 3.3.

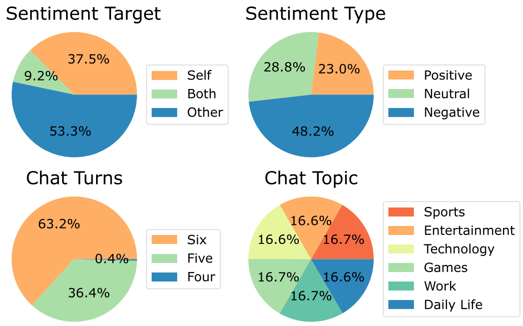


Figure 3: Statistical distributions within our dataset, showing sentiment targets, sentiment types, chat turns, and chat topics.

**PUNMEMECN Dataset** Our complete dataset contains 1,959 Chinese memes, comprising 653 pun memes and 1,306 non-pun memes. Each pun meme includes detailed annotations of its punchline, sentiment, and explanation, as well as two to five multi-turn chat conversations covering different topics. Figure 3 illustrates the distribution of sentiment targets, sentiment types, chat turns, and chat topics across the dataset.

## 3.2 Task Design

We propose three progressive tasks to comprehensively evaluate VLMs' capabilities in understanding Chinese pun memes. These tasks assess models' abilities in recognition, analysis, and application of pun memes.

**Task 1: Pun Meme Detection** The fundamental step in understanding pun memes is their correct identification. Our first task therefore focuses on pun meme detection, which requires determining whether a given meme contains a pun. We in-

---

[3] More information about our annotators is available in Appendix A.2.

[4] The "agreement" here refers to three annotators providing completely identical annotations.

[5] To maintain topical coherence, we limit conversations to 6 or fewer turns. The prompt used for generating chat conversations is available in Appendix D.1.

put each meme individually into the VLMs, which must classify it as either *pun* or *non-pun*.

**Task 2: Pun Meme Sentiment Analysis** In the second task, we examine whether VLMs can identify the sentiment expressions embedded within pun memes' punchlines. This task involves comprehensive sentiment analysis of pun memes. VLMs must determine both the sentiment target (*self*, *both*, or *other*) and sentiment type (*positive*, *neutral*, or *negative*) for each given pun meme.

**Task 3: Chat-Driven Meme Response** Considering the prevalent use of memes in online conversations, our final task evaluates models' ability to select appropriate pun meme responses in chat contexts. Specifically, we provide VLMs with a multi-turn chat conversation missing its final meme response, along with four candidate pun memes. The model must select the most contextually appropriate meme from these candidates. This represents a significant challenge, requiring comprehensive understanding of each pun meme's meaning and the ability to match it with the chat context.

### 3.3 Distractors for Meme Response in Chat

In Task 3, we encounter a problem: our chat dataset contains only correct pun meme responses without incorrect alternatives. To address this, we develop two methods for generating incorrect meme responses as distractors. The first method involves randomly selecting three pun memes from the dataset to serve as distractors. The second method employs a more sophisticated approach, retrieving three pun memes as distractors based on specific similarity criteria: *1) text similarity*, where the distractor meme's literal description bears resemblance to the chat content; *2) image similarity*, where the distractor meme visually resembles the correct meme; *3) sentiment similarity*, where the distractor meme expresses the same sentiment type as the correct meme. The meme options presented in the chat example in Figure 2 are selected using these three similarity criteria.[6] We consider random distractors to present less of a challenge than similarity-based distractors, therefore designating the former as the **easy mode** and the latter as the **hard mode**.

## 4 Experimental Setup

### 4.1 Models

To comprehensively evaluate existing models' capabilities in understanding Chinese pun memes, we select VLMs with diverse parameter scales and capabilities. These models can be categorized into two groups: state-of-the-art closed-source models (Claude-3.5-Sonnet (Anthropic, 2024) and GPT-4o (OpenAI, 2024)) and various open-source models (MiniCPM-V-2.6 (Yao et al., 2024), LLaVA-OneVision (Li et al., 2024), Qwen2.5-VL (QwenTeam, 2024), and InternVL-2.5 (Chen et al., 2024b)).[7] All selected VLMs demonstrate strong capabilities in Chinese language understanding and image processing.

### 4.2 Evaluation Metrics

PUNMEMECN consists of classification and multiple-choice tasks, enabling more precise and efficient automated evaluation compared to generative tasks. We employ the following metrics to assess VLM performance across three proposed tasks: *1)* For pun meme detection, we evaluate models using three metrics: True Positive Rate (**TPR**), measuring the proportion of correctly identified pun memes; True Negative Rate (**TNR**), measuring the proportion of correctly identified non-pun memes; and **F1-Score**, providing an overall performance assessment. *2)* For pun meme sentiment analysis, we measure accuracy across three aspects: sentiment target prediction ($\mathbf{Acc_{target}}$), sentiment type prediction ($\mathbf{Acc_{type}}$), and overall sentiment prediction ($\mathbf{Acc_{overall}}$). *3)* For chat-driven meme response, we assess accuracy in two scenarios: responses to easy distractors ($\mathbf{Acc_{easy}}$) and hard distractors ($\mathbf{Acc_{hard}}$).

## 5 Experimental Results

We evaluate the selected VLMs on PUNMEMECN from two perspectives. First, we conduct zero-shot evaluation (§ 5.1) to assess models' inherent understanding of Chinese pun memes. Second, we explore prompt engineering techniques (§ 5.2), including few-shot learning (Brown et al., 2020), Chain-of-Thought (CoT) (Wei et al., 2022), and pinyin hint, to enhance model performance. The experimental prompts are provided in Appendix D.2.

---

[6] Meme options B, C, and D in the chat example correspond to distractors based on text similarity, image similarity, and sentiment similarity, respectively. More details on the distractor construction process are available in Appendix A.4.

[7] The sources and endpoints of the evaluated VLMs are detailed in Appendix B.1.

| Model | Pun Meme Detection | | | Pun Meme Sentiment Analysis | | | Chat-Driven Meme Response | |
|---|---|---|---|---|---|---|---|---|
| | TPR | TNR | F1 | $\text{Acc}_{\text{target}}$ | $\text{Acc}_{\text{type}}$ | $\text{Acc}_{\text{overall}}$ | $\text{Acc}_{\text{hard}}$ | $\text{Acc}_{\text{easy}}$ |
| *Only Input Meme Image (Embedded Optical Caption)* | | | | | | | | |
| MiniCPM-V-2.6 | **0.998** | 0.001 | 0.499 | 0.424 | 0.541 | 0.254 | 0.291 | 0.282 |
| LLaVA-OneVision-7B | 0.643 | 0.309 | 0.425 | 0.367 | 0.506 | 0.188 | 0.271 | 0.254 |
| LLaVA-OneVision-72B | 0.948 | 0.053 | 0.493 | 0.381 | 0.489 | 0.179 | 0.296 | 0.378 |
| Qwen2.5-VL-7B | 0.997 | 0.002 | 0.499 | 0.472 | 0.568 | 0.302 | 0.274 | 0.371 |
| Qwen2.5-VL-72B | 0.960 | 0.517 | 0.656 | **0.538** | 0.587 | 0.344 | 0.362 | 0.481 |
| InternVL-2.5-8B | 0.969 | 0.068 | 0.506 | 0.447 | 0.507 | 0.226 | 0.306 | 0.390 |
| InternVL-2.5-26B | 0.965 | 0.244 | 0.555 | 0.492 | 0.511 | 0.265 | 0.338 | 0.436 |
| InternVL-2.5-38B | 0.969 | 0.308 | 0.578 | 0.507 | 0.495 | 0.238 | 0.346 | 0.464 |
| InternVL-2.5-78B | 0.853 | **0.739** | **0.718** | 0.493 | 0.507 | 0.253 | 0.370 | 0.472 |
| Claude-3.5-Sonnet | 0.739 | 0.720 | 0.643 | 0.489 | 0.555 | 0.307 | 0.320 | 0.430 |
| GPT-4o | 0.986 | 0.357 | 0.603 | 0.522 | **0.590** | **0.347** | **0.383** | **0.483** |
| *Input Both Meme Image and Caption Text* | | | | | | | | |
| MiniCPM-V-2.6 | **1.000** | 0.001 | 0.500 | 0.488 | 0.607 | 0.312 | 0.336 | 0.362 |
| LLaVA-OneVision-7B | 0.989 | 0.030 | 0.504 | 0.398 | 0.596 | 0.253 | 0.316 | 0.351 |
| LLaVA-OneVision-72B | 0.963 | 0.613 | 0.704 | 0.520 | 0.635 | 0.333 | 0.473 | 0.574 |
| Qwen2.5-VL-7B | 0.983 | 0.252 | 0.565 | 0.506 | 0.595 | 0.329 | 0.345 | 0.428 |
| Qwen2.5-VL-72B | 0.986 | 0.538 | 0.678 | 0.580 | 0.641 | 0.398 | 0.459 | 0.562 |
| InternVL-2.5-8B | 0.986 | 0.329 | 0.593 | 0.486 | 0.583 | 0.298 | 0.367 | 0.475 |
| InternVL-2.5-26B | 0.992 | 0.217 | 0.558 | 0.552 | 0.591 | 0.335 | 0.427 | 0.519 |
| InternVL-2.5-38B | 0.992 | 0.186 | 0.548 | 0.549 | 0.573 | 0.294 | 0.433 | 0.534 |
| InternVL-2.5-78B | 0.949 | **0.728** | **0.761** | 0.553 | 0.598 | 0.337 | 0.456 | 0.539 |
| Claude-3.5-Sonnet | 0.974 | 0.510 | 0.659 | 0.555 | 0.621 | 0.382 | **0.485** | **0.594** |
| GPT-4o | 0.995 | 0.377 | 0.614 | **0.586** | **0.653** | **0.414** | 0.457 | 0.575 |

Table 2: Results of the zero-shot evaluation on PunMemeCN. For each input setting, the top outcomes are **bolded**, and the second best are underlined.

### 5.1 Zero-shot Evaluation

The selected VLMs demonstrate strong image OCR capabilities.[8] This leads us to evaluate two input settings: image-only (with embedded optical caption) and image with additional caption text. Table 2 reveals several key findings: *1)* VLMs consistently perform better with additional caption text compared to image-only input. This suggests that while models can extract optical captions from meme images, they struggle to fully analyze pun meme punchlines and meanings based solely on visual input. *2)* Most VLMs show high TPR but low TNR in pun meme detection. This indicates a tendency to classify all memes as pun memes, which is problematic. *3)* VLMs demonstrate limited proficiency in analyzing sentiments in Chinese pun memes. Their performance significantly lags behind human annotators, who achieve average accuracies of 93.9% for sentiment target, 92.9% for sentiment type, and 87.5% for overall sentiment. *4)* VLMs face challenges in selecting appropriate pun meme responses, even in easy mode with randomly selected distractors. State-of-the-art models like Claude-3.5-Sonnet and GPT-4o show notably low accuracy. *5)* Our similarity-based meme distractors effectively challenge the models, resulting in significantly lower accuracy in hard mode compared to easy mode. In summary, **VLMs show considerable room for improvement in understanding and applying Chinese pun memes under zero-shot settings**.

### 5.2 Prompt Engineering Exploration

This section investigates methods to enhance VLM performance on PunMemeCN through prompt engineering. We explore three approaches: *1)* Few-shot learning, providing three examples for tasks 1 and 2, and one example for task 3.[9] *2)* CoT

---

[8] Our preliminary test of VLMs' OCR capabilities on Chinese memes shows that most models achieve up to 90% accuracy. Detailed results are available in Appendix B.2.

[9] Task 3 includes only one example because each chat contains four meme images. One-shot learning already processes eight meme images, approaching the multi-image localization

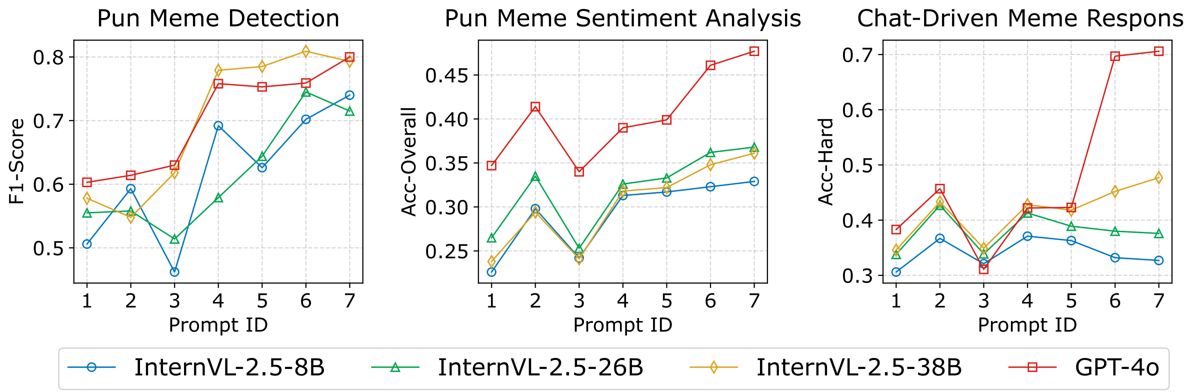


Figure 4: Results of the prompt engineering exploration on PUNMEMECN. We test the performance of four VLMs across seven different prompts for various tasks. The IDs of these prompts correspond to those in Table 3.

| Prompt | Meme Image | Meme Caption | Caption's Pinyin | Few-Shot | CoT Output |
|---|---|---|---|---|---|
| Prompt 1 | ✓ | × | × | × | × |
| Prompt 2 | ✓ | ✓ | × | × | × |
| Prompt 3 | ✓ | × | × | ✓ | × |
| Prompt 4 | ✓ | ✓ | × | ✓ | × |
| Prompt 5 | ✓ | ✓ | ✓ | ✓ | × |
| Prompt 6 | ✓ | ✓ | × | ✓ | ✓ |
| Prompt 7 | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 3: All the prompts used in our evaluation on PUNMEMECN. We mark their inputs and prompt techniques with ✓ and ×.

output, requiring models to describe meme content and analyze potential punchlines before concluding. *3)* Pinyin hint, providing caption pinyin to assist VLMs in identifying homophone wordplay.[10] We create five enhanced prompts by applying these methods individually or in combination. These prompts, along with the two zero-shot prompts from § 5.1, are listed in Table 3.

Figure 4 presents results from four VLMs across all seven prompts, revealing that: *1)* Few-shot learning with image-only input shows limited effectiveness across tasks, while few-shot learning with additional caption text primarily benefits pun meme detection. *2)* The CoT technique significantly improves advanced VLMs' understanding of Chinese pun memes. Models like GPT-4o can better analyze pun memes' sentiments and deeper meanings through CoT reasoning. However, complex homophone wordplay still challenges GPT-4o, as evidenced by incorrect punchline analyses. Several illustrative cases are provided in Appendix C. *3)* Pinyin hint generally fails to enhance VLM performance across tasks. Despite VLMs' good pinyin recognition (demonstrated in Appendix B.2), they struggle to utilize pinyin as a pronunciation cue for identifying homophone wordplay. In conclusion, **while stronger VLMs show better comprehension of Chinese pun memes through prompt engineering, particularly with CoT output, homophone wordplay remains a significant challenge to their understanding**.

## 6 Analysis and Discussion

### 6.1 Can Fine-tuning Resolve VLMs' Bias in Pun Meme Detection?

We observe that most VLMs perform better at recognizing pun memes compared to non-pun memes in zero-shot settings, as shown in Table 2. This phenomenon aligns with findings by (Xu et al., 2024), who observed similar patterns when LLMs distinguish between text puns and non-puns. These results suggest that large models exhibit a bias toward pun detection, tending to classify inputs as puns. We hypothesize that this bias originates from open-source pun datasets such as Semeval-2017-Task-7 (Miller et al., 2017) and ExPun (Sun et al., 2022), which predominantly contain pun entries with limited non-pun examples. This imbalance may lead LLMs and VLMs to develop stronger familiarity with puns while lacking sufficient exposure to non-pun concepts. To investigate whether increased exposure to labeled non-pun data can mitigate this bias, we conduct fine-tuning experiments using our dataset, which contains more non-pun memes than pun memes.[11]

---

capability limit of most VLMs.

[10] We generate pinyin for Chinese captions using the pypinyin Python package.

[11] Additional details about the fine-tuning setup for pun meme detection are provided in Appendix B.3.

| Model | Meme Image | Meme Caption | CoT Output | Our Dataset | | External |
|---|---|---|---|---|---|---|
| | | | | TPR | TNR | TNR |
| Qwen2.5-VL-7B | ✓ | × | × | 0.732 | 0.954 | 0.977 |
| Qwen2.5-VL-7B | ✓ | ✓ | × | 0.824 | 0.948 | 0.990 |
| Qwen2.5-VL-7B | ✓ | ✓ | ✓ | 0.712 | 0.964 | 0.971 |
| InternVL-2.5-8B | ✓ | × | × | 0.248 | **0.987** | **0.993** |
| InternVL-2.5-8B | ✓ | ✓ | × | **0.876** | 0.846 | 0.886 |
| InternVL-2.5-8B | ✓ | ✓ | ✓ | 0.810 | 0.843 | 0.824 |

Table 4: Results of fine-tuned VLMs on the pun meme detection task. We test fine-tuned models' performance on our validation set and an external non-pun meme dataset. The best results are in **bold**, and the second-best are underlined.

Table 4 reveals three key findings: *1)* **Fine-tuning VLMs on a dataset with sufficient non-pun memes effectively reduces the original bias in pun meme detection**, as evidenced by significant improvements in the TNR indicator compared to Table 2. *2)* For 7B size VLMs, the optimal prompt strategy involves inputting both the meme image and caption text, requiring direct model responses. **Despite our expectations that models would learn to identify homophone wordplay through CoT fine-tuning (with correct pun meme explanations), the fine-tuned models still struggle to analyze most homophone puns, resulting in inferior performance**. *3)* Our non-pun memes, constructed through pun disambiguation, successfully enhance the non-pun recognition capabilities of fine-tuned models, demonstrating effective generalization to external non-pun memes.

### 6.2 Does Puns in Memes Challenge Harmful Meme Detection?

Figure 3 indicates that approximately 25% of pun memes in our dataset express an "*other-negative*" sentiment. These memes often contain insults or mockery, concealing harmful content within their punchlines. This observation raises an important question: Can current harmful meme detectors, primarily trained on general harmful memes, effectively identify these harmful pun memes? To investigate this, we evaluate four representative harmful meme detectors on three categories: ToxiCN_MM (Lu et al., 2024) harmful memes, our non-pun harmful memes, and our pun harmful memes.[12] Our evaluation includes two state-of-the-art VLMs (GPT-4o and Claude-3.5-Sonnet) and two specialized models (CLIP-MKE (Lu et al., 2024) and HateCLIPper-RGCL (Mei et al., 2024)).[13]

| Model | ToxiCN_MM Harmful Meme | Our Non-pun Harmful Meme | Our Pun Harmful Meme |
|---|---|---|---|
| GPT-4o | 0.782 | 0.660 *(-0.112)* | 0.380 *(-0.402)* |
| Claude-3.5-Sonnet | 0.938 | 0.913 *(-0.025)* | 0.655 *(-0.283)* |
| CLIP-MKE | 0.659 | 0.643 *(-0.016)* | 0.535 *(-0.124)* |
| HateCLIPper-RGCL | 0.666 | 0.607 *(-0.059)* | 0.435 *(-0.231)* |

Table 5: Results of harmful meme detectors on different types of harmful memes. We use the accuracy metric to measure the performance of the detectors. The decrease in accuracy for non-pun and pun harmful memes compared to ToxiCN_MM harmful memes are highlighted in red font.

Table 5 presents two key findings: *1)* Most detectors show comparable performance in identifying ToxiCN_MM harmful memes and our non-pun harmful memes, suggesting that potential distribution differences between the datasets have minimal impact on detection accuracy. *2)* All detectors exhibit a significant performance decline when identifying harmful pun memes, demonstrating that **puns in harmful memes present a substantial challenge to current harmful meme detectors**.

## 7 Conclusion

In this paper, we introduce PUNMEMECN, a comprehensive benchmark designed to evaluate Vision-Language Models' (VLMs) capabilities in recognizing, analyzing, and applying Chinese pun memes. Through extensive experimentation with various VLMs under different prompt settings, we identify significant biases in pun meme detection and reveal limitations in sentiment analysis and meme response selection for online conversations. Our findings demonstrate effective approaches to mitigate detection bias and highlight homophone wordplay as the key challenge in understanding Chinese pun memes. We believe that PUNMEMECN and our insights will significantly advance research in multimodal pun understanding and contribute to the development of more sophisticated VLM capabilities.

## Limitations

While this study provides valuable insights into VLMs' understanding of pun memes, several limitations should be acknowledged. First, our benchmark primarily targets Chinese pun memes. This focus could limit the applicability of some exper-

---

[12] Following the definition of Chinese harmful memes in the ToxiCN_MM dataset, we selected 300 harmful non-pun memes and 200 harmful pun memes from our dataset.

[13] The prompts used for GPT-4o and Claude-3.5-Sonnet in harmful meme detection are available in Appendix D.3.

imental results to pun memes in other languages, given the variations in pun structures and meme usage across different languages. Such a limitation underscores the potential for future work to generalize findings to pun memes in other languages.

Second, our dataset contains only short chats with up to six chat turns, whereas real-world online chats often have dozens of chat turns. This leaves us unable to test VLMs' ability to respond with memes in lengthy chats. Additionally, we restrict memes to the last turn of chats, which may not be suitable for some memes, such as those used for greetings or simply making jokes. We expect future work to develop advanced chat generation frameworks that can produce longer chats and adjust meme placement more flexibly.

Finally, although our annotators perform very well, their annotations may still contain biases. The sentiments and meanings conveyed by memes largely depend on subjective judgment, meaning there may not be a single correct interpretation for some memes. Future studies can explore ways to achieve more objective meme annotations.

## Ethics Statement

We acknowledge that all authors of this work are aware of and comply with the ACL Code of Ethics and the Code of Conduct.

**Use of Human Annotations** Our institution recruits annotators for the PUNMEMECN annotations. We ensure that the privacy rights of the annotators are respected throughout the annotation process. Annotators receive compensation exceeding the local minimum wage and have consented to the use of their annotations for research purposes. Appendix A.2 provides further information on the annotations.

**Risks** All memes in PUNMEMECN are sourced from publicly available platforms such as WeChat, Weibo, Tieba, and RedNote. However, we cannot guarantee that they are free from socially harmful or toxic language. In addition, the annotations of Chinese pun memes rely on common sense, which may differ among individuals from diverse backgrounds. We use ChatGPT to correct grammatical errors in this paper.

## References

Anthropic. 2024. Claude-3.5-sonnet.

Salvatore Attardo. 2009. *Linguistic theories of humor*, chapter 3. Walter de Gruyter.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. 2024b. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Jiwan Chung, Seungwon Lim, Jaehyun Jeon, Seungbeen Lee, and Youngjae Yu. 2024. Can visual language models resolve textual ambiguity with visual cues? let visual puns tell you! In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2452–2469, Miami, Florida, USA. Association for Computational Linguistics.

Yufeng Diao, Hongfei Lin, Di Wu, Liang Yang, Kan Xu, Zhihao Yang, Jian Wang, Shaowu Zhang, Bo Xu, and Dongyu Zhang. 2018. WECA: A WordNet-encoded collocation-attention network for homographic pun recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2507–2516, Brussels, Belgium. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. SemEval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2009–2026, Mexico City, Mexico. Association for Computational Linguistics.

He He, Nanyun Peng, and Percy Liang. 2019. Pun generation with surprise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1734–1744, Minneapolis, Minnesota. Association for Computational Linguistics.

Kay Hearn. 2020. Peppa pig is gangsta: China's challenging memes. *Tracing Behind the Image*, pages 73–85.

Louis G Heller. 2014. Toward a general typology of the pun. In *Linguistic Perspectives on Literature (RLE Linguistics C: Applied Linguistics)*, pages 305–318. Routledge.

Homa Hosseinmardi, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. 2016. Prediction of cyberbullying incidents in a media-based social network. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 186–192. IEEE.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

EunJeong Hwang and Vered Shwartz. 2023. MemeCap: A dataset for captioning and interpreting memes. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1445, Singapore. Association for Computational Linguistics.

Junhui Ji, Wei Ren, and Usman Naseem. 2023. Identifying creative harmful memes via prompt based approach. In *Proceedings of the ACM Web Conference 2023*, pages 3868–3872.

Justine T Kao, Roger Levy, and Noah D Goodman. 2016. A computational model of linguistic humor in puns. *Cognitive science*, 40(5):1270–1285.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Chi-Chin Lin, Yi-Ching Huang, and Jane Yung-jen Hsu. 2014. Crowdsourced explanations for humorous internet memes based on linguistic theories. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 2, pages 143–150.

Hongzhan Lin, Ziyang Luo, Wei Gao, Jing Ma, Bo Wang, and Ruichao Yang. 2024. Towards explainable harmful meme detection through multimodal debate between large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 2359–2370.

Junyu Lu, Bo Xu, Xiaokun Zhang, Hongbo Wang, Haohao Zhu, Dongyu Zhang, Liang Yang, and Hongfei Lin. 2024. Towards comprehensive detection of chinese harmful memes. *arXiv preprint arXiv:2410.02378*.

Fuli Luo, Shunyao Li, Pengcheng Yang, Lei Li, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Pun-GAN: Generative adversarial network for pun generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3388–3393, Hong Kong, China. Association for Computational Linguistics.

Jingbiao Mei, Jinghong Chen, Weizhe Lin, Bill Byrne, and Marcus Tomalin. 2024. Improving hateful meme detection through retrieval-guided contrastive learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5333–5347, Bangkok, Thailand. Association for Computational Linguistics.

Tristan Miller, Christian Hempelmann, and Iryna Gurevych. 2017. SemEval-2017 task 7: Detection and interpretation of English puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 58–68, Vancouver, Canada. Association for Computational Linguistics.

Kate M Miltner. 2018. Internet memes. *The SAGE handbook of social media*, 55:412–428.

An Xiao Mina. 2014. Batman, pandaman and the blind man: A case study in social change memes and internet censorship in china. *Journal of Visual Culture*, 13(3):359–375.

Shreyash Mishra, S Suryavardan, Parth Patwa, Megha Chakraborty, Anku Rani, Aishwarya Reganti, Aman Chadha, Amitava Das, Amit Sheth, Manoj Chinnakotla, et al. 2023. Memotion 3: Dataset on sentiment and emotion analysis of codemixed hindi-english memes. *arXiv preprint arXiv:2303.09892*.

Anirudh Mittal, Yufei Tian, and Nanyun Peng. 2022. AmbiPun: Generating humorous puns with ambiguous context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1053–1062, Seattle, United States. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4o.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. MOMENTA: A multimodal framework for detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4439–4455, Punta Cana, Dominican Republic. Association for Computational Linguistics.

QwenTeam. 2024. Qwen2.5-vl.

Claude E Shannon. 1951. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.

Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. SemEval-2020 task 8: Memotion analysis- the visuolingual metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 759–773, Barcelona (online). International Committee for Computational Linguistics.

Shivam Sharma, Md Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2022. DISARM: Detecting the victims targeted by harmful memes. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1572–1588, Seattle, United States. Association for Computational Linguistics.

Jiao Sun, Anjali Narayan-Chen, Shereen Oraby, Alessandra Cervone, Tagyoung Chung, Jing Huang, Yang Liu, and Nanyun Peng. 2022. ExPUNations: Augmenting puns with keywords and explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4590–4605, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kohtaro Tanaka, Hiroaki Yamane, Yusuke Mori, Yusuke Mukuta, and Tatsuya Harada. 2022. Learning to evaluate humor in memes based on the incongruity theory. In *Proceedings of the Second Workshop on When Creative AI Meets Conversational AI*, pages 81–93, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Yufei Tian, Divyanshu Sheth, and Nanyun Peng. 2022. A unified framework for pun generation with humor principles. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3253–3261, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiaoyang Wang, Chen Li, Jianqiao Zhao, and Dong Yu. 2021. Naturalconv: A chinese dialogue dataset towards multi-turn topic-driven conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14006–14014.

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*.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, Florence, Italy. Association for Computational Linguistics.

Zhijun Xu, Siyu Yuan, Lingjie Chen, and Deqing Yang. 2024. "a good pun is its own reword": Can large language models understand puns? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11766–11782, Miami, Florida, USA. Association for Computational Linguistics.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Zhiwei Yu, Jiwei Tan, and Xiaojun Wan. 2018. A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1660, Melbourne, Australia. Association for Computational Linguistics.

Zhiwei Yu, Hongyu Zang, and Xiaojun Wan. 2020. Homophonic pun generation with lexically constrained rewriting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2870–2876, Online. Association for Computational Linguistics.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Tuo Zhang, Tiantian Feng, Yibin Ni, Mengqin Cao, Ruying Liu, Katharine Butler, Yanjun Weng, Mi Zhang, Shrikanth S Narayanan, and Salman Avestimehr. 2024. Creating a lens of chinese culture: A multimodal dataset for chinese pun rebus art understanding. *arXiv preprint arXiv:2406.10318*.

Yichao Zhou, Jyun-Yu Jiang, Jieyu Zhao, Kai-Wei Chang, and Wei Wang. 2020. "the boating store had its best sail ever": Pronunciation-attentive contextualized pun recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 813–822, Online. Association for Computational Linguistics.

Yanyan Zou and Wei Lu. 2019. Joint detection and location of English puns. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2117–2123, Minneapolis, Minnesota. Association for Computational Linguistics.

# A Details of Dataset Construction

## A.1 Sources of Memes

We collect all the pun memes and non-pun memes from the following Chinese social media platforms.

- **WeChat** (`https://weixin.qq.com`): China's largest messaging app with group chat and social feed features. Users frequently exchange memes in private and group chats, fostering a distinct digital subculture.
- **Weibo** (`https://weibo.com`): A Twitter-like platform for public discussions. Viral topics and visual content (text/images/videos) make it a common place for internet meme creation.
- **Tieba** (`https://tieba.baidu.com`): A classic platform full of Topic-based forums (*e.g.*, Gaming Bar, Food Bar) where niche communities develop inside jokes through persistent interactions.
- **RedNote** (`http://www.xiaohongshu.com`): A popular platform for youth-driven lifestyle content. While users primarily share daily life via photos/short videos, playful meme interactions commonly emerge in comment sections.

Most pun memes are directly obtained by searching for "双关/谐音梗表情包" (pun/homophone meme) on the platforms mentioned above. We review the top 200 search results from each platform and filter out duplicates and low-quality pun memes. Meanwhile, a small number of pun memes are modified from funny images found on these platforms. Non-pun memes are retrieved through a similar method, but the search terms are keywords or key elements from the caption or image.

## A.2 Crowd-sourcing

We have recruited a team of three undergraduates whose native language is Chinese. They are fully informed about the annotation requirements and details before starting their work. We pay each annotator $15 per hour, which is above the local minimum wage. Screenshots of the instructions and annotation interface are shown in Figure 5 and 6.

## A.3 Indicators for Measuring Chat Data

We use indicators such as Perplexity (Shannon, 1951), BERTScore (Zhang et al., 2020), Distinct-1&2 (Li et al., 2015) to measure the quality of multi-turn chat data from different perspectives.

**Perplexity** Perplexity is commonly used to measure a language model's ability to predict the next token. A lower perplexity on a fixed dataset indicates a better language modeling capability. Here, we use the fixed model, BERT-base-Chinese[14], to calculate the perplexity of sentences in different chat datasets, which can somewhat reflect the fluency of the chat sentences.

**BERTScore** BERTScore calculates the normalized inner product of two texts' embeddings, which reflects their semantic similarity. Here, we compute the BERTScore between adjacent turns in a multi-turn chat to estimate the relevance of the texts in adjacent turns. We also use BERT-base-Chinese for the calculations. A higher BERTScore indicates that the chat has greater coherence.

**Distinct-1&2** Distinct-n calculates the ratio of unique n-grams to total n-grams in a text dataset to assess its diversity. Here, we compute distinct-1 and distinct-2 to show the diversity of vocabulary and phrase usage in the chat dataset.

## A.4 Meme Distractors for Chat Response

As mentioned in § 3.3, we construct two types of meme distractors for chat-driven meme responses. Easy distractors randomly select three incorrect pun memes from the dataset, and hard distractors retrieve incorrect pun memes based on three similarity criteria: text, image, and sentiment similarities. The process of obtaining hard distractors involves the following steps: *1)* Use dense text embedding of BGE-M3 (Chen et al., 2024a) to calculate the similarity between all pun memes' literal descriptions and the given chat text, retaining the candidate memes of top $k$ with the highest text similarity. *2)* Use image embeddings of SigLIP-So400M (Zhai et al., 2023) to compute the similarity between other pun memes and the correct pun meme, retaining the candidate memes of the top $k$ with the highest image similarity. *3)* Randomly choose $k$ pun memes that match the sentiment type of the correct pun meme as sentiment-similar candidates. *4)* Randomly select one meme from each set of candidate memes for examination. If the selected meme conveys a different meaning than the correct meme, it is kept as the final distractor; otherwise, another selection is made. The examination work is conducted by GPT-4o, as it excels at determining whether two memes share the same

---

[14] `https://huggingface.co/google-bert/bert-base-chinese`

| Model | Source | Endpoint |
|---|---|---|
| MiniCPM-V-2.6 | Huggingface | openbmb/MiniCPM-V-2_6 |
| LLaVA-OneVision-7B | Huggingface | lmms-lab/llava-onevision-qwen2-7b-ov |
| LLaVA-OneVision-72B | Huggingface | lmms-lab/llava-onevision-qwen2-72b-ov-chat |
| Qwen2.5-VL-7B | Huggingface | Qwen/Qwen2.5-VL-7B-Instruct |
| Qwen2.5-VL-72B | Huggingface | Qwen/Qwen2.5-VL-72B-Instruct |
| InternVL-2.5-8B | Huggingface | OpenGVLab/InternVL2_5-8B |
| InternVL-2.5-26B | Huggingface | OpenGVLab/InternVL2_5-26B |
| InternVL-2.5-38B | Huggingface | OpenGVLab/InternVL2_5-38B |
| InternVL-2.5-78B | Huggingface | OpenGVLab/InternVL2_5-78B |
| Claude-3.5-Sonnet | Anthropic API | claude-3-5-sonnet-20241022 |
| GPT-4o | OpenAI API | gpt-4o-2024-08-06 |

Table 6: Sources and endpoints of all VLMs in our experiments

| **Model** | **Image OCR** | **Pinyin Transfer** |
|---|---|---|
| MiniCPM-V-2.6 | 0.927 | 0.900 |
| LLaVA-OneVision-7B | 0.456 | 0.848 |
| LLaVA-OneVision-72B | 0.579 | 0.975 |
| Qwen2.5-VL-7B | 0.954 | 0.925 |
| Qwen2.5-VL-72B | **0.964** | 0.989 |
| InternVL-2.5-8B | 0.899 | 0.823 |
| InternVL-2.5-26B | 0.952 | 0.973 |
| InternVL-2.5-38B | 0.951 | 0.980 |
| InternVL-2.5-78B | 0.946 | 0.982 |
| Claude-3.5-Sonnet | 0.892 | 0.986 |
| GPT-4o | 0.884 | **0.990** |

Table 7: Results of image OCR and pinyin transfer. We compute the average substring match rate between the model's output of captions and pinyin with the standard answers. The best results appear in **bold**, and the second best are underlined.

meaning after their explanations are provided. In practice, we set the parameter $k$ to 10.

# B Experimental Supplements

## B.1 Sources of Models

We list the sources and endpoints of all VLMs involved in the experiments in Table 6. These models can be easily accessed or downloaded to replicate our results.

## B.2 Image OCR and Pinyin Transfer

We randomly select 200 pun memes from the dataset to assess the ability of VLMs to extract image captions and convert Chinese captions into pinyin. The results in Table 7 illustrate that VLMs (except for the LLaVA-OneVision series) have strong image OCR capabilities and can accurately match Chinese characters with their pinyin. We provide the corresponding prompts in Table 14 and 15.

## B.3 Fine-tuning Setup

Our PunMemeCN contains 653 groups of Chinese memes (each group includes one pun meme and two relevant non-pun memes). For the fine-tuning experiments, we randomly select 500 groups as the training set and the remaining 153 as the validation set. We also collect approximately 1000 non-pun memes from ChineseBQB[15] as the external test set. We choose Qwen2.5-VL-7B and InternVL-2.5-8B for fine-tuning because both of them exhibit a clear bias in the pun meme detection task. We apply LoRA fine-tuning (Hu et al., 2022) to these two VLMs for 3 epochs with a learning rate of $1 \times 10^{-6}$, which ensures their performance essentially converges on the training set.

# C Case Study

We select three cases of GPT-4o's incorrect analysis of pun memes containing homophone wordplay during the CoT process, and present them in Table 8. Clearly, GPT-4o's main error is its inability to identify the correct homophone words behind the punchline.

# D Details of Prompts

## D.1 Prompt for Chat Generation

The prompt for multi-turn chat generation is shown in Table 9. During the chat generation process, we find that providing only text explanations and sentiments for memes yields a chat quality similar to also providing meme images. To save tokens and reduce costs, we decide not to input meme images. The temperature parameter is set to 1.0 to obtain diverse chat content.

## D.2 Prompts for Tasks in PunMemeCN

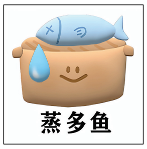
We craft seven types of prompts for each task in PunMemeCN, including two zero-shot prompts that use only meme images or provide additional caption texts, along with five enhanced prompts that utilize few-shot learning, CoT technique, and pinyin hints to improve model performance. Displaying all of them would be extremely space-consuming, so we only retain the zero-shot prompts that use both meme images and captions for three tasks, shown in Tables 10, 11, and 12. Other prompts are available in the code which will be released upon publication.

We primarily use Chinese prompts for the experiments since our memes are all in Chinese. However, one can also use the English prompts we provide, as we find that the language of the prompts does not significantly affect the results.

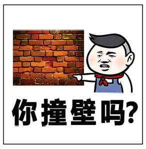
---

[15] https://github.com/zhaoolee/ChineseBQB

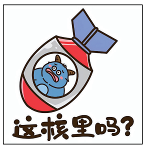
**GPT-4o's Incorrect Analysis of Pun Memes**



Case 1



Case 2



Case 3

*/* Meme in the Case 1 */*
**GPT-4o's Analysis**: This meme shows a steamer with a fish inside and a teardrop beside it. The caption reads "蒸多鱼" (steamed fish), which is a homophone for "争夺鱼" (fighting over fish). The meme uses this wordplay to humorously depict a competitive situation, possibly conveying sentiments about the intense competition for certain resources.
**Human Explanation**: The Meme shows a steamer basket containing a steamed fish. The steamer has a smiling face with a large blue sweat drop. Below the image is the Chinese caption "蒸多鱼" (steamed fish), which is a homophone for "真多余" (truly unnecessary). This meme uses the above wordplay to express that something is unnecessary or redundant, mocking actions or situations that seem pointless.

*/* Meme in the Case 2 */*
**GPT-4o's Analysis**: This meme shows a brick wall with a cartoon character pointing at it, with the caption "你撞壁吗？" (Do you hit the wall?). It plays on the double meaning of the Chinese phrase "撞壁", which literally refers to physically hitting a wall, but also metaphorically means encountering difficulties or obstacles. The meme humorously asks whether someone is facing challenges or setbacks, and is typically used to express concern about someone's situation.
**Human Explanation**: The meme features a cartoon character pointing at a brick wall, with the caption saying "你撞壁吗？" (Do you hit the wall?). This caption sounds similar to "你装逼吗？" (Are you showing off?). Therefore, the meme is used to tease or question someone's behavior, implying they are bragging or pretending to have abilities they don't possess.

*/* Meme in the Case 3 */*
**GPT-4o's Analysis**: The Meme shows a blue cartoon character inside a rocket or missile-shaped object, with the caption "这核里吗？" (Is this nuclear?). The word "核" (nuclear) in the caption serves a dual purpose - it references the character being inside what appears to be a nuclear bomb, while also sounding similar to "这里" (here). This creates a humorous way to ask if someone is in a particular place or situation.
**Human Explanation**: In this meme, a blue creature trapped inside a nuclear bomb appears terrified, with the caption "这核里吗？" (Is this nuclear?). This caption is a homophone for "这合理吗？" (Is this reasonable?). The meme uses wordplay to express dissatisfaction or questioning of a situation, emphasizing its unreasonableness or absurdity.

Table 8: Cases of GPT-4o's incorrect analysis of pun memes involving homophone wordplay compared to the correct explanation by humans. We mark GPT-4o's incorrect analyses in red, with the corresponding correct analyses in the human explanation marked in blue.

We set the temperature parameter to 0 for pun meme detection and text-driven meme response to minimize the randomness of the model predictions. For pun meme sentiment analysis, we increase the temperature to 0.7. This is because we oversample memes with a sentiment target of "both" (as they are rare in the dataset) to ensure the reliability and statistical validity of the evaluation results.

**Prompt for Chat Generation**

*/* 任务描述 */*
以下是一张编号为{img}的表情包，用表情包的解释文字替代了真实的图片。
表情包编号：{img}
表情包解释：{meme_explanation}
表情包情感指向对象：{sentiment_target}
表情包情感类别：{sentiment_type}
现在请你模拟甲、乙两人之间的一次网络聊天，以“发言人：发言内容”的格式算作一次完整发言，单次发言的内容必须为纯文字或者单张表情包，表情包的使用格式为“![表情包]({img})”。请你按照以下要求生成聊天内容：
1. 提供的表情包必须作为最后一次发言，根据情感指向和内涵的不同，可用该表情包回应对方，也可在纯文字发言后用表情包做二次补充发言。
2. 聊天应该顺畅合理地过渡到最后的表情包发言，且前面的纯文字发言不允许出现表情包中的字幕和画面元素。
3. 表情包的内涵和情感应与发言人在整个聊天中想表达的态度或观点相符合，聊天的情感倾向应与表情包一致。
4. 聊天主题必须与{topic}相关，可以有一定的延伸，但中途不能发生改变，也不能一直重复或复读某些相似的内容。
5. 聊天的总发言次数（包括表情包在内）应小于等于6次。
你的回答：

*/* Task Description */*
Below is a meme with the identifier {img}, where the text explanation for the meme replaces the actual image.
Meme Identifier: {img}
Meme Explanation: {meme_explanation}
Meme Sentiment Target: {sentiment_target}
Meme Sentiment Type: {sentiment_type}
Now, please simulate an online chat between person A and person B. The format "Speaker: Content of the speech" counts as one complete utterance. Each utterance must be either plain text or a single meme. The usage format for memes is "![meme]({img})". Please generate chat content according to the following requirements:
1. The provided meme must be used as the last utterance. Depending on its sentiment target and connotation, it can be used to respond to the other person or serve as a supplementary response after plain text.
2. The chat should smoothly and logically lead to the final meme utterance. No captions or visual elements from the meme are allowed in the preceding plain text utterances.
3. The connotation and sentiment of the meme should align with the attitude or viewpoint that the speaker wishes to express throughout the chat. The emotional tendency of the chat should match that of the meme.
4. The chat must relate to the {topic}. It can have some extension but should not change midway or repetitively echo similar content.
5. The total number of utterances in the chat (including the meme) should be less than or equal to 6.
Your response:

Table 9: Chinese prompt and its English translation for chat generation.

## D.3 Other Prompts

Tables 13, 14, and 15 contain the prompts for harmful meme detection, image OCR, and pinyin transfer, respectively. We adopt the prompt for harmful meme detection from Lu et al. (2024), with minor modifications.

**Prompt for Pun Meme Detection**

/* 表情包图像输入 */
{meme_image}
/* 定义 */
对于由图像和字幕组成的表情包，如果表情包字幕利用一词多义或谐音多义的方式有意使表情包具有两层或多层含义，且至少其中一层含义基于表情包的画面元素，则称该表情包为“双关表情包”，否则称为“非双关表情包”。
/* 任务描述 */
请你根据以上定义，判断给定的表情包图片是否为双关表情包。该表情包内的字幕为“{caption}”。你只需回答“双关”或“非双关”，不要输出其他内容。
你的回答：

*/* Input of Meme Image */*
{meme_image}
*/* Definition */*
For memes consisting of images and captions, if the caption intentionally uses polysemy or homophony to create two or more meanings, with at least one meaning based on the visual elements of the image, the meme is called a "pun meme". Otherwise, it is called a "non-pun meme".
*/* Task Description */*
Based on the above definition, determine whether the given meme is a pun meme or not. The caption within the meme is "{caption}". You only need to respond with "pun" or "non-pun". Do not output any other content.
Your response:

Table 10: Chinese and English prompt for pun meme detection.

**Prompt for Pun Meme Sentiment Analysis**

/* 表情包图像输入 */
{meme_image}
/* 定义 */
对于由图像和字幕组成的表情包，如果表情包字幕利用一词多义或谐音多义的方式有意使表情包具有两层或多层含义，且至少其中一层含义基于表情包的画面元素，则称该表情包为“双关表情包”，否则称为“非双关表情包”。
/* 任务描述 */
给定的表情包图片是满足以上定义的双关表情包。请你判断该表情包用于网络聊天时表达的情感，包括情感指向对象（自己、双方、他人）和情感类别（积极、中性、消极）。该表情包内的字幕为“{caption}”。你只需直接回答表情包的情感，回答格式为JSON格式：{"情感指向对象": "XXX", "情感类别": "XXX"}，不要输出其他内容。
你的回答：

*/* Input of Meme Image */*
{meme_image}
*/* Definition */*
For memes consisting of images and captions, if the caption intentionally uses polysemy or homophony to create two or more meanings, with at least one meaning based on the visual elements of the image, the meme is called a "pun meme". Otherwise, it is called a "non-pun meme".
*/* Task Description */*
The given meme image is a pun meme that meets the above definition. Please determine the sentiment conveyed by the meme when used in online chat, including the sentiment target (self, both, other) and the sentiment type (positive, neutral, negative). The caption in this meme is {caption}. You only need to respond with the meme sentiment in the JSON format {"Sentiment Target": "XXX", "Sentiment Type": "XXX"}. Do not output any other content.
Your response:

Table 11: Chinese and English prompt for pun meme sentiment analysis.

**Prompt for Chat-Driven Meme Response**

/* 表情包图像输入 */
{meme_image_1} {meme_image_2} {meme_image_3} {meme_image_4}
/* 定义 */
对于由图像和字幕组成的表情包，如果表情包字幕利用一词多义或谐音多义的方式有意使表情包具有两层或多层含义，且至少其中一层含义基于表情包的画面元素，则称该表情包为“双关表情包”，否则称为“非双关表情包”。
/* 任务描述 */
给定的四张表情包图片都是满足以上定义的双关表情包，现将这四张表情包按顺序分别记为第一、二、三、四张表情包，其中：
第一张表情包内的字幕为“{caption_1}”；
第二张表情包内的字幕为“{caption_2}”；
第三张表情包内的字幕为“{caption_3}”；
第四张表情包内的字幕为“{caption_4}”。
以下是甲、乙两人的一次网络聊天记录，缺少了最后的表情包回复。请你根据聊天内容，人物关系和情绪氛围，从四张表情包中选出最合适的一个作为回复。仅需回答选项的字母（A-D），不要输出其他内容。
聊天记录：{chat_history}
选项：{choices}
你的回答：

*/* Input of Meme Images */*
{meme_image_1} {meme_image_2} {meme_image_3} {meme_image_4}
*/* Definition */*
For memes consisting of images and captions, if the caption intentionally uses polysemy or homophony to create two or more meanings, with at least one meaning based on the visual elements of the image, the meme is called a "pun meme". Otherwise, it is called a "non-pun meme".
*/* Task Description */*
The four given meme images are pun memes that meet the above definition. Here we label these four memes in order as the first, second, third, and fourth meme. From these memes, we can know that:
The caption within the first meme is "{caption_1}";
The caption within the second meme is "{caption_2}";
The caption within the third meme is "{caption_3}";
The caption within the fourth meme is "{caption_4}".
Below is an online chat record between two people, A and B, which lacks a final meme reply. Please choose the most suitable meme from the four as a reply based on the chat content, character relationship, and emotional tone. Simply respond with the letter (A-D) representing your choice. Do not output any other content.
Chat Record: {chat_history}
Options: {choices}
Your response:

Table 12: Chinese and English prompt for chat-driven meme response.

**Prompt for Harmful Meme Detection**

/* 表情包图像输入 */
{meme_image}
/* 定义 */
中文有害表情包是由图像和图像内字幕组成的多模态单元，具有对个人、组织、社区、社会群体等造成伤害的潜力。它们往往反映和强化了中国互联网上潜在的负面价值观和文化态度。
/* 任务描述 */
以上是一张网络聊天用的中文表情包，其中包含字幕“{caption}”。请根据中文有害表情包的定义，将其分类为有害或无害。符合以下任何一个标准的表情包都将被判定为有害。
1. 表情包表达了对特定个人或社会群体的厌恶、偏见或刻板印象。
2. 表情包包含了讽刺或粗鲁的内容，没有特定的目标。
3. 表情包暗示了性意图，以引起性冲动。
4. 表情包包含了颓废和绝望的情绪，或传达了一种自我否定的态度。
评估步骤:
1. 结合表情包图像和字幕分析表情包的含义；
2. 与给定的评估标准进行比较，检查是否有任何匹配；
3. 返回“有害”或“无害”的结论。
要求必须以JSON格式输出三个评估步骤，形式为{"表情包含义": "XXXX", "匹配评估标准": "XXXX", "结论": "有害/无害"}

/* *Input of Meme Image* */
{meme_image}
/* *Definition* */
Chinese harmful memes are multimodal units consisting of an image and inline caption that have the potential to cause harm to individuals, organizations, communities, or social groups. They often reflect and reinforce underlying negative values and cultural attitudes on the Chinese Internet.
/* *Task Description* */
The image above is a Chinese meme used in online chat. The caption in the meme is "{caption}". Based on the definition of Chinese harmful memes, please classify it as harmful or harmless. A meme will be deemed harmful if it meets any of the following criteria:
1. The meme expresses disgust, prejudice, or stereotypes towards specific individuals or social groups.
2. The meme contains sarcastic or rude content without a specific target.
3. The meme implies sexual intent to arouse sexual impulses.
4. The meme integrates decadent and desperate emotions or conveys a self-negative attitude.
Evaluation Steps:
1. Analyze the meaning of the meme by considering both the image and the caption.
2. Compare the meme against the criteria to check for any matches;
3. Return the conclusion as "harmful" or "harmless".
The output must be in JSON format for the three evaluation steps, structured as {"Meme_Meaning": "XXXX", "Criteria_Matching": "XXXX", "Conclusion": "harmful/harmless"}.

Table 13: Chinese prompt and its English translation for harmful meme detection.

**Prompt for Image OCR**

/* 表情包图像输入 */
{meme_image}
/* 任务描述 */
以上是一张网络聊天用的中文表情包，请提取出该表情包中的主要字幕。不要输出其他内容。
表情包中的字幕：

/* *Input of Meme Image* */
{meme_image}
/* *Task Description* */
Here is a Chinese meme used in online chat. Please extract the main caption/subtitle from the meme. Do not output any other content.
Meme caption/subtitle:

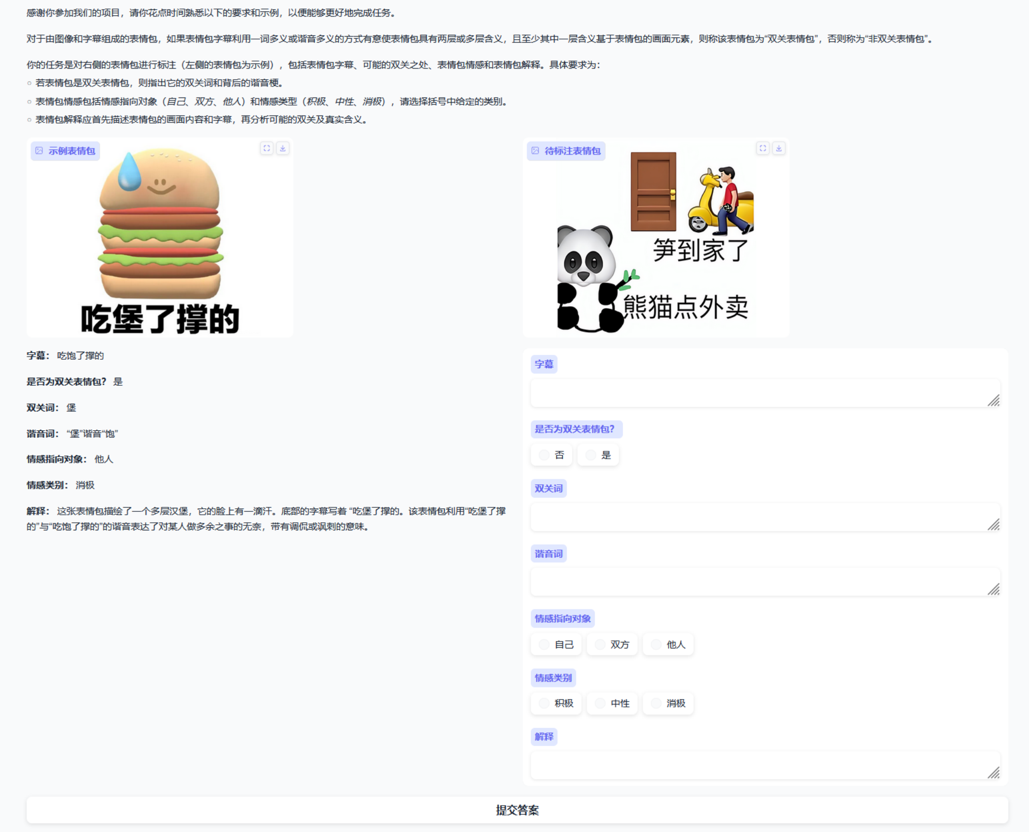
Table 14: Chinese prompt and its English translation for image OCR.

**Prompt for Pinyin Transfer**

/* 任务描述 */
拼音是一种辅助汉字读音的工具。请将以下中文文本转化成对应的带有声调的拼音。不要输出其他内容。
中文文本：{caption} -> 拼音：

/* *Task Description* */
Pinyin is a tool that aids in the pronunciation of Chinese characters. Please convert the following Chinese text into corresponding pinyin with tones. Do not output any other content.
Chinese text: {caption} -> Pinyin:

Table 15: Chinese prompt and its English translation for pinyin transfer.



感谢你参加我们的项目，请你花点时间熟悉以下的要求和示例，以便能够更好地完成任务。

对于由图像和字幕组成的表情包，如果表情包字幕利用一词多义或谐音多义的方式有意使表情包具有两层或多层含义，且至少其中一层含义基于表情包的画面元素，则称该表情包为“双关表情包”，否则称为“非双关表情包”。

你的任务是对右侧的表情包进行标注（左侧的表情包为示例），包括表情包字幕、可能的双关之处、表情包情感和表情包解释。具体要求为：
- 若表情包是双关表情包，则指出它的双关词和背后的谐音梗。
- 表情包情感包括情感指向对象（*自己、双方、他人*）和情感类型（*积极、中性、消极*），请选择括号中给定的类别。
- 表情包解释应首先描述表情包的画面内容和字幕，再分析可能的双关及真实含义。

**字幕：** 吃饱了撑的

**是否为双关表情包？** 是

**双关词：** 堡

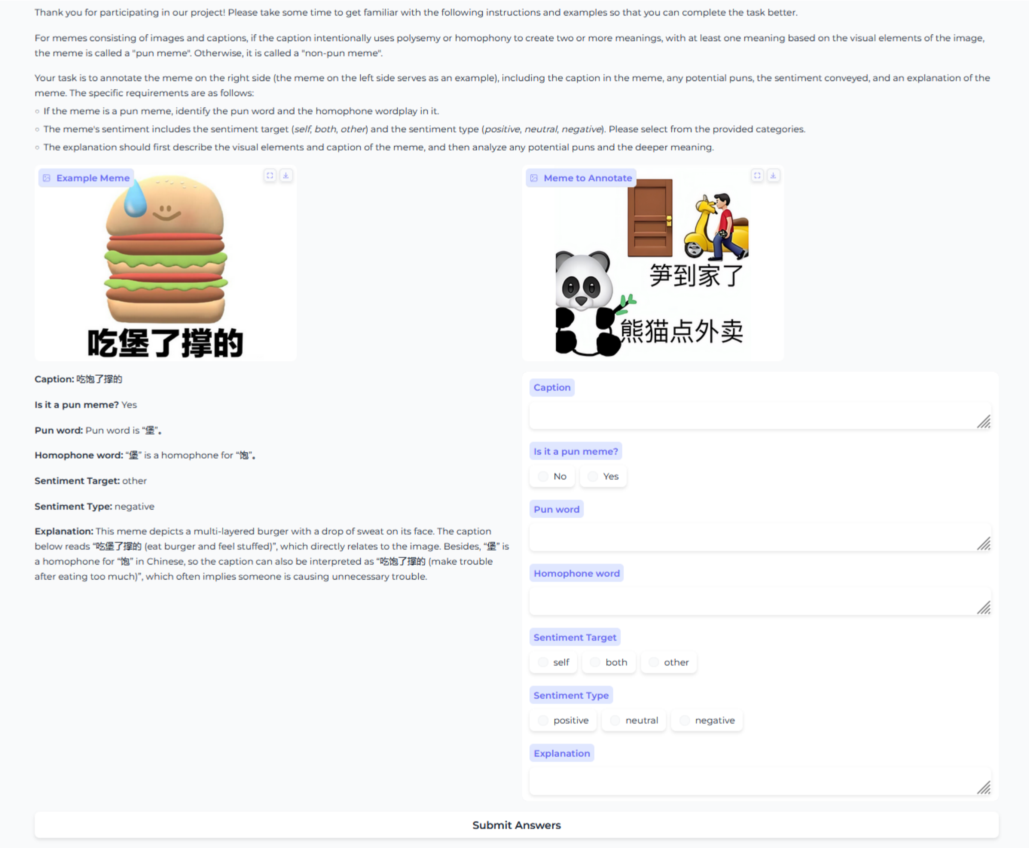
**谐音词：** “堡”谐音“饱”

**情感指向对象：** 他人

**情感类别：** 消极

**解释：** 这张表情包描绘了一个多层汉堡，它的脸上有一滴汗。底部的字幕写着 “吃堡了撑的。该表情包利用“吃堡了撑的”与“吃饱了撑的”的谐音表达了对某人做多余之事的无奈，带有调侃或讽刺的意味。

Figure 5: The screenshot of the original meme annotation interface.



Thank you for participating in our project! Please take some time to get familiar with the following instructions and examples so that you can complete the task better.

For memes consisting of images and captions, if the caption intentionally uses polysemy or homophony to create two or more meanings, with at least one meaning based on the visual elements of the image, the meme is called a "pun meme". Otherwise, it is called a "non-pun meme".

Your task is to annotate the meme on the right side (the meme on the left side serves as an example), including the caption in the meme, any potential puns, the sentiment conveyed, and an explanation of the meme. The specific requirements are as follows:
- If the meme is a pun meme, identify the pun word and the homophone wordplay in it.
- The meme's sentiment includes the sentiment target (*self*, *both*, *other*) and the sentiment type (*positive*, *neutral*, *negative*). Please select from the provided categories.
- The explanation should first describe the visual elements and caption of the meme, and then analyze any potential puns and the deeper meaning.

**Caption:** 吃饱了撑的

**Is it a pun meme?** Yes

**Pun word:** Pun word is “堡”。

**Homophone word:** “堡” is a homophone for “饱”。

**Sentiment Target:** other

**Sentiment Type:** negative

**Explanation:** This meme depicts a multi-layered burger with a drop of sweat on its face. The caption below reads “吃堡了撑的 (eat burger and feel stuffed)”, which directly relates to the image. Besides, “堡” is a homophone for “饱” in Chinese, so the caption can also be interpreted as “吃饱了撑的 (make trouble after eating too much)”, which often implies someone is causing unnecessary trouble.

Figure 6: The screenshot of the translated meme annotation interface.