"I See What You Did There": Can Large Vision-Language Models Understand Multimodal Puns?

Naen Xu[1], **Jiayi Sheng**[2], **Changjiang Li**[3], **Chunyi Zhou**[1], **Yuyuan Li**[4], **Tianyu Du**[1,5*], **Jun Wang**[6*], **Zhihui Fu**[6], **Jinbao Li**[7], **Shouling Ji**[1]
[1]Zhejiang University, [2]Beihang University,
[3]Palo Alto Networks, [4]Hangzhou Dianzi University,
[5]Ningbo Global Innovation Center, Zhejiang University,
[6]OPPO Research Institute, [7]Qilu University of Technology
{xunaen, zjradty}@zju.edu.cn, junwang.lu@gmail.com

Abstract

Puns are a common form of rhetorical wordplay that exploits polysemy and phonetic similarity to create humor. In multimodal puns, visual and textual elements synergize to ground the literal sense and evoke the figurative meaning simultaneously. Although Vision-Language Models (VLMs) are widely used in multimodal understanding and generation, their ability to understand puns has not been systematically studied due to a scarcity of rigorous benchmarks. To address this, we first propose a multimodal pun generation pipeline. We then introduce MULTIPUN, a dataset comprising diverse types of puns alongside adversarial non-pun distractors. Our evaluation reveals that most models struggle to distinguish genuine puns from these distractors. Moreover, we propose both prompt-level and model-level strategies to enhance pun comprehension, with an average improvement of 16.5% in F1 scores. Our findings provide valuable insights for developing future VLMs that master the subtleties of human-like humor via cross-modal reasoning.[1]

1 Introduction

Puns, also known as paronomasia in linguistics, are a form of wordplay that exploits multiple meanings of a term or similar-sounding words to create humor (Kao et al., 2016). Interpreting multimodal puns requires resolving a complex visual synthesis beyond simple image captioning: the image fuses a literal object with a metaphorical context, while the text forces a dual interpretation by unifying the object's visual identity with its behavioral state. Compared to other forms of humor like jokes (Dynel, 2009) or comedies (Stott, 2014), puns are structurally simpler and possess more precise linguistic definitions (Attardo, 2018). These qualities make them an ideal testbed for evaluating multimodal reasoning in Vision-Language Models (VLMs) (Team et al., 2023; Liu et al., 2023).

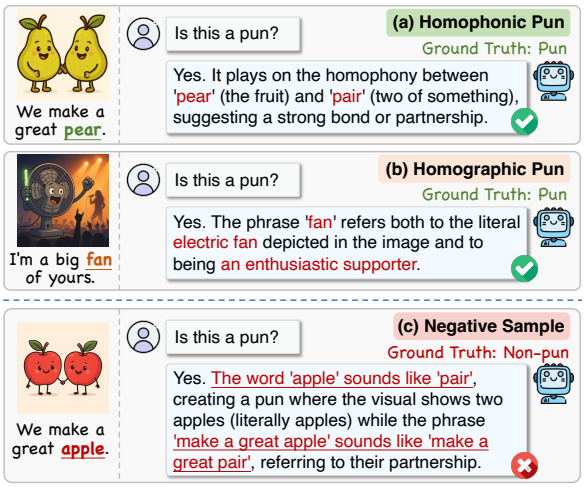


Figure 1: The recognition of multimodal pun examples from MULTIPUN. (a) A pun relying on phonetic similarity ("pear" and "pair"). (b) A pun based on word polysemy (double meaning of "fan"). (c) A negative sample illustrating a false positive case, where the model mistakenly interprets a non-pun as pun.

Consider the examples in Figure 1. Figure 1(a) depicts two pears (literal objects) holding hands like a romantic couple (figurative behavior). The caption "We make a great *pear*" exploits the sound similarity of "pear" to "pair". The humor emerges by connecting the visual intimacy (holding hands) with the auditory implication of being a romantic "pair". Similarly, Figure 1(b) also relies on the double meanings of the same word. The caption "I'm a big *fan* of yours" uses the polysemy of "fan" (cooling device vs. enthusiast). Notably, the image depicts an industrial fan (literal object), cheering with a glow stick (figurative behavior). Crucially, Figure 1(c) presents a negative example. The image still depicts intimate fruits (apples) and the sentence structure remains identical, but the phonetic connection to "pair" is broken. A robust model should recognize it as non-pun, whereas existing models might mistakenly interpret it as a pun.

*Corresponding Author.
[1]The code is available here.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9786–9805
July 2-7, 2026

Recent studies on pun detection (Zhou et al., 2020), explanation (Zangari et al., 2025), and generation (Xu et al., 2024b) face three critical limitations. (*i*) **Unimodal confinement.** Prior research predominantly targets textual puns (Miller et al., 2017), overlooking the cross-modal interplay where visual modality can cause ambiguity. (*ii*) **Deficiencies in multimodal benchmarks.** Existing multimodal efforts (Xu et al., 2025e) lack detailed pun categorization and non-puns as negative samples. This positive-only approach prevents us from knowing whether models truly understand the pun or just superficially link playful visual scenes with humor. (*iii*) **Conflation of preference and comprehension.** Previous evaluations (Xu et al., 2025e; Zangari et al., 2025) rely on single-sided querying (e.g., "Is this a pun?"), failing to separate true reasoning from the model's affirmative language bias (Zhuang et al., 2024). To address these gaps, we summarize three research questions (RQs):

- **RQ$_1$** – How effectively can VLMs recognize multimodal puns against non-puns?
- **RQ$_2$** – To what extent can VLMs explain puns?
- **RQ$_3$** – How can we enhance VLMs' understanding of puns?

To assess the abilities of VLMs in multimodal pun understanding, we propose MULTIPUN, a linguistically grounded multimodal benchmark with both pun and non-pun samples. To address **RQ$_1$**, we assess models' performance in pun detection, localization, and explanation tasks. We ask the same question in both direct and reverse forms, analyzing responses together to disentangle model preference from true pun understanding. For **RQ$_2$**, we employ both a fine-grained pun component verification and a coarse-grained explanation pairwise comparison to assess VLMs' comprehension of puns. To answer **RQ$_3$**, we propose prompt-level and model-level strategies to enhance VLMs' understanding of puns. Our contributions are as follows:

- We introduce the multimodal pun generation pipeline and propose MULTIPUN, a benchmark containing 445 puns and 890 non-puns to evaluate VLMs' understanding of puns.
- We design three pun detection, localization, and explanation tasks, and find that most VLMs superficially connect puns to common language patterns rather than truly understand them.
- We provide prompt-level method Pun-CoT and model-level method Pun-Tuning to enhance VLMs' understanding of puns, resulting in an average increase of 16.5% in F1 scores.

2 Related Work

Textual pun understanding. Puns are a linguistic art form that relies on phonological or semantic ambiguity. Early research primarily focuses on curating textual pun collections from web sources (Miller et al., 2017). The field gained momentum with SemEval-2017 Task 7 (Miller et al., 2017), which established benchmarks for pun detection and location. Recently, researchers have used Large Language Models (LLMs) to advance the detection (Zou and Lu, 2019; Zhou et al., 2020), explanation (Sun et al., 2022), and generation (Yu et al., 2020) of puns. However, these studies are confined to the textual modality, ignoring the cognitive complexity of multimodal ambiguity. Our work extends this by integrating the visual modality as an essential component for resolving ambiguity.

Multimodal humor and pun understanding. Understanding visual humor is crucial for assessing multimodal reasoning in VLMs. While there is growing interest in memes (Liu et al., 2024; Xu et al., 2025e), sarcasm (Wang et al., 2025), comics (Hu et al., 2024) and Chinese pun rebus (Zhang et al., 2025), research on multimodal puns is limited. Existing datasets lack fine-grained linguistic categorization, failing to distinguish between phonological and semantic strategies. More critically, most benchmarks evaluate models solely on puns without rigorous negative samples (Xu et al., 2025e; Chung et al., 2024). This makes it hard to determine whether models truly understand cross-modal alignment or merely generate hallucinatory humor. Our work bridges this gap with a benchmark including adversarial negatives.

3 MULTIPUN

MULTIPUN is a multimodal benchmark with 445 puns (homophonic and homographic, Section 3.1) and 890 non-pun distractors from two substitution strategies. Figure 2 shows the construction pipeline (Section 3.2). We introduce an evaluation suite for pun detection, localization, and explanation (Section 3.2.4) to assess VLM performance.

3.1 Preliminary

We focus on two main types of multimodal puns: *homophonic puns* and *homographic puns* (Miller et al., 2017). We formalize a multimodal pun instance as a tuple $\mathcal{P} = \langle w_p, w_a, S_p, S_a \rangle$ following Xu et al. (2024b). Here, w_p denotes the *pun word* in the image caption, and w_a represents the *alterna-*

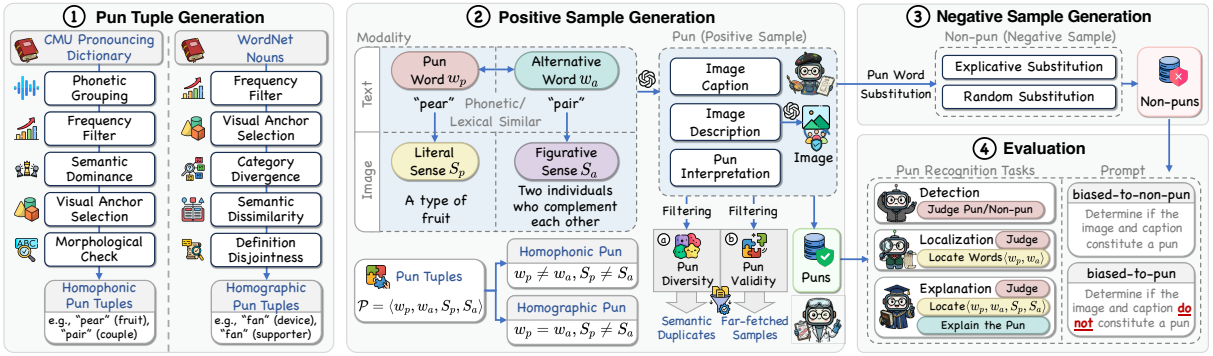


Figure 2: Overview of the MULTIPUN construction pipeline. Our pipeline generates both pun and non-pun samples.

tive word. Crucially, the image fuses two semantics: S_p is the literal concrete object corresponding to the meaning of w_p, and S_a is the figurative behavior or state associated with w_a.

- **Homophonic Pun:** This category uses the sound similarity between the w_p in the caption and w_a, which differ in spelling and meaning (Attardo, 2024). For instance, Figure 1(a) shows pears (S_p) holding hands like a couple (S_a), hinting at the "pair" (w_a), phonetically triggered by "We make a great *pear*" (w_p).
- **Homographic Pun:** This category exploits the dual meaning of homographs (Attardo, 2024), where w_p and w_a are spelled the same but have different meanings. For example, in Figure 1(b), "fan" serves as both the cooling device (w_p) and the enthusiast (w_a). The visual subject physically embodies the device (S_p) while functionally enacting the cheering behavior (S_a).

3.2 Dataset Construction

As shown in Figure 2, we construct the MULTIPUN benchmark using the following pipeline: pun tuple generation, positive sample generation, negative sample generation, and evaluation.

3.2.1 Pun Tuples Generation

Homophonic Puns. We retrieve word pairs w_p and w_a with identical pronunciation but distinct spellings with the following steps: (*i*) *Phonetic Grouping:* Use the CMU Pronouncing Dictionary (Carnegie Mellon University, 2015) to find word pairs with identical pronunciation. (*ii*) *Frequency Filter:* Apply a Zipf frequency threshold (> 3.0) to ensure words are commonly used. (*iii*) *Semantic Dominance:* Select the top-3 most frequent synsets in WordNet (Miller, 1992) to prioritize primary meanings. (*iv*) *Visual Anchor Selection:* Keep concrete nouns in visually depictable categories (*e.g.*, *noun.animal*, *noun.artifact*) so that w_p can be clearly illustrated. (*v*) *Morphological Check:* Use lemmatization checks to remove trivial variants, ensuring w_p and w_a are distinct lemmas.

Homographic Puns. We retrieve word w_p with two different meanings S_p and S_a with the following steps: (*i*) *Frequency Filter:* Select nouns with a Zipf frequency over 3.8 and choose their top-3 WordNet (Miller, 1992) synsets. (*ii*) *Visual Anchor Selection:* Keep candidates with a concrete sense (S_p) in visually depictable noun categories (*e.g.*, *noun.animal*, *noun.artifact*) so that w_p can be clearly illustrated. (*iii*) *Category Divergence:* Ensure S_a is in a different lexical file from S_p. (*iv*) *Semantic Dissimilarity:* Require low path similarity (< 0.1) and reject pairs where both in natural categories (*e.g.*, *noun.plant*, *noun.animal*), avoiding part-whole metonymy (e.g., apple tree vs. apple fruit). (*v*) *Definition Disjointness:* Remove synsets with definitions containing the target word, avoiding circular meanings (e.g., rejecting the "ball game" sense of *baseball* includes "ball").

After filtering, we obtain a set of pun tuples $\mathcal{P} = \langle w_p, w_a, S_p, S_a \rangle$ as seeds for sample generation.

3.2.2 Positive Sample Generation

Generation. Based on the pun tuples from the previous step, we employ `GPT-4o` to create multimodal samples. Specifically, the model is prompted to generate three distinct components for each tuple: (*i*) an image caption containing the pun word w_p, (*ii*) an image description detailed enough to guide the text-to-image generation, and (*iii*) a pun interpretation explaining the ambiguity. The image description is subsequently fed into the image generator `GPT-image-1` to create the visual scene. We manually verify image-description alignment and refine prompts to regenerate images when mismatches occur. The visual scene grounds the object's identity in the literal sense (S_p) while enacting its behavior in the figurative sense (S_a).

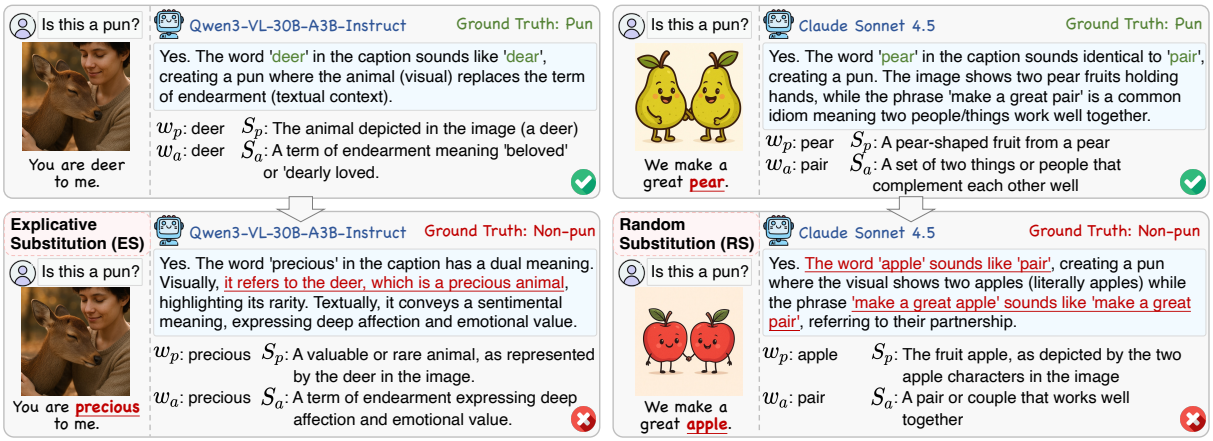


Figure 3: Examples of adversarial negative samples.

Filtering. We use the following filtering steps: (*i*) **Diversity Filtering:** Embedding-based filtering model `text-embedding-3-large` (OpenAI, 2024) removes highly similar samples to eliminate redundancy (see Appendix B.3 for the algorithm). (*ii*) **Validity Filtering:** We employ human-in-the-loop quality control to final verification (details in Appendix D). We discard *far-fetched samples* where the connection between the image and the caption is insufficient to form a recognizable pun.

3.2.3 Negative Sample Generation

To mitigate the positive-only bias in existing benchmarks and distinguish genuine comprehension from superficial overfitting, we construct adversarial negatives that disrupt the pun mechanism while maintaining surface coherence. We employ two primary disruption strategies:

- **Explicative Substitution (ES):** This variant resolves the linguistic ambiguity by replacing the pun word w_p with a direct description of the behavioral meaning S_a.
- **Random Substitution (RS):** This variant replaces w_p with a semantically unrelated entity (*e.g.*, "chair", "apple"), and creates a new image where a new entity performs the original action.

3.2.4 Evaluation Tasks

To systematically assess VLMs' capabilities in multimodal pun comprehension, we design a progressive evaluation suite comprising three tasks.

- **Detection** asks for binary judgment (pun or not) without definitions or guidance.
- **Localization** requires first judging and explicitly identifying words w_p and w_a.
- **Explanation** requires judging, providing a rationale that explains why it's a pun, and extracting the full tuple $\langle w_p, w_a, S_p, S_a \rangle$.

To separate true reasoning from the model's affirmative language bias (Zhuang et al., 2024; Xu et al., 2024b), we ask the same question twice in both direct and opposite form: (*i*) a *biased-to-pun* prompt that asks whether the given multimodal context is a pun, and (*ii*) a *biased-to-non-pun* prompt that asks whether the given multimodal context is not a pun.[2]

3.3 Experimental Setup

Models. We evaluate 11 representative VLMs on MULTIPUN across three tasks to evaluate their understanding of the puns, including GPT (OpenAI, 2025), Gemini (Comanici et al., 2025), Claude (Anthropic, 2025), Qwen (Bai et al., 2025), LLaVA (Liu et al., 2023) series.[3]

Metrics. We use two categories of metrics to evaluate model performance. (*i*) **Pun Recognition.** For all tasks (detection, localization, and explanation), we measure recognition accuracy through: (*a*) True Positive Rate (TPR) measures the proportion of correctly identified puns. (*b*) True Negative Rate (TNR) indicates the proportion of correctly identified non-puns. (*c*) F1-Score provides an overall performance assessment. (*d*) Variations (Δ) in TPR and TNR when the prompt leans towards non-pun compared to pun. (*e*) Cohen's Kappa (κ) (Cohen, 1960) measures agreement between two sets of biased recognitions. (*ii*) **Word Extraction and Explanation Quality.** For localization and explanation tasks, we use: (*a*) Mention ratio measures the proportion of ground-truth w_p and w_a in the extracted tuples that models correctly identify puns. (*b*) Win/tie/loss rates measure the judge's result by comparing model-generated explanations to ground-truth explanations.

[2] All prompts are provided in Appendix E.

[3] Detailed settings of VLMs are given in Appendix G.

Type	Model	Task	Homophonic Pun						Homographic Pun					
			TPR ↑	Δ_{TPR} ↓	TNR ↑	Δ_{TNR} ↓	F1 ↑	κ ↑	TPR ↑	Δ_{TPR} ↓	TNR ↑	Δ_{TNR} ↓	F1 ↑	κ ↑
Closed-Source VLMs	GPT-5.1	Detection	**0.933**	**-0.026**	0.379	+0.198	0.588	0.241	**0.956**	**-0.036**	0.243	+0.201	0.551	0.146
		Localization	0.887	-0.046	0.768	+0.072	0.754	0.601	0.876	-0.108	0.695	+0.141	0.705	0.508
		Explanation	0.794	-0.062	**0.910**	**+0.059**	**0.804**	**0.708**	0.757	-0.143	**0.878**	**+0.060**	**0.757**	**0.637**
	GPT-4o	Detection	**0.933**	**0.000**	0.332	+0.144	0.571	0.202	**0.956**	**-0.004**	0.211	+0.122	0.541	0.121
		Localization	0.923	-0.015	0.582	+0.088	0.669	0.425	0.888	-0.028	0.480	+0.120	0.607	0.299
		Explanation	0.840	-0.026	**0.786**	**+0.072**	**0.741**	**0.587**	0.873	-0.064	**0.659**	**+0.096**	**0.683**	**0.467**
	Gemini-3-Pro	Detection	**0.979**	-0.015	0.268	+0.142	0.569	0.181	0.984	-0.008	0.209	+0.135	0.552	0.139
		Localization	0.974	+0.005	0.250	+0.039	0.561	0.163	**0.996**	-0.016	0.221	+0.064	0.561	0.158
		Explanation	0.969	**-0.005**	**0.686**	**+0.023**	**0.746**	**0.579**	0.980	**-0.004**	**0.625**	**+0.008**	**0.718**	**0.520**
	Claude Sonnet-4.5	Detection	0.974	**-0.005**	0.134	+0.134	0.526	0.076	0.992	-0.012	0.102	+0.110	0.524	0.065
		Localization	**0.990**	+0.010	0.072	+0.072	0.515	0.042	**0.996**	**0.000**	0.046	**+0.052**	0.510	0.028
		Explanation	0.969	-0.010	**0.353**	**+0.070**	**0.594**	**0.245**	0.984	+0.004	**0.235**	+0.127	**0.560**	**0.159**
Open-Source VLMs	Qwen3-VL 8B-Instruct	Detection	**0.923**	**-0.160**	0.193	+0.338	0.522	0.084	**0.968**	-0.263	0.147	+0.351	**0.527**	0.082
		Localization	0.799	-0.222	0.487	**+0.291**	**0.566**	0.237	0.681	-0.359	0.490	+0.307	0.504	**0.146**
		Explanation	0.418	-0.268	**0.881**	+0.111	0.505	**0.329**	0.207	**-0.191**	**0.904**	**+0.084**	0.296	0.131
	Qwen3-VL 30B-Instruct	Detection	**0.990**	**-0.031**	0.018	**+0.201**	0.501	0.005	**1.000**	**-0.048**	0.028	**+0.506**	0.507	0.019
		Localization	0.985	-0.129	0.067	+0.343	0.511	0.035	0.996	-0.155	0.052	+0.275	**0.512**	0.033
		Explanation	0.943	-0.273	**0.209**	+0.469	**0.535**	**0.110**	0.944	-0.267	**0.125**	+0.490	0.511	**0.050**
	LLaVA-v1.6 Vicuna-13B	Detection	**0.969**	-0.923	0.023	+0.933	0.494	-0.005	**0.980**	-0.944	0.024	+0.950	0.498	0.003
		Localization	0.866	-0.392	0.072	+0.356	**0.465**	-0.043	0.928	-0.434	0.102	+0.359	0.498	0.021
		Explanation	0.031	**-0.015**	**0.972**	**+0.023**	0.057	**0.004**	0.028	**-0.012**	**0.966**	**+0.026**	0.051	-0.007
	Llama-4 Scout-17B	Detection	0.912	-0.149	0.423	+0.381	0.595	0.265	**0.912**	-0.275	0.341	+0.408	**0.565**	0.193
		Localization	**0.933**	**0.000**	0.407	**-0.064**	0.598	0.266	0.837	**+0.044**	0.355	**-0.112**	0.535	0.147
		Explanation	0.799	-0.072	**0.624**	+0.142	**0.626**	**0.372**	0.749	-0.100	**0.494**	+0.145	0.543	**0.204**
Open-Source Reasoning-based VLMs	GLM-4.1V 9B-Thinking	Detection	**0.969**	-0.206	0.124	+0.487	0.521	0.050	**0.956**	-0.247	0.092	+0.484	0.507	0.026
		Localization	0.887	-0.129	0.567	+0.245	**0.644**	0.367	0.841	-0.175	**0.550**	+0.052	0.613	0.411
		Explanation	0.835	**-0.015**	**0.629**	**+0.062**	0.648	**0.376**	0.940	**-0.044**	0.550	**+0.052**	**0.662**	**0.411**
	Qwen3-VL 8B-Thinking	Detection	**0.990**	-0.211	0.054	+0.593	0.510	0.023	0.980	-0.215	0.048	+0.554	0.505	0.016
		Localization	0.985	**-0.031**	0.106	+0.263	0.522	0.090	**0.992**	-0.052	0.118	+0.309	0.528	0.117
		Explanation	0.943	-0.077	**0.387**	**+0.119**	**0.595**	**0.325**	0.960	**-0.044**	**0.367**	**+0.197**	**0.595**	**0.343**
	Qwen3-VL 30B-A3B Thinking	Detection	0.990	-0.149	0.106	+0.448	0.524	0.049	0.992	-0.112	0.078	+0.390	0.517	0.036
		Localization	**1.000**	**0.000**	0.165	+0.227	**0.545**	0.145	**1.000**	**-0.008**	0.151	+0.319	**0.541**	0.135
		Explanation	0.985	-0.026	**0.399**	**+0.155**	0.618	**0.273**	**1.000**	-0.020	**0.414**	**+0.163**	0.631	**0.298**

Table 1: Results of pun recognition in detection, localization, and explanation tasks. Metrics (TPR, TNR, F1, κ) are evaluated under the *biased-to-pun* prompt. Δ measures variations when prompt bias shifts from pun to non-pun. Darker colors indicate better performance. The best results (smallest variations or highest scores) are **bolded**.

4 Results and Analysis

4.1 How Effectively Can VLMs Recognize Multimodal Puns Against Non-puns?

Table 1 shows the results of VLMs on pun recognition tasks, including detection, localization, and explanation. We have the following observations.

VLMs often classify non-pun samples as puns. Most models achieve high TPR in pun recognition but struggle with low TNR, particularly in detection and localization tasks. For example, Qwen3-VL-30B-A3B-Instruct identifies almost every input as a pun, achieving a near-perfect TPR of 0.990, but its TNR drops to 0.018 in detecting homophonic puns. Similarly, closed-source models such as GPT-5.1, GPT-4o, Gemini-3-Pro, and Claude-Sonnet-4.5 exhibit TNR scores mostly below 0.38 in detection tasks. Even in the explanation task, although GPT-5.1 and GPT-4o improve their TNR to above 0.75, Gemini-3-Pro and Claude-Sonnet-4.5 remain lower at 0.686 and 0.353, respectively. This imbalance results in poor Cohen's Kappa scores ($\kappa < 0.4$), indicating that models frequently misclassify non-puns as puns rather than a genuine understanding of pun mechanisms.

Open-source models exhibit greater prompt-induced bias in pun recognition. We measure prompt-induced bias (i.e., where model decisions are influenced by prompt phrasing rather than content) through the variations in Δ_{TPR} and Δ_{TNR} when switching from *biased-to-pun* prompt to *biased-to-non-pun* prompt. These variations reveal that many VLMs, particularly open-source ones, are easily influenced by the way questions are asked and lack robust internal reasoning for pun recognition. Notably, LLaVA-V1.6-Vicuna-13B exhibits a dramatic Δ_{TPR} of -0.923, suggesting that its decisions are primarily driven by prompt question format rather than the genuine multimodal understanding. In contrast, closed-source models such as GPT-4o and Gemini-3-Pro maintain consistency across prompt variations, with low absolute values of Δ_{TPR} and Δ_{TNR}, demonstrating superior robustness in reasoning.

Explanation tasks improve non-pun rejection but slightly compromise pun detection. Models perform better at rejecting non-puns when tasked with explaining the pun rather than simply detecting or localizing it. A clear upward trend in TNR is observed across most models during explanation

tasks. For instance, the TNR of GPT-5.1 for homophonic puns increases sharply from 0.379 in detection to 0.910 in explanation. This suggests that requiring models to explicitly identify pun components and explain their reasoning helps ground their judgments in evidence, effectively reducing hallucinated false positives. However, this stricter verification process consistently leads to a drop in TPR, indicating that models sometimes discard valid puns when they fail to correctly explain the underlying punning mechanism.

Closed-source models outperform open-source counterparts in pun recognition. Closed-source models such as GPT-5.1, GPT-4o, and Gemini-3-Pro consistently demonstrate superior performance across detection, localization, and explanation tasks, achieving high F1 scores. In contrast, open-source models often struggle to recognize puns accurately, exhibiting lower F1 scores and more pronounced performance gaps between TPR and TNR. A notable example is LLaVA-V1.6-Vicuna-13B, whose performance collapses in the explanation task, with the F1 score dropping to approximately 0.058. This failure suggests deficiencies in pun comprehension, likely due to limited training data or architectural constraints.

Reasoning-based models do not guarantee improved pun recognition. Comparing standard models with their reasoning-based "Thinking" variants reveals mixed results based on model scale. For smaller models such as Qwen3-VL-8B-Instruct, introducing reasoning processes worsens performance, with TNR dropping from 0.193 to 0.054, indicating hallucination in pun recognition. Conversely, larger models such as Qwen3-VL-30B-A3B-Instruct benefit from reasoning, improving both pun detection and non-pun rejection. Specifically, its TPR increases from 0.943 to 0.985, while its TNR improves from 0.209 to 0.399.

Error analysis of negative samples. We categorize false positives into four distinct hallucination patterns, covering the lexical, phonological, semantic, and visual levels. (*i*) *Pun word hallucination.* VLMs prioritize idiomatic priors over visual evidence. The model ignores the actual word written in the text and shown in the image (e.g., "lamp") and mistakenly imagines the common word that usually fits the idiom (e.g., "fan"). (*ii*) *Phonetic hallucination.* To force a connection, the model wrongly claims that two words sound alike, even when they sound completely different (e.g., claiming "banana" sounds like "soul"). (*iii*) *Semantic hallucination.* Models correctly identify the alternative word w_a but invent a meaning that does not exist. For instance, it tries to force the meaning of "pair" onto the word "banana", even though they are not related. (*iv*) *Visual object hallucination.* Misled by the text, the model imagines seeing objects that are not actually in the image. For example, reading about a "date" makes the model say it sees the fruit "date" in the image, when it is actually an apple. We provided detailed case studies in Appendix L.1.

Model	Homophonic Pun				Homographic Pun			
	Localization		Explanation		Localization		Explanation	
	w_p	w_a	w_p	w_a	w_p	w_a	w_p	w_a
Closed-Source VLMs								
GPT-5.1	**98.8**	**87.8**	**100.0**	**89.0**	97.7	97.7	97.9	97.9
GPT-4o	96.1	84.9	92.6	75.5	97.3	97.3	97.3	97.3
Gemini-3-Pro	97.4	86.8	97.9	88.8	**98.8**	**98.8**	**98.8**	**98.8**
Claude-Sonnet-4.5	93.2	82.8	94.7	81.9	96.8	96.8	96.8	96.8
Open-Source VLMs								
Qwen3-VL-8B-Instruct	**92.3**	73.5	**90.1**	40.7	**96.5**	**96.5**	**96.2**	**96.2**
Qwen3-VL-30B-Instruct	84.3	75.4	82.5	59.0	96.0	96.0	94.5	94.5
LLaVA-v1.6-Vicuna-13B	79.2	38.7	50.0	**83.3**	91.0	91.0	42.9	42.9
Llama-4-Scout-17B	91.7	**84.0**	81.9	29.7	91.9	91.9	93.6	93.6
Open-Source Reasoning-Based VLMs								
GLM-4.1V-9B-Thinking	96.5	80.8	86.4	59.3	98.1	98.1	95.8	95.8
Qwen3-VL-8B-Thinking	94.8	81.7	**95.6**	68.3	96.8	96.8	97.9	97.9
Qwen3-VL-30B-Thinking	**96.9**	**90.7**	94.2	**81.2**	**100.0**	**100.0**	**98.4**	**98.4**

Table 2: Pun component verification for pun localization and explanation. We represent the average mention ratio of the pun words w_p and alternative words w_a.

4.2 To What Extent Can VLMs Explain Puns?

Beyond recognition, we explore pun understanding by: (*i*) **pun component verification** check how accurately pun words w_p and alternatives w_a are identified, and (*ii*) **explanation pairwise comparison** assesses the quality of the pun explanation.

4.2.1 Pun Component Verification

We calculate mention ratios for verifying the pun word w_p and the alternative word w_a. As shown in Table 2, we have the following observations.

VLMs accurately identify the pun word w_p. The mention ratio of w_p remains consistently high across most models for both homophonic and homographic puns. For example, closed-source models such as GPT-5.1 and reasoning-based models such as Qwen3-VL-30B-A3B-Thinking achieve mention ratios over 94%. Even smaller open-source models perform well (e.g., Qwen3-VL-8B-Instruct achieves 92.3% in homophonic pun localization). This high accuracy is due to w_p appearing directly in the caption, making it easy to identify.

Identifying the alternative word w_a is the bottleneck for homophonic puns. Comparing the mention ratio of w_p, we observe a decrease in the mention ratio of w_a. For instance, while Qwen3-

VL-8B-Instruct achieves a 90.1% mention ratio for w_p in the explanation task, its performance on w_a drops drastically to 40.7%. This challenge arises because w_a in homophonic puns does not directly appear in the text but depends on semantic inference and similar pronunciation to w_p.

Reasoning improves pun component identification. Compared to instruction-based models, reasoning-based models show a superior ability to identify both w_p and w_a through explicit thinking steps. For example, for homophonic puns, Qwen3-VL-30B-A3B-Thinking increases the mention ratio of w_a from 59.0% (Instruct version) to 81.2% in the explanation task. It also achieves highest mention ratio of both w_p and w_a on homographic puns (100% in the localization task and 98.4% in the explanation task). This suggests that the extended reasoning phase helps the model to explore phonetic or semantic connections more effectively.

4.2.2 Explanation Pairwise Comparison

While pun component verification measures recall on pun words, it does not assess the quality of the pun explanation. To evaluate this, we conduct a pairwise comparison where an advanced LLM judge compares the VLM-generated explanation to the ground-truth explanation from the MULTIPUN dataset. The judge classifies the comparison as a *Win* (VLM is better), *Tie* (Comparable), or *Loss* (Ground truth is better). As shown in Figure 4, we have the following observations.

Ground-truth explanations generally outperform VLM-generated explanations. Across all evaluated models, the loss rate is exceptionally higher than the win rate. For instance, even the advanced GPT-5.1 loses to the ground truth in about 90% of cases. This suggests that while models can identify pun components, recognizing them does not necessarily mean they understand the underlying logic of the pun effectively.

VLMs explain homographic puns better than homophonic puns. We observe a consistent trend where models achieve higher win rates on homographic puns compared to homophonic ones. This aligns with the findings from the pun component verification and findings by Xu et al. (2024b), where VLMs are better at explaining a word's multiple meanings than at articulating phonetic bridges by finding an alternative word w_a. Thus, alternative words do not affect pun recognition but are crucial for explaining puns more effectively.

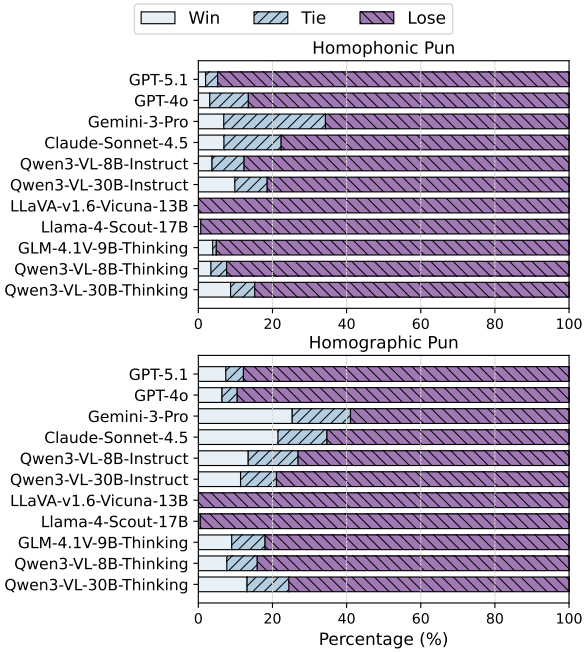


Figure 4: Pairwise comparison for pun explanations.

4.2.3 Error Analysis in Pun Explanation

VLMs exhibit distinct error patterns in explaining puns. We categorize the primary errors as follows: (*i*) *Detection Failure.* VLMs identify pun as non-pun, failing to recognize the double meaning. (*ii*) *Pun Word Error.* VLMs detect the pun but fails to identify the pun word w_p. (*iii*) *Alternative Word Error.* VLMs identify the correct pun word w_p but fails to retrieve the intended alternative word w_a. (*iv*) *Cross-modal Integration Error.* VLMs identify both visual and textual content but explain them separately, failing to integrate them with the proper linguistic mechanism. We provide cases for each error type in Appendix L.2. We believe that addressing these errors is pivotal to advancing VLMs' capability to recognize and understand puns.

4.3 How Can We Enhance VLMs' Understanding of Puns?

4.3.1 Pun-CoT

To mitigate the hallucinations identified in our error analysis, we propose **Pun-CoT**. Pun-CoT enforces the following process (see Appendix F for the complete prompt): (*i*) Visual Grounding. The model verifies the literal visual content to prevent visual object hallucinations. (*ii*) Lexical Anchoring. The model extracts exact keywords from the caption as w_p, thereby preventing hallucinated words not present in the caption. (*iii*) Cross-Modal Verification. The model checks if the visual content links to the text via a valid phonetic (for homophonic

Model	Method	Homophonic Pun			Homographic Pun		
		TPR ↑	TNR ↑	F1 ↑	TPR ↑	TNR ↑	F1 ↑
GPT-5.1	Vanilla	0.794	0.910	0.804	0.757	0.878	0.757
	Pun-CoT	**0.840**	**0.915**	**0.836**	**0.813**	**0.894**	**0.803**
GPT-4o	Vanilla	0.840	0.786	0.741	0.873	0.659	0.683
	Pun-CoT	**0.876**	**0.835**	**0.794**	**0.888**	**0.727**	**0.730**
Gemini-3 Pro	Vanilla	**0.969**	0.686	0.746	**0.980**	0.625	0.718
	Pun-CoT	0.959	**0.719**	**0.761**	0.976	**0.655**	**0.732**
Claude Sonnet-4.5	Vanilla	**0.969**	0.353	0.594	**0.984**	0.235	0.560
	Pun-CoT	0.948	**0.495**	**0.641**	0.972	**0.480**	**0.646**
Qwen3-VL 8B-Instruct	Vanilla	0.418	**0.881**	0.505	0.207	**0.904**	0.296
	Pun-CoT	**0.799**	0.495	**0.569**	**0.685**	0.490	**0.507**
Qwen3-VL 30B-Instruct	Vanilla	0.943	0.209	0.535	0.944	0.125	0.511
	Pun-CoT	**0.974**	**0.214**	**0.549**	**0.992**	**0.139**	**0.534**
LLaVA-v1.6 Vicuna-13B	Vanilla	0.031	**0.972**	0.057	0.028	**0.966**	0.051
	Pun-CoT	**0.979**	0.036	**0.501**	**0.984**	0.102	**0.521**
Llama-4 Scout-17B	Vanilla	0.799	0.624	0.626	0.749	0.494	0.543
	Pun-CoT	**0.866**	**0.629**	**0.664**	**0.757**	**0.522**	**0.558**
GLM-4.1V 9B-Thinking	Vanilla	0.835	**0.629**	0.648	**0.940**	0.550	0.662
	Pun-CoT	**0.948**	0.608	**0.694**	0.916	**0.757**	**0.763**
Qwen3-VL 8B-Thinking	Vanilla	0.943	0.387	0.595	**0.960**	0.367	0.595
	Pun-CoT	**0.979**	**0.776**	**0.807**	0.920	**0.797**	**0.791**
Qwen3-VL 30B-Thinking	Vanilla	**0.985**	0.399	0.618	**1.000**	0.414	0.631
	Pun-CoT	0.887	**0.567**	**0.644**	0.976	**0.480**	**0.647**

Table 3: Comparison of pun recognition with and without Pun-CoT across VLMs under the explanation task.

puns) or semantic (for homographic puns) bridge, rejecting weak or fabricated associations.

Results. Table 3 demonstrates the efficacy of Pun-CoT in balancing pun sensitivity with hallucination mitigation. Pun-CoT yields consistent improvements in F1 scores across diverse architectures, primarily driven by a substantial boost in TNR. Notably, for models prone to over-interpretation such as Qwen3-VL-8B-Thinking and Claude-Sonnet-4.5, Pun-CoT significantly enhances their ability to reject non-puns (e.g., doubling Qwen3-VL-8B-Thinking's TNR from 0.387 to 0.776) while maintaining competitive TPR. This confirms that explicitly grounding reasoning in verified visual and lexical evidence effectively filters out forced associations for robust comprehension.

4.3.2 Pun-Tuning

Motivation. As illustrated in Section 4.1, current VLMs exhibit three challenges in pun understanding, including: (*i*) Over-interpretation, where models misclassify non-puns as puns due to a reliance on superficial pun pattern matching rather than a robust understanding; (*ii*) Imprecise explanations, revealing deficits in understanding fine-grained phonetic and orthographic similarity; and (*iii*) Prompt sensitivity, driven by alignment-induced *sycophancy*, where models prioritize agreeableness with the user's premise over factual accuracy.

To address these, our data construction includes: (*i*) We incorporate non-pun samples to suppress hallucinations. (*ii*) We utilize pun samples with high-quality responses to enhance recall and explanatory depth. (*iii*) We employ both *biased-to-pun* prompt and *biased-to-non-pun* prompt. This improves robustness against prompt-induced bias. We use the constructed dataset to fine-tune VLMs. The implementation details are provided in Appendix I.

	Model	Method	TPR ↑	Δ_{TPR} ↓	TNR ↑	Δ_{TNR} ↓	F1 ↑
Homophonic Pun	Qwen3-VL 8B-Instruct	Vanilla	0.418	-0.268	0.881	+0.111	0.505
		Pun-Tuning	**0.577**	**-0.155**	**0.938**	**+0.098**	**0.679**
	Qwen3-VL 30B-Instruct	Vanilla	**0.943**	-0.273	0.209	+0.469	0.535
		Pun-Tuning	0.732	**-0.062**	**0.948**	**+0.196**	**0.798**
	LLaVA-v1.6 Vicuna-13B	Vanilla	0.031	**-0.015**	0.972	**+0.023**	0.057
		Pun-Tuning	**0.495**	-0.103	**0.974**	+0.098	**0.640**
	Llama-4 Scout-17B	Vanilla	**0.799**	**-0.072**	0.624	+0.142	0.626
		Pun-Tuning	0.722	-0.093	**0.918**	**+0.119**	**0.765**
Homographic Pun	Qwen3-VL 8B-Instruct	Vanilla	0.207	-0.191	0.904	**+0.084**	0.296
		Pun-Tuning	**0.556**	**-0.159**	**0.948**	+0.119	**0.670**
	Qwen3-VL 30B-Instruct	Vanilla	**0.944**	**-0.267**	0.125	+0.490	0.511
		Pun-Tuning	0.722	-0.548	**0.960**	**+0.222**	**0.802**
	LLaVA-v1.6 Vicuna-13B	Vanilla	0.028	**-0.012**	0.966	**+0.026**	0.051
		Pun-Tuning	**0.460**	-0.238	**0.984**	+0.365	**0.617**
	Llama-4 Scout-17B	Vanilla	**0.749**	**-0.100**	0.494	+0.145	0.543
		Pun-Tuning	0.706	-0.105	**0.921**	**+0.103**	**0.757**

Table 4: Comparison of pun recognition with and without Pun-Tuning on VLMs under the explanation task.

Results. Table 4 reveals two key findings: (*i*) Fine-tuning VLMs on non-pun samples enhances the non-pun recognition capabilities of fine-tuned models, as evidenced by improvements in the TNR and F1 scores. (*ii*) Fine-tuning VLMs on pun samples enhances robustness against prompt-induced bias, with a decrease in the absolute values of Δ_{TPR} and Δ_{TNR}. Additionally, we conduct the explanation pairwise comparison in the same way as Section 4.2.2. As shown in Appendix H, we observe that fine-tuning VLMs on pun samples enhances models' understanding of puns with a higher win rate compared to the model before fine-tuning.

5 Conclusion

In this paper, we propose MULTIPUN, a benchmark for evaluating VLMs' understanding of multimodal puns. Our benchmark includes both puns and non-puns. Through systematic evaluation of 11 VLMs across three pun recognition tasks–pun detection, localization, and explanation, we observe significant biases in pun recognition and deficits in understanding fine-grained phonetic and orthographic similarity of puns. To enhance pun comprehension, we propose a prompt-level method, Pun-CoT, and a model-level method, Pun-Tuning. Our experiments show that both strategies improve VLMs' understanding of puns while preventing non-puns from being misidentified as puns. We hope that our findings and the MULTIPUN benchmark will contribute to the advancement of multimodal pun understanding and encourage the development of more resilient and reliable VLM capabilities.

Limitations

While MULTIPUN represents a significant step toward rigorous evaluation of multimodal pun comprehension, several limitations exist. First, our benchmark focuses exclusively on English puns. Since puns are deeply rooted in language-specific phonology, extending the dataset to other languages would test models' ability to handle multilingual settings. Second, our evaluation includes 11 representative VLMs, but newer models may exhibit different behaviors. Additionally, our fine-tuning experiments are limited to three open-source models due to computational constraints. Expanding fine-tuning experiments to more models and larger scales would strengthen our conclusions. Finally, while our adversarial negatives effectively disrupt pun mechanisms, they may not cover all possible failure modes. Future work could design more diverse types of negative samples to probe model robustness comprehensively.

Ethics Considerations

All data in MULTIPUN is generated using publicly available text-to-image models and LLMs, strictly following their intended purposes and respective licenses. No personally identifiable information or real individuals are depicted in the images. While advancements in pun understanding can enhance human-AI interaction, we acknowledge the dual-use nature of such technologies, where AI systems capable of linguistic manipulation could be weaponized for social engineering or propaganda. We advocate for transparent reporting of model capabilities and limitations, as well as ongoing dialogue between researchers, ethicists, and policymakers to ensure responsible development.

Acknowledgments

This work was partly supported by the Science Challenge Project under No. TZ2025005, NSFC under No. U2441239, 62402418 and U24A20336, the "Pioneer and Leading Goose" R&D Program of Zhejiang under No. 2026C02A1233 and 2025C02034, the Key R&D Program of Ningbo under No. 2024Z115, the Ningbo Yongjiang Talent Project, the China Postdoctoral Science Foundation under No. 2024M762829 and 2025M781522, and Zhejiang Key Laboratory of Decision Intelligence under No. 2025E10006.

References

Hengyu An, Minxi Li, Jinghuai Zhang, Naen Xu, Chunyi Zhou, Changjiang Li, Xiaogang Xu, Tianyu Du, and Shouling Ji. 2026. Aciarena: Toward unified evaluation for agent cascading injection. *Preprint*, arXiv:2604.07775.

Hengyu An, Jinghuai Zhang, Tianyu Du, Chunyi Zhou, Qingming Li, Tao Lin, and Shouling Ji. 2025. IPI-Guard: A novel tool dependency graph-based defense against indirect prompt injection in LLM agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1039, Suzhou, China. Association for Computational Linguistics.

Anthropic. 2025. Claude sonnet 4. https://www.anthropic.com/claude/sonnet.

Salvatore Attardo. 2018. Universals in puns and humorous wordplay. *Cultures and traditions of wordplay and wordplay research*, pages 89–110.

Salvatore Attardo. 2024. *Linguistic theories of humor*, volume 1. Walter de Gruyter GmbH & Co KG.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Carnegie Mellon University. 2015. The CMU pronouncing dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

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.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Marta Dynel. 2009. Beyond a joke: Types of conversational humour. *Language and linguistics compass*, 3(5):1284–1299.

Zhe Hu, Tuo Liang, Jing Li, Yiren Lu, Yunlai Zhou, Yiran Qiao, Jing Ma, and Yu Yin. 2024. Cracking the code of juxtaposition: Can ai models understand the humorous contradictions. *Advances in Neural Information Processing Systems*, 37:47166–47188.

International Organization for Standardization. 2024a. ISO/TS 5777:2024 Health informatics — The architecture of internet healthcare service network. Technical Specification ISO/TS 5777:2024, International Organization for Standardization.

International Organization for Standardization. 2024b. ISO/TS 5788:2024 Health informatics — Internet healthcare service pattern. Technical Specification ISO/TS 5788:2024, International Organization for Standardization.

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

Guangchen Lan, Sipeng Zhang, Tianle Wang, Yuwei Zhang, Daoan Zhang, Xinpeng Wei, Xiaoman Pan, Hongming Zhang, Dong-Jun Han, and Christopher G Brinton. 2025. Mappo: Maximum a posteriori preference optimization with prior knowledge. *arXiv preprint arXiv:2507.21183*.

Zixu Li, Zhiwei Chen, Haokun Wen, Zhiheng Fu, Yupeng Hu, and Weili Guan. 2025. Encoder: Entity mining and modification relation binding for composed image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5101–5109.

Zixu Li, Yupeng Hu, Zhiwei Chen, Qinlei Huang, Guozhi Qiu, Zhiheng Fu, and Meng Liu. 2026a. Retrack: Evidence-driven dual-stream directional anchor calibration network for composed video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 23373–23381.

Zixu Li, Yupeng Hu, Zhiwei Chen, Shiqi Zhang, Qinlei Huang, Zhiheng Fu, and Yinwei Wei. 2026b. Habit: Chrono-synergia robust progressive learning framework for composed image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 6762–6770.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Xuan Liu, Haoyang Shang, and Haojian Jin. 2025. Cobra: Programming cognitive bias in social agents using classic social science experiments. *arXiv preprint arXiv:2509.13588*.

Xuan Liu, Haoyang Shang, Zizhang Liu, Xinyan Liu, Yunze Xiao, Yiwen Tu, and Haojian Jin. 2026. Humanstudy-bench: Towards ai agent design for participant simulation. *arXiv preprint arXiv:2602.00685*.

Ziqiang Liu, Feiteng Fang, Xi Feng, Xeron Du, Chenhao Zhang, Noah Wang, Qixuan Zhao, Liyang Fan, CHENGGUANG GAN, Hongquan Lin, and 1 others. 2024. Ii-bench: An image implication understanding benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 37:46378–46480.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

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.

OpenAI. 2024. New embedding models and api updates. Blog post.

OpenAI. 2025. Gpt-5 is here. https://openai.com/gpt-5/.

Andrew Stott. 2014. *Comedy*. Routledge.

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.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Xinyu Wang, Yue Zhang, and Liqiang Jing. 2025. Can large vision-language models understand multimodal sarcasm? In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 5340–5345.

Yebo Wu, Jingguang Li, Zhijiang Guo, and Li Li. 2025a. Elastic mixture of rank-wise experts for knowledge reuse in federated fine-tuning. *arXiv preprint arXiv:2512.00902*.

Yebo Wu, Jingguang Li, Zhijiang Guo, and Li Li. 2026a. Developmental federated tuning: A cognitive-inspired paradigm for efficient LLM adaptation. In *The Fourteenth International Conference on Learning Representations*.

Yebo Wu, Jingguang Li, Chunlin Tian, Zhijiang Guo, and Li Li. 2025b. Memory-efficient federated fine-tuning of large language models via layer pruning. *arXiv preprint arXiv:2508.17209*.

Yebo Wu, Li Li, Chunlin Tian, Tao Chang, Chi Lin, Cong Wang, and Cheng-Zhong Xu. 2024. Heterogeneity-aware memory efficient federated learning via progressive layer freezing. In *2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, pages 1–10. IEEE.

Yebo Wu, Li Li, and Cheng-zhong Xu. 2025c. Breaking the memory wall for heterogeneous federated learning via progressive training. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1623–1632.

Yebo Wu, Feng Liu, Ziwei Xie, Zhiyuan Liu, Changwang Zhang, Jun Wang, and Li Li. 2026b. Tsembed: Unlocking task scaling in universal multimodal embeddings. *arXiv preprint arXiv:2603.04772*.

Naen Xu, Hengyu An, Shuo Shi, Jinghuai Zhang, Chunyi Zhou, Changjiang Li, Tianyu Du, Zhihui Fu, Jun Wang, and Shouling Ji. 2026a. When agents "misremember" collectively: Exploring the mandela effect in LLM-based multi-agent systems. In *The Fourteenth International Conference on Learning Representations*.

Naen Xu, Changjiang Li, Tianyu Du, Minxi Li, Wenjie Luo, Jiacheng Liang, Yuyuan Li, Xuhong Zhang, Meng Han, Jianwei Yin, and 1 others. 2024a. Copyrightmeter: Revisiting copyright protection in text-to-image models. *arXiv preprint arXiv:2411.13144*.

Naen Xu, Jinghuai Zhang, Changjiang Li, Hengyu An, Chunyi Zhou, Jun Wang, Boyu Xu, Yuyuan Li, Tianyu Du, and Shouling Ji. 2026b. Bridging the copyright gap: Do large vision-language models recognize and respect copyrighted content? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35949–35957.

Naen Xu, Jinghuai Zhang, Changjiang Li, Zhi Chen, Chunyi Zhou, Qingming Li, Tianyu Du, and Shouling Ji. 2025a. Videoeraser: Concept erasure in text-to-video diffusion models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5965–5994.

Zhenhua Xu, Dongsheng Chen, Shuo Wang, Jian Li, Chengjie Wang, Meng Han, and Yabiao Wang. 2026c. Adamarp: An adaptive multi-agent interaction framework for general immersive role-playing. *Preprint*, arXiv:2601.11007.

Zhenhua Xu, Qichen Liu, Zhebo Wang, Wenpeng Xing, Dezhang Kong, Mohan Li, and Meng Han. 2025b. Fingerprint vector: Enabling scalable and efficient model fingerprint transfer via vector addition. *Preprint*, arXiv:2409.08846.

Zhenhua Xu, Xubin Yue, Zhebo Wang, Qichen Liu, Xixiang Zhao, Jingxuan Zhang, Wenjun Zeng, Wengpeng Xing, Dezhang Kong, Changting Lin, and Meng Han. 2025c. Copyright protection for large language models: A survey of methods, challenges, and trends. *Preprint*, arXiv:2508.11548.

Zhenhua Xu, Xixiang Zhao, Xubin Yue, Shengwei Tian, Changting Lin, and Meng Han. 2025d. CTCC: A Robust and Stealthy Fingerprinting Framework for Large Language Models via Cross-Turn Contextual Correlation Backdoor. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6978–7000, Suzhou, China. Association for Computational Linguistics.

Zhijun Xu, Siyu Yuan, Lingjie Chen, and Deqing Yang. 2024b. "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.

Zhijun Xu, Siyu Yuan, Yiqiao Zhang, Jingyu Sun, Tong Zheng, and Deqing Yang. 2025e. PunMemeCN: A benchmark to explore vision-language models' understanding of Chinese pun memes. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18705–18721, Suzhou, China. 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.

Alessandro Zangari, Matteo Marcuzzo, Andrea Albarelli, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2025. Pun unintended: LLMs and the illusion of humor understanding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27924–27959, Suzhou, China. Association for Computational Linguistics.

Tuo Zhang, Tiantian Feng, Yibin Ni, Mengqin Cao, Ruying Liu, Kiana Avestimehr, Katharine Butler, Yanjun Weng, Mi Zhang, Shrikanth Narayanan, and 1 others. 2025. Creating a lens of chinese culture: A multimodal dataset for chinese pun rebus art understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22473–22487.

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.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 2024. Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels. In *Proceedings of the 2024 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies (volume 2: short papers)*, pages 358–370.

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 Dataset Statistics

As shown in Table 5, MULTIPUN comprises a total of 445 positive pun instances: 194 Homophonic Puns and 251 Homographic Puns. For each positive instance, we generate two types of adversarial negatives, yielding a total of 890 negative samples.

Category	Homophonic	Homographic	Total
Positive Samples	194	251	445
Negative Samples:			
Explicative Substitution (ES)	194	251	**445**
Random Substitution (RS)	194	251	**445**
Total Negatives	388	502	**890**
Total (Pos + Neg)	**582**	**753**	**1335**

Table 5: Dataset statistics for MULTIPUN.

B Linguistic Filtering Criteria

B.1 WordNet Lexical File Categories

Table 6 lists the WordNet lexical file categories used in our filtering pipeline. We retain only nouns from *visual* categories (e.g., `noun.animal`, `noun.artifact`) to ensure imageability, while filtering out abstract concepts.

Category	Lexname	Description
Visual	`noun.animal`	Animals and distinct biological organisms
	`noun.artifact`	Man-made objects, tools, and instruments
	`noun.body`	Body parts (used restrictively)
	`noun.food`	Edible substances and dishes
	`noun.object`	Natural inanimate objects (e.g., stones)
	`noun.plant`	Vegetation and botanical entities
Abstract	`noun.location`	Spatial locations and regions
	`noun.substance`	Substances and bodies of matter
	`noun.act`	Actions, events, and processes
	`noun.attribute`	Qualities, properties, and attributes
	`noun.cognition`	Cognitive processes and contents
	`noun.communication`	Communicative processes and contents
	`noun.feeling`	Emotions, feelings, and sensations
	`noun.motive`	Goals, motives, and wants
	`noun.quantity`	Quantities, units, and measurements
	`noun.time`	Temporal points and periods
	`noun.Tops`	Top-level unique beginners

Table 6: Classification of WordNet Lexnames into Visual Anchor Categories (retained) and Abstract Categories (filtered).

B.2 Frequency Thresholds

To ensure common usage, we apply specific Zipf frequency thresholds. For homophonic puns, we require a frequency greater than 3.0 for both w_p and w_a. For homographic puns, we impose a higher threshold of 3.8 for w_p to ensure recognizability given that both senses share the same word form.

Algorithm 1 Diversity Filtering

1: **Input:** candidate dataset $\mathcal{D} = \{d_i\}_{i=1}^{N}$, target size k ($k < N$), embedding function EMB
2: **Output:** filtered diverse subset $\mathcal{D}' \subseteq \mathcal{D}$ with $|\mathcal{D}'| = k$, minimum pairwise distance $d_{\min}$
3: Compute sentence embeddings $e_i \leftarrow \text{EMB}(d_i)$ for all $i = 1, \ldots, N$
4: Construct pairwise cosine distance matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ by $D_{ij} = 1 - \frac{e_i^\top e_j}{\|e_i\| \, \|e_j\|}$, $D_{ii} \leftarrow +\infty$ ▷ Lower D_{ij} indicates higher semantic similarity
5: Initialize active candidate set $\mathcal{S} \leftarrow \{1, \ldots, N\}$
6: **for** iteration $t = 1$ **to** $N - k$ **do**
7: Identify the most similar pair $(i, j) \leftarrow \arg\min_{p \neq q,\, p,q \in \mathcal{S}} D_{pq}$ ▷ Find the closest pair with minimum distance
8: Calculate redundancy scores for the closest pair $\phi_i = \sum_{v \in \mathcal{S}} D_{iv}, \phi_j = \sum_{v \in \mathcal{S}} D_{jv}$ ▷ Lower ϕ indicates higher centrality
9: Select the more redundant candidate:$u \leftarrow \arg\min\{\phi_i, \phi_j\}$ ▷ Choose the candidate closer to the remaining set
10: Update active set: $\mathcal{S} \leftarrow \mathcal{S} \setminus \{u\}$ ▷ Remove the more redundant candidate
11: **end for**
12: Construct final subset $\mathcal{D}' \leftarrow \{d_i \mid i \in \mathcal{S}\}$
13: Compute diversity $d_{\min} \leftarrow \min_{i \neq j,\, i,j \in \mathcal{S}} D_{ij}$
14: **return** $\mathcal{D}'$, $d_{\min}$

B.3 Diversity Filtering

We use the deterministic filtering process outlined in Algorithm 1 to select the final k items. Given the candidate dataset $\mathcal{D}$ of N items, we first compute the sentence embeddings $e_i = \text{EMB}(d_i)$ for all items using `text-embedding-3-large`, where d_i is the ground-truth rationale text. We then construct the pairwise cosine distance matrix $\mathbf{D}$. The algorithm iteratively prunes the dataset $N-k$ times. In each iteration, it identifies the most similar pair of candidates (i, j) in the active set $\mathcal{S}$ (Line 6). To decide which candidate to remove, it calculates a redundancy score ϕ for both i and j, defined as the sum of distances to all other active candidates (Line 8). The candidate with the smaller ϕ is deemed more central or more redundant and is removed from $\mathcal{S}$ (Lines 10 and 12). By iteratively removing the most redundant candidate from each closest pair, this process ensures that semantic outliers are preserved (Li et al., 2025, 2026a,b; Lan et al., 2025), and the final set of k items maintains

maximum conceptual diversity and coverage (Wu et al., 2026b, 2024, 2025c).

C Generation Prompts

This section provides the prompt templates used for generating positive pun samples and adversarial negative samples in the MULTIPUN dataset.

C.1 Positive Sample Generation

C.1.1 Homophonic Pun Creation Prompt

Creative Prompt for Homophonic Puns

Role
You are an expert in multimodal humor. Your task is to generate visual pun data based on **Homophones** (words that sound the same but have different meanings and spellings).
Task Definition
I will provide you with two words:
1. **Word A (Visual Object):** The word that determines the visual appearance (S_p).
2. **Word B (Hidden Context):** The word that determines the behavior/action (S_a).
You need to generate:
1. **Image Description:** Description of Object A acting out the meaning of Word B.
2. **Caption:** A sentence containing Word A, but implying Word B.
3. **Interpretation:** An analysis of the pun.
Example
Input:
* **Word A:** pear: sweet juicy gritty-textured fruit available in many varieties
* **Word B:** pair: two items of the same kind
Output:
Image Description: Two cartoon pears holding hands and smiling happily at each other.
Caption: We make a great pear.
Interpretation: Visual depicts two pears (literal object, S_p) holding hands like a romantic pair (figurative behavior, S_a). The caption exploits the homophonic relationship between 'pear' (w_p) and 'pair' (w_a), creating humor through sound similarity between different meanings.
Current Input
* **Word A: [Insert Word A, e.g., Chili: a small hot-tasting pod of a variety of capsicum]**
* **Word B: [Insert Word B, e.g., Chilly: uncomfortably cool or cold]**
# Output

C.1.2 Homographic Pun Creation Prompt

Creative Prompt for Homographic Puns

Role
You are an expert in multimodal humor. Your task is to generate visual pun data based on **Homographic Puns** (a single word with multiple meanings in the same spelling).
Task Definition
I will provide you with one word and its two distinct definitions:
1. **The Word:** The lexical item used in the caption.
2. **Definition 1 (Visual Object):** The literal/concrete meaning that determines the physical appearance of the object (S_p).
3. **Definition 2 (Hidden Context):** The figurative behavior/state meaning that determines the behavior, action, or setting (S_a).
You need to generate:
1. **Image Description:** A description of the object from Definition 1 performing the action or situated in the context of Definition 2.
2. **Caption:** A witty sentence using "The Word", where the sentence structure strongly implies Definition 2.
3. **Interpretation:** A concise explanation of the pun mechanism.
Example
Input:
* **The Word:** fan
* **Definition 1:** a device for creating a current of air by movement of a surface or surfaces
* **Definition 2:** an ardent follower and admirer
Output:
* **Image Description:** A large electric floor fan in a stadium seat, holding a foam finger and cheering loudly.
* **Caption:** I'm your biggest fan.
* **Interpretation:** Visual shows a cooling fan (literal object, S_p); caption uses 'fan' as admirer (figurative behavior, S_a), creating a homographic pun where the same word embodies both meanings.
# Current Input
* **The Word: [Insert Word Here]**
* **Definition 1 (Visual Object): [Insert Literal Definition Here]**
* **Definition 2 (Hidden Context): [Insert Abstract/-Contextual Definition Here]**
# Output

C.2 Adversarial Negative Sample Generation

C.2.1 Explicative Substitution

Explicative Substitution Generation

You are a data augmentation expert. Given the following pun, generate an Explicative Substitution variant:
Original Caption: **{caption}**
Pun Word (w_p): **{word}**
Hidden Meaning (S_a): **{meaning}**
Task: Replace w_p with an EXPLICIT STATEMENT of the hidden meaning S_a.
Constraints:
- Do NOT use w_p or w_a directly
- Use paraphrases or synonyms to express S_a
- Adjust grammar if needed for naturalness
- Prefer single-word replacements when possible
Example:
Original: "We make a great pear."
Hidden Meaning: romantic couple
Output: "We make a great romantic couple."

C.2.2 Random Substitution

Random Substitution Generation

You are a data augmentation expert. Given the following pun, generate a Random Substitution variant:
Original Image Prompt: **{visual description}**
Original Caption: **{caption}**
Pun Word (w_p): **{word}**

Task:
1. Select a RANDOM concrete noun (e.g., chair, banana, bicycle, umbrella, book) that is SEMANTICALLY UNRELATED to the original pun context
2. Replace the main object in the image prompt with this random entity
3. Replace w_p in the caption with the same random entity
4. Keep the same action/context structure
Constraints:
- The random entity must be a concrete, visualizable noun
- Must be completely unrelated to original pun
- Do NOT reuse common examples (vary your selection)
Example:
Original Visual: "Two cartoon pears holding hands..."
Original Caption: "We make a great pear."
Random Entity: banana
New Visual: "Two cartoon bananas holding hands..."
New Caption: "We make a great banana."

D Human Verification Protocol

We recruited three graduate students from our institution with prior experience in NLP research (Liu et al., 2026). All participants were aged 20-28 years and consisted of two male and one female doctoral students in computer science. Participants were compensated at $25 USD/hour (approximately 8 hours per participant) and provided informed consent. All annotations were anonymized and used only for academic research (International Organization for Standardization, 2024a,b). All generated samples (positive and negative) undergo human verification. Three annotators independently evaluate each sample based on:

1. **Image Quality:** Is the visual content clear, non-distorted, and depicts the intended object?
2. **Visual-Textual Coherence:** For positive samples, does the visual content coherently connect to the text description? For negative samples, is the intended disruption (ES/RS) clearly present?
3. **Ambiguity Presence:** For positive samples, is there genuine dual-layer semantics? For negative samples, is the ambiguity properly resolved?
4. **Naturalness:** Are the caption and visual scenario natural and plausible?

Samples are retained if at least 2 out of 3 annotators agree on acceptance. Rejected samples are either regenerated with refined prompts or discarded. The inter-annotator agreement (Fleiss' Kappa) across all samples is 0.78, indicating substantial agreement.

E Evaluation Suite Task Descriptions

Our evaluation suite comprises three recognition tasks with progressive levels of structural guidance: **Detection**, **Localization**, and **Explanation**. For each task, we use two prompt variants to separate true reasoning from affirmative language bias (Liu et al., 2025): (1) *biased-to-pun* prompt that asks whether the given context is a pun, and (2) *biased-to-non-pun* prompt that asks whether the given context is not a pun. The key difference is in the task description and output order, while the definitions and requirements remain identical.

All experiments are run three times, and the reported results are averages. All baselines follow their official implementations.

E.1 Detection

This task asks for binary judgment (pun or not). We provide two variants: one without formal definitions and one with formal definitions and notation.

E.1.1 Pun Detection

Detection without Definitions (Biased-to-Pun)

You are an expert linguist specializing in Multimodal Puns.
Task Description
Analyze the provided image and caption to determine if they constitute a **Multimodal Pun**.
Input Data
Caption: **{caption}**
Output Requirements
Output ONLY a JSON object:
{"is_pun": true/false}
IMPORTANT: Output ONLY the JSON object, no additional text or explanation.

*Note: The biased-to-non-pun variant changes the task description to "determine if they constitute a **Non-Pun** (not a pun)" and adds "Note: Answer true if it is a pun, false if it is a non-pun."*

E.2 Pun Localization

This task requires first judging and explicitly identifying words w_p and w_a.

Localization (Biased-to-Pun)

You are an expert linguist specializing in Multimodal Puns.
Task Description
Analyze the provided image and caption to determine if they constitute a **Multimodal Pun**. If yes, categorize the pun type and extract ONLY the word pair (w_p and w_a).
Definitions
1. **Homophonic Pun:** The caption contains a word that sounds like another word with **different spelling and meaning**.

- w_p: The word **actually appearing in the caption**
- w_a: The hidden word it sounds like (different spelling/meaning)
- Example: "pear" (in caption) sounds like "pair" (hidden meaning)

2. **Homographic Pun:** The caption contains a word with **two distinct meanings in the same spelling**.
   - w_p and w_a are the **same word** appearing in the caption (both should be identical)
   - Example: "fan" means both "cooling device" and "enthusiast"

Input Data
Caption: **{caption}**
Output Requirements
If it is NOT a pun:
{"is_pun": false}
If it IS a pun:
{ "is_pun": true, "type": "<Homophonic or Homographic>", "tuple": { "wp": "<The EXACT word appearing in the caption>", "wa": "<The hidden/alternative word>" } }
IMPORTANT: Output ONLY the JSON object with the fields shown above. Do NOT include semantic definitions (S_p or S_a). Only provide the word pair (wp and wa). No additional text or explanation.

E.3 Pun Explanation

This task requires judging, providing a rationale that explains why it's a pun, and extracting the full tuple $\langle w_p, w_a, S_p, S_a \rangle$.

Explanation (Biased-to-Pun)

You are an expert linguist specializing in Multimodal Puns.
Task Description
Analyze the provided image and caption to determine if they constitute a **Multimodal Pun**. If yes, categorize the pun type and extract the linguistic components following the formal notation $P = \langle w_p, w_a, S_p, S_a \rangle$.
CRITICAL RULE: What is a Multimodal Pun?
A multimodal pun MUST satisfy ALL of the following conditions:

1. **The pun word MUST explicitly appear in the caption text**
2. **This word must create dual meanings through either:**
   - **Phonetic similarity** (sounds like another word with different spelling/meaning)
   - **Lexical polysemy** (same spelling but two distinct meanings)
3. **Visual-linguistic coupling:** The image fuses a literal object (S_p) with a figurative behavior/state (S_a), while the text unifies them through the pun word

IMPORTANT: If the caption does not contain the pun word, or if the visual and textual meanings are not genuinely linked, it is NOT a multimodal pun.
Definitions

1. **Homophonic Pun:** Exploits sound similarity between words with **different spelling and meaning**.
   - w_p: The word **actually appearing in the caption**
   - w_a: The hidden word it sounds like (different spelling/meaning)
   - S_p: The literal/concrete object depicted in the image
   - S_a: The figurative behavior/state associated with the alternative word
   - Example: "We make a great pear" — image shows pears (S_p) holding hands like a romantic pair (S_a)
2. **Homographic Pun:** Exploits dual meanings of a word with **the same spelling**.
   - w_p and w_a are the **same word** appearing in the caption
   - S_p: The concrete/literal sense depicted visually in the image
   - S_a: The figurative/abstract sense implied by the textual context
   - Example: "I'm a big fan of yours" — image shows a cooling fan (S_p) cheering like an enthusiast (S_a)

Input Data
Caption: **{caption}**
Analysis Steps

1. **First, identify if there is a word in the caption that could have dual meanings**
2. **Check if one meaning relates to the image and another to the text context**
3. **Only if BOTH conditions are met, classify as a pun**

Output Requirements
Condition A: If it is NOT a pun:
Output exactly this JSON:
{"is_pun": false}
Condition B: If it IS a pun:
The pun word MUST be present in the caption. Output:
{ "is_pun": true, "type": "<Homophonic or Homographic>", "explanation": "<Brief explanation of how the pun creates humor through visual-linguistic interplay>", "tuple": { "wp": "<The EXACT word appearing in the caption that creates the pun>", "wa": "<The alternative word: different spelling if Homophonic, same spelling if Homographic>", "Sp": "<The literal/concrete meaning shown in the image>", "Sa": "<The figurative/abstract meaning implied by context>" } }
IMPORTANT: Output ONLY the JSON object, no additional text or explanation.

F Pun-CoT: Enhanced Prompt with Three-Stage Verification

To address the hallucination errors identified in our error analysis (Section 4.1), we propose **Pun-CoT** (Pun-aware Chain-of-Thought), an enhanced prompt that enforces a structured three-stage verification process. This method is designed to mitigate four common error patterns: pun keyword hallucination, phonetic hallucination, semantic hallucination, and visual object hallucination.

Pun-CoT Enhanced Prompt (Biased-to-Pun)

You are an expert linguist specializing in Multimodal Puns.
Task Description

Analyze the provided image and caption to determine if they constitute a **Multimodal Pun**. Use a structured three-stage verification process to avoid common errors.

Formal Definition

A multimodal pun is represented as $P = \langle w_p, w_a, S_p, S_a \rangle$ where:

- w_p: The pun word **explicitly appearing in the caption**
- w_a: The alternative word (hidden meaning)
- S_p: The literal/concrete object sense (depicted visually in the image)
- S_a: The figurative behavior/state sense (implied by textual context)

Pun Types

1. **Homophonic Pun:** Exploits sound similarity between words with **different spelling and meaning**
   - Example: "pear" (in caption) sounds like "pair" (hidden meaning)
   - Image shows pears (literal object) holding hands like a romantic pair (figurative behavior)
2. **Homographic Pun:** Exploits dual meanings of a word with **the same spelling**
   - Example: "fan" means both "cooling device" and "enthusiast"
   - Image shows a fan device (literal object) cheering like an enthusiast (figurative behavior)

CRITICAL THREE-STAGE VERIFICATION

STAGE 1: Visual Grounding (Prevent Visual Object Hallucination)

- First, describe EXACTLY what visual object you see in the image
- DO NOT infer objects based on text context
- DO NOT assume objects that are not visually present
- Example: If you see apples, do NOT call them "dates" even if the text mentions "date"

STAGE 2: Lexical Anchoring (Prevent Pun Keyword Hallucination)

- Identify the EXACT words in the caption text
- DO NOT mentally replace words with idiom components
- Example: If caption says "I'm your biggest lamp", do NOT treat it as if it says "fan"
- List all potential pun candidates from the ACTUAL caption words

STAGE 3: Cross-Modal Verification (Prevent Phonetic/Semantic Hallucination)

For each potential pun word, verify:

a) **Phonetic Bridge (for Homophonic):** Do w_p and w_a ACTUALLY sound similar?

- REJECT if phonetically distinct (e.g., "banana" does NOT sound like "soul")
- Require genuine phonetic similarity

b) **Semantic Bridge (for Homographic):** Does the word have TWO established meanings?

- REJECT if forcing meanings onto unrelated words
- Example: "banana" does NOT have a meaning related to "pair" or "couple"

c) **Visual-Textual Link:** Does the visual object connect to text via valid pun mechanism?

- For Homophonic: Visual shows S_p (literal object of w_p), text implies S_a (figurative behavior of w_a)
- For Homographic: Same word connects both the literal visual sense and figurative textual sense
- REJECT weak or fabricated connections

Input Data

Caption: **{caption}**

Output Requirements

If it is NOT a pun (failed any verification stage):
{"is_pun": false}
If it IS a pun (passed all verification stages):
{ "is_pun": true, "type": "<Homophonic or Homographic>", "explanation": "<Brief explanation of the verified pun mechanism>", "tuple": { "wp": "<The EXACT word appearing in the caption>", "wa": "<The alternative word: different spelling if Homophonic, same spelling if Homographic>", "Sp": "<The literal/concrete meaning shown in the image>", "Sa": "<The figurative/abstract meaning implied by context>" } }
IMPORTANT:

- Execute ALL three verification stages before making judgment
- Be conservative: when in doubt, classify as NOT a pun
- The pun word MUST explicitly appear in the caption
- Output ONLY the JSON object, no additional text

G Model Configuration

We evaluate a total of 11 VLMs. Tables 7 and 8 provide comprehensive overviews of all evaluated models and their configurations.

G.1 Closed-Source VLMs

Table 7 presents the configuration details for closed-source models accessed via API.

Model	API Version
OpenAI Family	
GPT-5.1	gpt-5.1
GPT-4o	gpt-4o-2024-08-06
Google Gemini Family	
Gemini-3-Pro	gemini-3-pro-preview
Anthropic Family	
Claude-Sonnet-4.5	claude-sonnet-4-5-20250929

Table 7: Closed-source VLM configurations.

G.2 Open-Source VLMs

Table 8 presents the configuration details for open-source models. All models are evaluated using their officially released checkpoints from Hugging Face by hosting the model on a vLLM server.

G.3 Hardware

All open-source models are evaluated on two NVIDIA A100 80GB GPUs. Closed-source models are accessed via their official APIs.

Model	Checkpoint	Type
Meta Llama-4 Family		
Llama-4-Scout-17B	meta-llama/Llama-4-Scout-17B-16E-Instruct	Instruct
Alibaba Qwen3-VL Family		
Qwen3-VL-8B-Instruct	Qwen/Qwen3-VL-8B-Instruct	Instruct
Qwen3-VL-30B-A3B-Instruct	Qwen/Qwen3-VL-30B-A3B-Instruct	Instruct
Qwen3-VL-8B-Thinking	Qwen/Qwen3-VL-8B-Thinking	Reasoning
Qwen3-VL-30B-A3B-Thinking	Qwen/Qwen3-VL-30B-A3B-Thinking	Reasoning
LLaVA Family		
LLaVA-V1.6-Vicuna-13B	liuhaotian/llava-v1.6-vicuna-13b	Instruct

Table 8: Open-source VLM configurations.

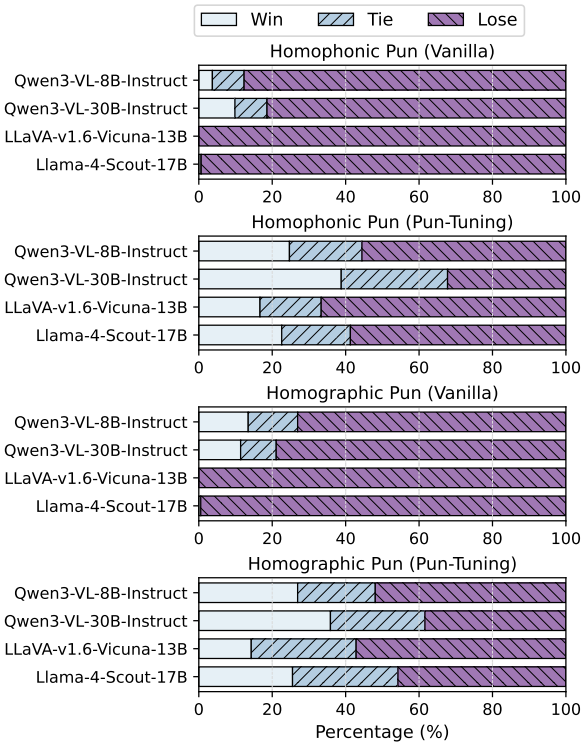


Figure 5: Pairwise comparison for pun explanations before and after Pun-Tuning.

H Additional Results

Figure 5 shows the pairwise comparison for pun explanations before and after Pun-Tuning.

I Pun-Tuning Implementation Details

I.1 Dataset Splits

We split the dataset ensuring no test samples leak into training. The 194 homophonic puns are divided into 97 training and 97 test samples; the 251 homographic puns are split into 125 training and 126 test samples. Negative samples maintain a 2:1 ratio with positive samples (each positive sample paired with 2 negatives: one Explicative Substitution and one Random Substitution). Table 9 shows the complete breakdown.

I.2 Hyperparameters

We fine-tune three open-source models (Qwen3-VL-8B-Instruct, Qwen3-VL-30B-A3B-Instruct, and LLaVA-V1.6-Vicuna-13B) with batch size 4 per A100 GPU, learning rate 2e-5, AdamW optimizer, linear warmup (100 steps) followed by cosine decay, weight decay 0.01, gradient clipping (max norm 1.0), and FP16 mixed precision for 3 epochs. Training uses both *biased-to-pun* and *biased-to-non-pun* prompt variants. Evaluation is performed on the held-out test set (669 samples) across all three tasks (Wu et al., 2025b, 2026a, 2025a).

Category	Pun Type	Train	Test	Total
Positive	Homophonic	97	97	194
	Homographic	125	126	251
Negative	Homophonic	194	194	388
	Homographic	250	252	502
Total		**666**	**669**	**1335**

Table 9: Dataset splits for Pun-Tuning experiments.

J Software Packages

We use the following Python packages: NLTK (version 3.9.2) for WordNet access and lemmatization, and the pronouncing package (version 0.2.0) for CMU Pronouncing Dictionary access.

K The Use of Large Language Models

We utilize LLMs to assist with language and code polishing, as well as error checking, during the preparation of this manuscript. The content, ideas, and scientific contributions remain entirely our own, and all substantive intellectual work is conducted by the authors (Xu et al., 2024a, 2025a,b,c,d; Attardo, 2024; Xu et al., 2026a,b,c; An et al., 2025, 2026).

L Failure Cases

L.1 Error Analysis of Negative Samples

VLMs often fail to correctly reject non-pun distractors, exhibiting systematic hallucination patterns across lexical, phonological, semantic, and visual levels. Table 10 provides qualitative examples of these four error types, demonstrating how models incorrectly interpret negative samples as genuine puns despite the absence of valid pun mechanisms.

L.2 Error Analysis in Pun Explanation

While VLMs can often detect puns, they struggle with providing accurate explanations. Table 11 categorizes four distinct failure modes: detection

failures, pun word errors, alternative word errors,
and cross-modal integration errors.

Pun Keyword Hallucination

Homographic Pun

I'm your biggest **fan**.

claude-sonnet-4-5
w_p: fan
S_p: A mechanical device with rotating blades that circulates air
w_a: fan
S_a: An enthusiastic admirer or supporter
Explanation: The word 'fan' has two meanings: (1) a mechanical device that circulates air, which is visually depicted in the image as an anthropomorphized electric fan, and (2) an enthusiastic supporter or admirer, as indicated by the foam finger and the phrase 'biggest fan' which is an idiom meaning someone's greatest supporter.

Random Substitution (RS)

I'm your biggest **lamp** in the room.

claude-sonnet-4-5
w_p: lamp
S_p: A lighting fixture consisting of a shade and base
w_a: fan
S_a: An enthusiastic supporter or admirer
Explanation: The word 'lamp' creates a pun by having two meanings: literally referring to the lamp character in the image, and playing on the phrase 'biggest fan' where 'lamp' sounds like it's substituting for 'fan' in the context of being a supporter at what appears to be a stadium/arena setting.

Phonetic Hallucination

Homophonic Pun

You are my **sole** mate.

Gemini-3-Pro
w_p: sole
S_p: The bottom part of a shoe or foot, which is visually depicted as two shoe bottoms hugging.
w_a: soul
S_a: The spiritual or immaterial part of a human being, used in the phrase 'soul mate' to mean a perfect romantic partner.
Explanation: The humor relies on the phonetic similarity between 'sole' (the bottom of a shoe, depicted in the image) and 'soul' (as in 'soul mate', a romantic partner). The image shows two shoe soles hugging, literalizing the pun.

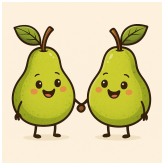
Random Substitution (RS)

You are my **banana** mate.

Gemini-3-Pro
w_p: banana
S_p: A long curved fruit which grows in clusters and has soft pulpy flesh and yellow skin when ripe.
w_a: soul
S_a: The spiritual or immaterial part of a human being or animal, regarded as immortal.
Explanation: The pun relies on the phonetic similarity between the word 'banana' and the word 'soul'. The phrase 'banana mate' sounds like 'soulmate', which is a common term for a perfect romantic partner or close friend. The image depicts two bananas hugging affectionately, reinforcing the 'mate' aspect, while the word 'banana' directly describes the visual subjects.

Semantic Hallucination

Homophonic Pun



We make a great **pear**.

GPT-4o
w_p: pear
S_p: A type of fruit depicted in the image.
w_a: pair
S_a: Two individuals who complement each other, forming a pair.
Explanation: The word 'pear' in the caption sounds like 'pair'. The image depicts two pears holding hands (a visual representation of 'pear'), while the caption implies a 'pair' of individuals who complement each other.

Random Substitution (RS)

We make a great **banana**.

GPT-4o
w_p: banana
S_p: A yellow fruit with a curved shape, as depicted in the image.
w_a: banana
S_a: A metaphorical reference to a pair or team that complements each other.
Explanation: The word 'banana' refers to the fruit depicted in the image, while in the context of the caption, it metaphorically refers to the idea of being a 'pair' or 'team' that works well together.

Visual Object Hallucination

Homophonic Pun

It's our first **date**.

Qwen3-VL-30B-A3B-Instruct
w_p: date
S_p: A dried fruit, specifically a date palm fruit.
w_a: date
S_a: A social meeting between two people, typically for romantic purposes.
Explanation: The word 'date' in the caption has two meanings. Visually, the image shows two dates (the fruit), which are literal 'dates'. Contextually, the phrase 'first date' refers to a romantic meeting. The humor arises from the play on the dual meaning of 'date'.

Explicative Substitution (ES)

It's our first **romantic meeting**.

Qwen3-VL-30B-A3B-Instruct
w_p: meeting
S_p: The visual representation of two dates (the fruit) as if they are on a romantic date.
w_a: meeting
S_a: A formal or social appointment, specifically a romantic date.
Explanation: The word 'meeting' in the caption has two meanings. Visually, the image shows two dates (the fruit) holding menus, which is a play on the word 'date' as in a romantic appointment. The pun lies in the word 'meeting,' which can mean a romantic date (the fruit) or a formal appointment. The humor arises from the dual meaning of 'meeting' in the context of a romantic encounter.

Table 10: Qualitative examples of false positive errors on negative samples. We illustrate four distinct hallucination patterns where VLMs fail to reject non-pun distractors: (1) Pun Keyword Hallucination (Lexical Level), where models ignore visible text to restore idiomatic keywords; (2) Phonetic Hallucination (Phonological Level), where models fabricate sound similarities between distinct terms; (3) Semantic Hallucination (Semantic Level), where models force polysemous meanings onto unambiguous synonyms; and (4) Visual Object Hallucination (Visual Level), where models misidentify visual objects to align with textual cues.

Detection Failure

Good **knight**.

Homophonic Pun
w_p: knight
S_p: originally a person of noble birth trained to arms and chivalry; today in Great Britain a person honored by the sovereign for personal merit
w_a: night
S_a: the time after sunset and before sunrise while it is dark outside
Explanation: The visual shows a chess knight piece wearing a nightcap and lying in bed, symbolizing bedtime. The caption, "Good knight," creates a pun by substituting "knight" (the chess piece) for "night" in the common bedtime farewell, "Good night." The humor arises from the homophonic wordplay, as "knight" and "night" are pronounced the same but have different spellings and meanings.

Qwen3-VL-8B-Instruct
is pun: False

Pun Words Error

Super **bowl**.

Homophonic Pun
w_p: bowl
S_p: a round vessel that is open at the top; used chiefly for holding food or liquids
w_a: bowl
S_a: a wooden ball (with flattened sides so that it rolls on a curved course) used in the game of lawn bowling
Explanation: The visual features a soup bowl (dish) depicted as engaging in the sport of bowling, humorously anthropomorphized to throw a ball at pins. The caption, "Super bowl," creates a pun through homography, as it plays on the identical spelling of "bowl" (a dish) and "bowl" (a sporting action). The humor arises from the unexpected juxtaposition of a kitchen item with sports terminology.

gemini-3-pro
w_p: super
S_p: An adjective describing the bowl of soup depicted in the image, specifically highlighting its identity as 'soup'.
w_a: soup-er
S_a: An adjective meaning excellent or of high grade; also part of the proper noun 'Super Bowl' referring to the NFL championship game.
Explanation: The humor relies on the phonetic similarity between the phrase 'Super bowl' (referring to the major American football championship event) and the phrase 'soup-er bowl' (describing a bowl of soup that is excellent at bowling). The image literally depicts a bowl of soup bowling, creating a visual pun on the homophone.

Alternative Word Error

What a **waist**.

Homophonic Pun
w_p: waist
S_p: the narrowing of the body between the ribs and hips
w_a: waste
S_a: any materials unused and rejected as worthless or unwanted
Explanation: The visual shows a garbage bin (representing "waste") wearing a belt around its middle, resembling a human waist. The caption, "What a waist," creates a homophonic pun by playing on the identical pronunciation of "waist" (a body part) and "waste" (garbage). The humor arises from the interplay between the visual of the trash bin and the double meaning of the caption.

Qwen3-VL-30B-A3B-Thinking
w_p: waist
S_p: The belt wrapped around the trash bin, mimicking a human waistline
w_a: waist
S_a: The narrow part of the human torso between the ribs and hips, typically used in compliments about body shape
Explanation: The word 'waist' in the caption refers to both the belt around the trash bin (visual sense) and the human body part (textual context), creating humor through the unexpected application of a human compliment to an inanimate object.

Cross-modal Integration Error

I'm so **board**.

Homophonic Pun
w_p: board
S_p: a flat piece of material designed for a special purpose
w_a: bored
S_a: uninterested because of frequent exposure or indulgence
Explanation: The visual features a wooden plank (a "board") depicted with a sleepy, disinterested face. The caption, "I'm so board," creates a homophonic pun by playing on the similar pronunciation of "board" (a piece of wood) and "bored" (a state of monotony). The humor arises from the dual interpretation of the word, linking the object to the emotion.

GPT-4o
w_p: board
S_p: A flat, thin, rectangular piece of material, such as wood, depicted in the image.
w_a: board
S_a: The state of being bored or uninterested, implied by the context of the caption.
Explanation: The word 'board' has two meanings: one relates to the wooden plank shown in the image, and the other refers to the feeling of boredom implied in the caption.

Table 11: Qualitative examples of failures on positive samples (genuine puns). We identify four failure modes: (1) Detection Failure, where the pun is missed entirely; (2) Pun Words Error, where the model focuses on the wrong lexical trigger; (3) Alternative Word Error, where the model fails to retrieve the hidden meaning (w_a) of the anchor word; and (4) Cross-modal Integration Error, where the model confuses the linguistic mechanism (e.g., treating homophony as polysemy).