Can LLMs Deceive CLIP? Benchmarking Adversarial Compositionality of Pre-trained Multimodal Representation via Text Updates

Jaewoo Ahn* **Heeseung Yun*** **Dayoon Ko** **Gunhee Kim**
Seoul National University
{jaewoo.ahn, heeseung.yun, dayoon.ko}@vision.snu.ac.kr, gunhee@snu.ac.kr
https://vision.snu.ac.kr/projects/mac

Abstract

While pre-trained multimodal representations (*e.g.*, CLIP) have shown impressive capabilities, they exhibit significant compositional vulnerabilities leading to counterintuitive judgments. We introduce Multimodal Adversarial Compositionality (MAC), a benchmark that leverages large language models (LLMs) to generate deceptive text samples to exploit these vulnerabilities across different modalities and evaluates them through both sample-wise attack success rate and group-wise entropy-based diversity. To improve zero-shot methods, we propose a self-training approach that leverages rejection-sampling fine-tuning with diversity-promoting filtering, which enhances both attack success rate and sample diversity. Using smaller language models like Llama-3.1-8B, our approach demonstrates superior performance in revealing compositional vulnerabilities across various multimodal representations, including images, videos, and audios.

1 Introduction

Recent advances in multimodal systems have demonstrated remarkable capabilities in generating multimodal content from multimodal inputs. At the core of these developments lies pre-trained multimodal representations, which can encode rich information from different modalities. Such representations, notably illustrated by Contrastive Image-Language Pre-Training (CLIP) (Radford et al., 2021), has become an indispensable component in modeling complex contextual understanding in crossmodal settings, finding widespread applications across retrieval (Luo et al., 2022; Ahn et al., 2023), generation (Ramesh et al., 2022), and reward modeling (Yu et al., 2023a; Rocamonde et al., 2024). Moreover, its usage has become commonplace across various modalities beyond image-language pairs.

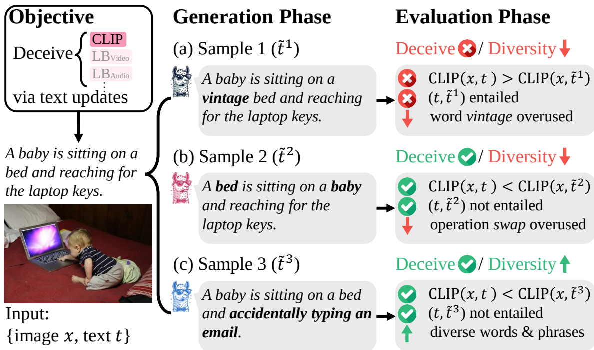


Figure 1: Key idea of Multimodal Adversarial Compositionality (MAC). MAC benchmarks compositional vulnerabilities of a pre-trained multimodal representation (*e.g.*, CLIP, LanguageBind) with a comprehensive set of criteria. CLIP$(\cdot, \cdot)$ denotes the cosine similarity between image and text embeddings from CLIP.

Contrary to their prevalence in a wide range of downstream applications, pre-trained multimodal representations are known to be considerably brittle. This brittleness can be intuitively exemplified by compounding text elements. As illustrated in Fig. 1-(b), with an image of a baby sitting, these systems may assign a high similarity score to an erroneous description like "a *bed* is sitting on a *baby*" than the correct description. Such counterintuitive judgments occur surprisingly often, implying a critical issue where the vulnerabilities in the embeddings are inherited by the models that utilize them. Consequently, there have been active efforts to identify these weaknesses through negative samples constructed from the perspective of visual compositional reasoning (*i.e.*, structured relationship between words and their corresponding visual elements), such as negation, event swapping, and attribute replacement (Thrush et al., 2022; Ma et al., 2023). However, developing a comprehen-

*Equal Contribution

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26382–26402
July 27 - August 1, 2025

sive understanding of diverse *compositional vulnerabilities*, without assuming specific scenarios, remains an open challenge.

In this work, we introduce the challenge of large language models (LLMs) deceiving CLIP, *i.e.*, exploiting weaknesses in how pre-trained multimodal representations encode relationships between objects and attributes in multimodal contents (*e.g.*, image). To this end, we propose to benchmark the **M**ultimodal **A**dversarial **C**ompositionality (MAC) of a target representation. Given multimodal data pairs (*e.g.*, image-caption), LLMs generate deceptive captions by slightly modifying ground-truth captions in a way that misaligns or contradicts the original content. We then rigorously evaluate whether the target representation mistakenly prefers these generated captions over the original ones. Unlike previous studies that address compositionality within specific modalities (Thrush et al., 2022; Bansal et al., 2024; Ghosh et al., 2024), our work highlights a key distinction in deceiving a target representation in a modality-agnostic manner (*e.g.*, image, video, audio).

For evaluation, given a set of captions generated by LLMs for deceiving, we propose a testbed that assesses their effectiveness through *sample-wise* and *group-wise* evaluation. We first evaluate whether each generated sample successfully executes an attack (*sample-wise*). This success requires meeting multifaceted conditions: the generated deceptive sample should (i) maintain high crossmodal similarity with the original multimodal input, (ii) contain non-entailing content while (iii) maintaining lexical similarity to the original text, and (iv) adhere to prescribed instructions without relying on shortcuts. Furthermore, if they are predictable or monotonous, they become easily defensible and fail to unravel *diverse* compositional vulnerabilities. Therefore, we design entropy-based metrics to measure the diversity of composition elements used in deception across the set of generated samples (*group-wise*).

In addition, we leverage the self-training of LLMs (Huang et al., 2023), particularly rejection sampling fine-tuning (Touvron et al., 2023) for the first time, where generated samples are used for additional training to promote deceptive response generation. Existing zero-shot sample generation for compositionality and naïve self-training methods often fail to elicit diverse compositions using a limited set of elements. To address this limitation, we propose a *diversity-promoting* self-training approach by thorough sampling among sample candidates. Even with smaller LLMs centered around Llama-3.1-8B (Dubey et al., 2024), our simple yet effective framework can substantially improve both attack success rates and diversity. We achieve superior deception performance compared to prior work across various representations for multiple modalities, including image, video, and audio. In particular, our method outperforms existing approaches (Yarom et al., 2023; Momeni et al., 2023; Ghosh et al., 2024), when evaluated on COCO (Lin et al., 2014), MSRVTT (Xu et al., 2016), and AudioCaps (Kim et al., 2019), successfully deceiving target models, notably CLIP (Radford et al., 2021) and LanguageBind (Zhu et al., 2024).

2 Related Work

Multimodal Compositional Reasoning. Often studied in the vision-language domain, it refers to the structured relationship between words and their corresponding visual elements (Thrush et al., 2022). It serves as a key indicator of whether models truly understand multimodal contexts, impacting critical tasks such as negative sample mining (Shekhar et al., 2017; Zhao et al., 2022; Yuksekgonul et al., 2022) and hallucination mitigation (Li et al., 2023b). To evaluate compositional reasoning, multiple benchmarks have been introduced to focus on robustness (Park et al., 2024), systematicity (Ma et al., 2023), and cross-domain alignment (Yarom et al., 2023). Another line of work enhances compositional reasoning by curating training data (Doveh et al., 2023; Li et al., 2024b; Patel et al., 2024) and regularizing learning objectives (Oh et al., 2024). Recent efforts have expanded beyond image-text interactions to explore and improve compositionality in video-language (Liu et al., 2020; Park et al., 2022; Momeni et al., 2023; Bansal et al., 2024) and audio-language contexts (Ghosh et al., 2024).

Most closely related to our work is SugarCrepe (Hsieh et al., 2023), which addresses the limitations of existing benchmarks by filtering nonsensical and non-fluent text to avoid trivial solutions. NaturalBench (Li et al., 2024a) focuses on generating challenging visual QA pairs easy for humans but difficult for models. While both works employ adversarial filtering for compositional vulnerability, they primarily address bias balancing or human plausibility within image-text interactions. In contrast, we approach compositionality from a modality-agnostic perspective and demonstrate this

Method	Modality (Image, Video, Audio)	Generation	Text Update (Replace, Swap, Add)	Compositionality Criteria			
				Crossmodal	Unimodal	Lexical	Diversity
FOIL (Shekhar et al., 2017)	I	Rule-based	Specific (R)	E, F	F	F	-
Winoground (Thrush et al., 2022)	I	Human-annotated	Specific (S)	E, F	F	F	-
VL-CheckList (Zhao et al., 2022)	I	Rule-based	Specific (R)	E, F	F	F	-
RoCOCO (Park et al., 2024)	I	Rule-based	Specific (R)	E, F	F	F	-
ARO (Yuksekgonul et al., 2022)	I	Rule-based	Specific (S)	E, F	F	F	-
SVLC (Doveh et al., 2023)	I	Rule-based	Specific (R)	E, F	F	F	-
CREPE (Ma et al., 2023)	I	Rule + LLM	Specific (R, S, A)	E, F	F	F	-
SugarCrepe (Hsieh et al., 2023)	I	LLM (ChatGPT)	Specific (R, S, A)	E, F	F	F	-
SeeTrue (Yarom et al., 2023)	I	LLM (PaLM)	General	E, F	F	-	-
LLaVA-Score (Li et al., 2024b)	I	LLM (GPT-4)	Specific (R, S)	E, F	F	F	-
FSC-CLIP (Oh et al., 2024)	I	Rule-based	Specific (R, S)	E, F	F	F	-
TripletCLIP (Patel et al., 2024)	I	SLM (Mistral-7B)	General	E, F	F	-	-
NaturalBench (Li et al., 2024a)	I	Human-annotated	General	E, F	F	F	-
VIOLIN (Liu et al., 2020)	V	Human-annotated	General	E, F	F	-	-
VLContrastSet (Park et al., 2022)	V	Rule + LLM	Specific (R)	E, F	F	F	-
VFC (Momeni et al., 2023)	V	LLM (PaLM)	Specific (R)	E, F	F	F	-
VideoCon (Bansal et al., 2024)	V	LLM (PaLM-2)	Specific (R, S, A)	E, F	F	F	-
Vinoground (Zhang et al., 2024)	V	Human + LLM	Specific (S)	E, F	F	F	-
CompA (Ghosh et al., 2024)	A	LLM (GPT-4)	Specific (R, S)	E, F	F	F	-
MATCH (Kuan and Lee, 2025)	A	Human-annotated	Specific (S)	E, F	F	F	-
MAC (Ours)	I, V, A	SLM (Llama3-8B)	General, Specific	E, F	E, F	E, F	E, F

Table 1: Overview of text-centric frameworks/benchmarks for multimodal compositionality. General/Specific denotes whether specific types of text operations are requested upon sample generation or not. Lexical indicates additional sample-wise constraints like instruction-following capability. (**E**: Evaluate, **F**: Filter).

across image, video, and audio modalities. While Tang et al. (2024) uses a claim manipulator model to contradict these modalities, our work highlights a key distinction by grounding the contradiction and diversity in a *quantifiable* measure of deceiving the target multimodal representation. Moreover, we extend our filtering criteria to better *generate* such samples in terms of diversity and successful deception via self-training.

Multimodal Adversarial Attack on Text. Adversarial attacks (Szegedy et al., 2014) manipulate input data to perturb a model's embedding space or induce incorrect predictions, systematically revealing vulnerabilities. In continuous domains like images, attacks typically inject subtle noise to mislead inference or maliciously control model behavior (Dong et al., 2018; Su et al., 2019; Shayegani et al., 2023a). In discrete domains like text, common strategies include identifying and replacing vulnerable words (Li et al., 2020), gradient-based attacks with Gumbel-softmax (Guo et al., 2021), masked token perturbations (Li et al., 2021), and LLM-based refinement (Mehrotra et al., 2024).

Text-based adversarial attacks can be extended to multimodal data, particularly targeting retrieval performance in image-text pairs by combining image noise injection and text perturbation. For instance, Co-Attack (Zhang et al., 2022) applies multimodal distribution-aware collaborative perturbations to image-text pairs while maintaining crossmodal consistency. Other methods enhance attack transferability via crossmodal guidance (Lu et al., 2023; Xu et al., 2024; Gao et al., 2024) or iterative search-based black-box attacks (Yin et al., 2023; Yu et al., 2023b). Recent studies have expanded attacks to video (Yang et al., 2024b) or audio (Bagdasaryan et al., 2024) beyond image-text pairs. However, these approaches focus on embedding perturbations, often resulting in either simple paraphrasing or unnatural text modifications without considering their entailment with the original text. To address these limitations, we instead apply a compositionality-aware modification that enables embedding-level perturbations while maintaining naturalness and semantic plausibility.

3 MAC: Multimodal Adversarial Compositionality

3.1 Problem Definition

Our **M**ultimodal **A**dversarial **C**ompositionality benchmark (**MAC**) is illustrated in Fig. 2. Given a target pre-trained multimodal representation that we want to deceive (*e.g.*, CLIP), MAC evaluates how effectively we can expose compositional vulnerabilities by updating text elements in multimodal data pairs. We use text updates as an anchor since it allows for modality-agnostic assessment and is more intuitively aligned with human interpretation than noise injection (Szegedy et al., 2014). Given a set of paired data $\mathcal{D} = (t_i, x_i)_{i=1}^{M_\mathcal{D}}$,

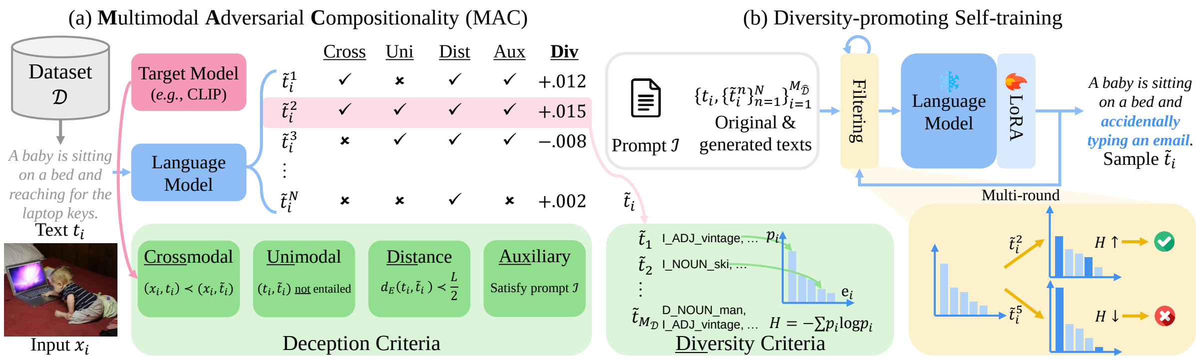


Figure 2: Overview of (a) multimodal adversarial compositionality and (b) diversity-promoting self-training.

where t_i represents text and x_i represents a paired input modality (*e.g.*, images), we aim to generate a set of adversarial text $\{\tilde{t}_i\}_{i=1}^{M_\mathcal{D}}$ that effectively exploit the compositional vulnerabilities of a target pre-trained multimodal representation denoted by f, which encodes both t_i and x_i into embeddings $y_{t_i}, y_{x_i} = f(t_i, x_i) \in \mathbf{R}^d$.

The generation of adversarial text $\{\tilde{t}_i\}_{i=1}^{M_\mathcal{D}}$ comprises two key components: (1) an adversarial sample *generator* g that produces up to N adversarial text samples $\{\tilde{t}_i^n\}_{n=1}^N$ under a specified budget constraint, and (2) a sample *filterer* h that identifies the most effective adversarial text sample $\tilde{t}_i$ from the N candidates based on their potential to deceive the pre-trained model f.

Defining the multimodal compositionality problem as MAC offers several advantages. First, since MAC does not assume a specific type of modality, it can be seamlessly applied to various formats including image, video, and audio. Second, previous compositionality frameworks that utilize rule-based or LLM-based generators for text updates, as well as our self-training-based generators (Sec. 4) can be consistently compared under our testbed to determine which framework more effectively deceives the target representation.

3.2 Sample-wise Deception Evaluation

Crossmodal Criterion. First and foremost, the generated sample should achieve the intended attack. The criterion is to deceive the target model f such that the model determines the generated adversarial sample is more closely aligned with the input modality than the original text. For an i-th data pair (t_i, x_i) and a generated sample $\tilde{t}_i$, crossmodal attack success is

$$s_i^c = \mathbf{I}(d_\theta(y_{t_i}, y_{x_i}) < d_\theta(y_{\tilde{t}_i}, y_{x_i})), \qquad (1)$$

where $\mathbf{I}$ is an indicator function, and d_θ is an embedding distance, where we use cosine similarity. For instance, in Fig. 1-(c), $d_\theta(y_{t_i}, y_{x_i})$ and $d_\theta(y_{\tilde{t}_i}, y_{x_i})$ are 0.34 and 0.37, respectively, indicating a successful attack on CLIP.

Unimodal Criterion. While the crossmodal distance is a well-established measure, this criterion alone may lead to results that merely amount to paraphrasing, as demonstrated in various adversarial attack scenarios (Zhang et al., 2022; Lu et al., 2023). To prevent this, another crucial criterion is that there should be a meaningful semantic distinction between the generated sample and the original text. Unimodal attack success for the i-th data pair is defined as follows:

$$s_i^u = \Pi_j \mathbf{I}(l_j(t_i, \tilde{t}_i) < \tau), \qquad (2)$$

where τ is a threshold for similarity and l_j indicates an unimodal text model to measure entailment between two text samples (Yarom et al., 2023; Ma et al., 2023). We use the agreement of multiple off-the-shelf NLI models (Liu et al., 2019; Lewis et al., 2020; He et al., 2021). We use $\tau = 0.5$, following Bansal et al. (2024). In Fig. 1-(c), all NLI models assess that the generated caption "accidentally typing an email" does not entail "reaching for the keys", indicating a successful unimodal attack. Note that we perform a preliminary evaluation using GPT-4 on 1K samples to verify the robustness of s_i^u, showing a concordance rate of over 93% with GPT-4.

Distance Criterion. Model-based evaluation of unimodal gap effectively reflects the differences between embeddings; however, it may unfairly favor irrelevant text samples, which goes against the purpose of deceiving the original pair. Therefore, the generated sample should execute attack with only limited lexical deviation from the original sample:

$$s_i^d = \mathbf{I}(d_E(t_i, \tilde{t}_i) < L_{\mathcal{D}}/2), \quad (3)$$

where d_E is the Levenshtein distance between original and generated samples (Ostrovsky and Rabani, 2007; Andoni and Nosatzki, 2020) and $L_{\mathcal{D}}$ is the average token length of dataset $\mathcal{D}$ for providing a dataset-specific limits in updates. In Fig. 1-(c), $d_E(t_i, \tilde{t}_i) = 4$ is less than $L_{\mathcal{D}}/2 \approx 5.21$, satisfying the distance criterion.

Auxiliary Criterion. Lastly, we evaluate whether a generated sample follows a set of predefined rules. For instance, as utilized by several frameworks in Table 1, if generation should be performed through specific operations (*e.g.*, `swap`), failing to comply with this cannot be considered a successful deception. Similarly, if trivial solutions are used, *e.g.*, negation (Ma et al., 2023), it is desirable for these to be filtered out as well. The auxiliary attack success of i-th pair s_i^a evaluates to true if it satisfies all predefined constraints (*e.g.*, prompt) through rule-based lexical validation. In Fig. 1-(b), the generated sample follows the `swap` operation by exchanging only two nouns ('baby' and 'bed') without additional modifications.

In total, the attack success rate R is

$$R = \frac{1}{M_{\mathcal{D}}} \sum_i (s_i^c \cdot s_i^u \cdot s_i^d \cdot s_i^a). \quad (4)$$

Although these elements have been partially highlighted in previous research, our key contribution lies in bringing them together to quantify the attack effectiveness. It enables consistent comparison across frameworks for revealing compositional vulnerabilities.

3.3 Group-wise Diversity Evaluation

Another crucial criterion for successfully exposing compositional vulnerability is the diversity of generated samples. While repeatedly employing similar and simple attack patterns might boost immediate attack success rates, such approaches are easily defensible and lack generalizability. Indeed, when samples are generated without considering diversity, the attack becomes overly focused on specific distributional weaknesses of the representation, resulting in frequently utilizing a limited set of vocabulary (*e.g.*, man, woman, and vintage in Fig. 8 in Appendix B.3). Therefore, a thorough analysis of pre-trained multimodal representation's compositional vulnerabilities necessitates the construction and utilization of adversarial samples that encompass diverse patterns of text updates, which has largely been overlooked.

To this end, we first construct a set of attribute-enriched tokens that represents a transformation from t_i to $\tilde{t}_i$ through a series of insertion and deletion of words from the Levenshtein distance computation. The token e_i^j is defined as `OP_POS_LEMMA`, where `OP`, `POS`, `LEMMA` corresponds to an "word-level" operation (insertion or deletion), a part-of-speech (POS) tag, and a lemmatized word, respectively (*e.g.*, `I_NOUN_man`). Such tokens distinguish which word-level operations or POS tags as well as words are involved when generating $\tilde{t}_i$ from t_i.

Using a set of attribute-enriched tokens from all data pairs, *i.e.*, $\{\{e_i^j\}_{j=1}^{E_i}\}_{i=1}^{M_{\mathcal{D}}}$, we compute probability distribution of unique tokens with respect to their frequency to obtain entropy $H = -\sum_j p_j \log p_j$, which indicates the extent to which the distribution is spread across different tokens. p_j denotes the probability of a j-th unique token and E_i is the number of tokens for an i-th sample. Note that higher H implies a more diverse set of lexical operations are involved when composing deceptive samples. To prevent pathological cases where the generator might produce arbitrary text to achieve high entropy values, we only consider samples that meet the edit distance criterion (Eq. 3) for diversity evaluation, discarding attribute-enriched tokens from samples that exceed this threshold. This ensures that our diversity metrics reflect meaningful variations in text transformations rather than random deviations from the ground truth.

Since H does not account for how many unique tokens are involved in generation, we also report two additional complementary measures. Following Li et al. (2016) and Zhang et al. (2021), distinct-1 (D_1) captures the ratio of unique tokens out of all tokens. On the other hand, the normalized entropy $\hat{H}$ compromises H and D_1 by normalizing H by the number of unique tokens.

3.4 Threat Model Categorization

In a nutshell, we can categorize the threat model of our framework by following the taxonomy established in adversarial learning (Zhang et al., 2020; Laidlaw et al., 2021; Schwinn et al., 2023; Shayegani et al., 2023b; Vassilev et al., 2024):

- **Model knowledge** - (i) *Gray-box* for cross-modal assessment (*e.g.*, CLIP, LanguageBind); we use only output embeddings with respect to queries without accessing gradients and model parameters. (ii) Black-box for unimodal assessment; we use entailment scores of off-the-shelf NLI models without other information.

- **Attack target** - Untargeted; we induce incorrect predictions instead of eliciting specific responses.
- **Attack granularity** - Mix of word-level and sentence-level perturbation
- **Perturbation constraint** - Distance and auxiliary criteria (§3.2) and diversity evaluation (§3.3) for perceptually plausible attacks
- **Evaluation** - The sample-wise attack success rate and group-wise diversity evaluation
- **Modality** - Language + X, where X can be image, video, and audio
- **Budget** - Number of sampling with LLM (N), which will be further discussed (§4).

4 Approach

4.1 Motivation

Among diverse generators g (*e.g.*, rule-based, human-based, LLM-based) in Table 1, we prioritize LLM-based methods for the following reasons: (1) Rule-based methods (*e.g.*, word swapping) often produce nonsensical and non-fluent text. Additionally, these methods tend to yield simplistic text focused on specific scenarios that models can easily defend against (Hsieh et al., 2023). (2) While human-generated annotations provide fluent text, they are difficult to scale due to resource constraints and the labor-intensive nature of the annotation process. (3) LLMs address these limitations by generating fluent text at scale. Thanks to these advantages, recent multimodal compositionality studies have increasingly adopted LLM-based methods instead of relying on rule-based or human-annotated methods.

4.2 Preliminary: Revealing Compositional Vulnerabilities via Filtering

While attacks in vision-language compositionality literature typically occur only once ($N = 1$), leveraging multiple attempts ($N > 1$) with sample selection could be more effective in revealing such vulnerabilities (Shekhar et al., 2017; Yarom et al., 2023; Park et al., 2022). To incorporate sample selection into MAC, we adopt a Best-of-N strategy—a widely used and general sampling approach—that selects the best sample. Given N samples $\{\tilde{t}_i^n\}_{n=1}^N$, it prioritizes those that meet all sample-wise criteria in Sec. 3.2. If such samples exist, we randomly select from them; otherwise, we sample randomly from the entire set:

$$\mathcal{T}_i = \{\tilde{t}_i^n \mid (s_i^c \cdot s_i^u \cdot s_i^d \cdot s_i^a)(\tilde{t}_i^n, t_i, x_i) = 1\}, \quad (5)$$

$$\tilde{t}_i \sim \begin{cases} \text{Uniform}(\mathcal{T}_i), & \text{if } \mathcal{T}_i \neq \emptyset, \\ \text{Uniform}(\{\tilde{t}_i^n\}_{n=1}^N), & \text{otherwise.} \end{cases} \quad (6)$$

As demonstrated in Table 2, while the filtering approach with $N > 1$ shows improved performance compared to baseline methods, this approach faces several limitations. First, the computational cost scales linearly with N when generating samples for each pair, and the time complexity increases significantly when performed sequentially (see Table 14 in Appendix B.2). Moreover, relying on larger N masks the true effectiveness of adversarial strategies by enabling brute-force attempts. Thus, we limit N to evaluate attack efficiency rather than persistence.

4.3 Self-training

To address the limitations of filtering-based approaches, we propose a learnable method designed to enhance the exposure of compositional vulnerabilities for the first time. Given the absence of annotations or ground truth, we employ self-training (Huang et al., 2023) by promoting responses similar to the condition-satisfying samples generated by the base language model. This approach falls into the category of rejection sampling fine-tuning (RFT) (Touvron et al., 2023). From the training set $\mathcal{D}_{\text{train}} = (t_i, x_i)_{i=1}^{M_{\mathcal{D}_{\text{train}}}}$, we first generate and filter samples $\{\tilde{t}_i\}_{i=1}^{M_{\mathcal{D}_{\text{train}}}}$ using Eq. 6, then only use $M_{\hat{\mathcal{D}}}$ successful adversarial samples to train the model using RFT loss:

$$\{\tilde{t}_i\}_{i=1}^{M_{\hat{\mathcal{D}}}} = \left\{\tilde{t}_i \mid s_i^c \cdot s_i^u \cdot s_i^d \cdot s_i^a = 1\right\}, \quad (7)$$

$$\mathcal{L} = -\frac{1}{M_{\hat{\mathcal{D}}}} \sum_i \sum_j \log g(\tilde{t}_{i,j} | \tilde{t}_{i,<j}, \mathcal{I}, t_i; \Theta), \quad (8)$$

where $\mathcal{I}$ denotes instruction prompt and Θ is a set of learnable parameters of the generator g.

As shown in Table 2, self-training significantly improves the attack success rate by learning to favor samples that effectively attack vulnerabilities with small N (*e.g.*, $N = 4$). To further enhance attack performance beyond naïve self-training, one

Algorithm 1 Diversity-promoting Self-training Data Selection

Require: Set of N samples $\{\tilde{t}_i^n\}_{n=1}^N$ generated for each training instance $i \in [1, M_{\hat{\mathcal{D}}}]$, and diversity function H

Ensure: Diverse successful samples $\{\tilde{t}_i\}_{i=1}^{M_{\hat{\mathcal{D}}}}$

  Initialize $\{\tilde{t}_i\}_{i=1}^{M_{\hat{\mathcal{D}}}}$ randomly from $\{\tilde{t}_i^n | (s_i^c \cdot s_i^u \cdot s_i^d \cdot s_i^a)(\tilde{t}_i^n, t_i, x_i) = 1\}$

  for iteration $k = 1$ to K **do**

    for $i = 1$ to $M_{\hat{\mathcal{D}}}$ **do**

      $\mathcal{T}_i \leftarrow \{\tilde{t}_i^n | (s_i^c \cdot s_i^u \cdot s_i^d \cdot s_i^a)(\tilde{t}_i^n, t_i, x_i) = 1\}$

      $\tilde{t}_i \leftarrow \text{argmax}_{\tilde{t}_i^n \in \mathcal{T}_i} H(\tilde{t}_1, ..., \tilde{t}_i^n, ..., \tilde{t}_{M_{\hat{\mathcal{D}}}})$

    end for

  end for

  return $\{\tilde{t}_i\}_{i=1}^{M_{\hat{\mathcal{D}}}}$

can either train with a larger $N (> 4)$ or iterate self-training as needed. While self-training requires additional computational cost, it can be amortized during inference and leads to more efficient inference by reducing the number of attempts N required to achieve high attack success rates. In our experiments, we set $N = 64$ as the default value for large-N distilled self-training.

4.4 Diversity-promoting Self-training

Although effective at generating successful attacks, self-training tends to generate monotonous samples focused on specific distributional weaknesses rather than maintaining sample diversity, resulting in decreased diversity. The selection of samples involved in training is therefore more important than the training process itself from the perspective of exposing compositional vulnerability. To enhance diversity while maintaining successful attacks, we introduce a Gibbs sampling-based selection process described in Algorithm 1. This approach iteratively selects sample that maximize diversity among successful attacks. While we employ entropy H as a representative diversity metric, it can be substituted with any quantifiable diversity measure (e.g., D_1).

5 Experiments

5.1 Evaluation Protocol

Target representation. We primarily use CLIP (Radford et al., 2021) and LanguageBind (LB) (Zhu et al., 2024) as target multimodal representations. They are representative models with dual-modality and multi-modality pre-training. Additionally, to analyze the transferability of deception across different representations, we also evaluate SigLIP (Zhai et al., 2023), NegCLIP (Yuksekgonul et al., 2022), and BLIP (Li et al., 2022).

Sample generation. Our methodology operates by modifying text (Sec. 3.1). We generate samples that reveal compositional vulnerability using representative multimodal datasets: COCO (Lin et al., 2014) for image, MSRVTT (Xu et al., 2016) for video, and AudioCaps (Kim et al., 2019) for audio.

Unless mentioned otherwise, we use Llama-3.1-8B (Dubey et al., 2024) for sample generation and self-training. We explore its applicability across different LLMs, including GPT-4o (Achiam et al., 2023), noting that larger or proprietary models do not necessarily lead to more effective deception, as discussed in Appendix B.1. We employ two instruction prompts (*i.e.*, $\mathcal{I}$ in Eq. 8). The *deceptive-general prompt* instructs to expose vulnerability without constraints on text updates, while the *deceptive-specific prompt* instructs to perform text updates corresponding to `replace`, `swap`, and `add` based on taxonomy from existing literature, as in Table 1. See Appendix A.3 for prompt demonstrations. For better performance, we primarily use the general prompt.

Evaluation metrics. We conduct sample-wise and group-wise evaluations as described in Sec. 3. For sample-wise evaluation, we report the attack success rate (ASR) focusing on crossmodal criterion (*Cross*) and all criteria (*Total*), while for group-wise diversity evaluation, we report entropy (H) and distinct-1 (D_1). Fine-grained performance comparisons are discussed in Appendix B.4.

Baselines. We establish a set of competitive baselines using existing compositionality frameworks. For models generating with $N = 1$ budget, we utilize RoCOCO (Park et al., 2024), SugarCrepe (Hsieh et al., 2023), LLaVA-Score (Li et al., 2024b), TripletClip (Patel et al., 2024), and VideoCon (Bansal et al., 2024). For filtering-based models, we employ SeeTrue (Yarom et al., 2023), VFC (Momeni et al., 2023), and CompA (Ghosh et al., 2024), using $N = 4$ for inference. For the studies that use proprietary models like GPT-4, we substitute Llama-3.1-8B for it and modify the prompts to ensure effective sample generation with this model for fair comparison and cost constraints. For experimental details, see Appendix A.

5.2 Experimental Results

Table 2 summarizes the overall results, showing our approach outperforms prior methods in both ASR and diversity. As evident from RoCOCO's first

Method	(a) Image (CLIP/COCO) ASR↑		(a) Diversity↑		(b) Video (LB/MSRVTT) ASR↑		(b) Diversity↑		(c) Audio (LB/AudioCaps) ASR↑		(c) Diversity↑	
	Cross	Total	H	D_1	Cross	Total	H	D_1	Cross	Total	H	D_1
N=1												
$\text{RoCOCO}_{\text{rand-voca}}$ (Park et al., 2024)	24.33	1.99	7.642	**0.196**	-	-	-	-	-	-	-	-
$\text{RoCOCO}_{\text{Danger}}$ (Park et al., 2024)	20.24	7.88	4.454	0.052	-	-	-	-	-	-	-	-
$\text{RoCOCO}_{\text{same-concept}}$ (Park et al., 2024)	17.09	5.29	7.098	0.088	-	-	-	-	-	-	-	-
$\text{RoCOCO}_{\text{diff-concept}}$ (Park et al., 2024)	17.92	2.75	7.128	0.089	-	-	-	-	-	-	-	-
SugarCrepe* (Hsieh et al., 2023)	10.84	2.40	7.312	0.103	-	-	-	-	-	-	-	-
LLaVA-Score* (Li et al., 2024b)	24.81	5.71	7.201	0.110	-	-	-	-	-	-	-	-
TripletCLIP (Patel et al., 2024)	12.81	6.34	7.551	0.092	-	-	-	-	-	-	-	-
VideoCon* (Bansal et al., 2024)	-	-	-	-	16.30	7.10	6.702	0.610	-	-	-	-
Deceptive-General Prompt (zero-shot)	28.52	6.88	7.562	0.131	32.20	7.70	6.809	0.638	28.68	10.47	6.572	0.182
N=4												
SeeTrue (Yarom et al., 2023)	34.67	23.33	7.168	0.124	-	-	-	-	-	-	-	-
VFC* (Momeni et al., 2023)	-	-	-	-	42.60	36.90	5.929	0.381	-	-	-	-
CompA* (Ghosh et al., 2024)	-	-	-	-	-	-	-	-	49.38†	5.76†	6.009†	0.171†
Deceptive-General Prompt (zero-shot)	37.29	19.19	7.571	0.130	42.40	24.80	6.808	0.626	42.60	29.02	6.566	0.172
+ Self-Train	43.08	34.64	7.507	0.120	48.90	39.70	6.900	0.587	55.37	47.35	6.472	0.157
+ Self-Train + Large-N Distilled	**48.29**	42.03	7.452	0.117	52.90	44.20	6.839	0.594	58.38	51.57	6.508	0.157
+ Self-Train + Large-N Distilled + Diversity-Promoted (**Ours**)	47.93	**42.10**	**7.747**	0.129	**53.50**	**45.60**	**7.125**	**0.667**	**60.25**	**52.87**	**6.868**	**0.191**

Table 2: Main Results. '-' indicates that the method is not applicable. (*: the prompts from the original papers are slightly modified. †: the results are computed for a subset to which the method can be applied).

$\text{ASR}_{\text{Total}}$	CLIP	SigLIP	NegCLIP	BLIP
CLIP	42.10 (+22.91)	28.63 (+15.68)	24.84 (+12.71)	25.25 (+14.13)
SigLIP	29.37 (+16.13)	41.04 (+21.32)	23.84 (+12.17)	25.01 (+13.76)
NegCLIP	25.40 (+12.68)	23.63 (+11.47)	40.81 (+20.10)	23.77 (+12.33)
BLIP	19.84 (+10.60)	19.11 (+10.04)	18.02 (+8.94)	32.50 (+17.80)

Table 3: Cross-model transfer analysis ($N = 4$). Columns are source models for filtering, and rows are target models for evaluation. Numbers in parentheses are absolute gains from our proposed self-training compared to the zero-shot baselines.

two variants, there exists a trade-off where maximizing ASR leads to a sharp decline in diversity and vice versa, indicating that focusing on either metric alone is far from optimal. Generating multiple samples and applying filtering improves ASR across all modalities compared to $N = 1$, though this does not translate to enhanced diversity. See Appendix B.3 (Fig. 8) for qualitative distribution in terms of diversity.

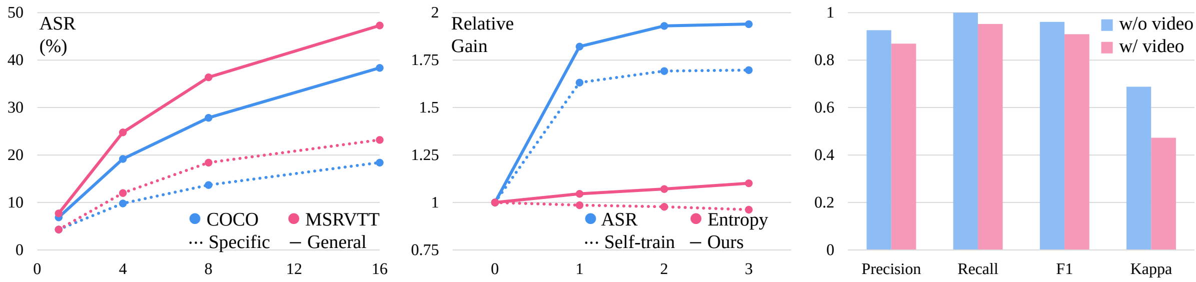
The last four rows reveal the ablation study of our method. Using only the deceptive-general prompt yields performance comparable to existing methods. Adding self-training for a single iteration dramatically increases ASR, *i.e.*, +68% on average, underscoring its role in addressing compositionality. Yet, this alone does not enhance diversity and may even reduce it. This implies naïve self-training, while effective for ASR, falls short in diverse exposure of compositional vulnerability. Instead, incorporating diversity-promoting filtering leads to consistent improvements in both diversity metrics without sacrificing ASR (+2%), advancing the pareto front in the attack-diversity trade-off.

Table 3 examines the transferability of deceptive samples across multimodal representations. The results show high transferability, often exceeding the best performing baseline (23.33). Notably, the performance gains from self-training are substantial across all settings, achieving 2.1× improvement on average. BLIP shows slightly lower performance presumably due to its use of yes/no classification logits instead of embedding similarity.

5.3 Performance Analysis

General vs. specific prompt. As summarized in Table 1, various compositionality frameworks employ either general or specific types of prompts, necessitating an analysis of their effectiveness in ASR. Fig. 3-(a) compares performance under different instruction types for generation budget N. Methods without specific text update constraints consistently outperform constrained ones, with this trend persisting as N increases. Notably, our self-training approach with $N = 4$ matches the performance of non-self-training methods with an $N = 16$ budget.

Influence of multi-round self-training. Self-training enables multiple iterations by refining filtering models across training rounds. Fig. 3-(b) shows the relative gains of diversity-promoting vs. naïve self-training on AudioCaps. Our self-training significantly improves ASR performance, reaching saturation by the third round. While entropy degrades with conventional self-training, our approach sustains continuous improvement. For



(a) Influence of text update types (b) Influence of multi-round self-training (c) Human evaluation

Figure 3: Analysis of our proposed framework. Please refer to Sec. 5.3 for detailed explanation.

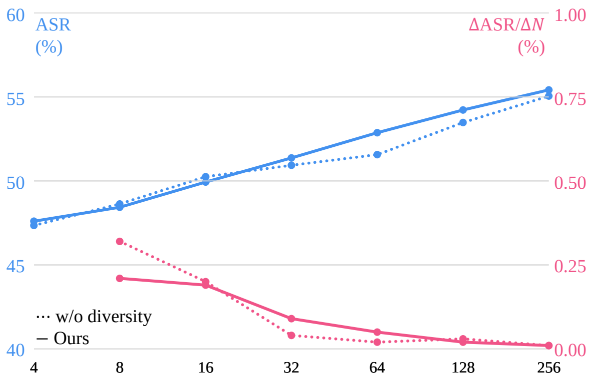


Figure 4: Influence of N in self-training.

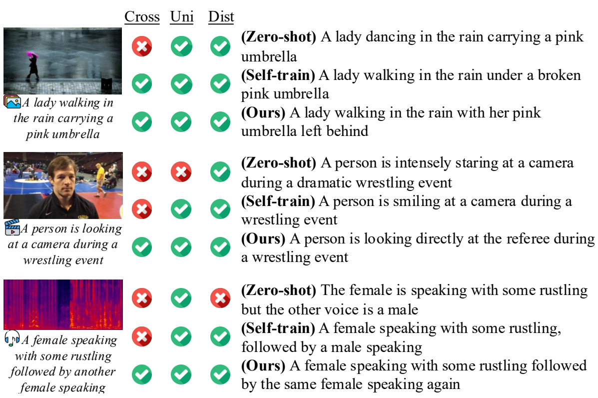


Figure 5: Qualitative examples from COCO, MSRVTT, and AudioCaps datasets (from top to bottom).

MSRVTT results, please refer to Appendix B.5.

Influence of large N in self-training. To better understand the influence of N in distillation-based self-training, we report the ASR of our method using AudioCaps in Fig. 4. While increasing N does not display a clear signal of saturation, the relative performance gain with respect to N ($\Delta\mathrm{ASR}/\Delta N$) does. This diminishing return suggests that $N = 64$ offers a reasonable balance between performance improvement and time constraint.

Human evaluation. A potential limitation is our reliance on the model-based unimodal entailment assessment, necessitating evaluation on human agreement. Fig. 3-(c) compares our criterion against human evaluation by five annotators on 50 random MSRVTT test samples. Results show high agreement (F1 > 0.9) regardless of video presence, with moderate to substantial inter-annotator agreement κ (Fleiss, 1971). Although κ is slightly lower for evaluations with videos—likely due to subjective interpretation of longer contexts—overall agreement remains strong (F1 = 0.9091), confirming the reliability of our unimodal assessment.

Qualitative examples. Fig. 5 compares generated samples from variants of our method across different modalities. Compared to other variants, our self-training successfully applies various modification without being constrained to specific patterns. Additional examples are provided in Appendix B.9.

6 Conclusion

We explored the compositional vulnerability of pretrained multimodal representations using LLMs. First, we established a testbed by proposing **MAC**, which provides a comprehensive set of criteria for evaluating how effectively and diversely a target representation can be deceived. Furthermore, we suggested the application of self-training to multimodal compositionality for the first time via iterative RFT with diversity-promoting filtering to improve both ASR and diversity. Lastly, our modality-agnostic assessment allowed for a thorough analysis of compositional vulnerabilities across image, video, and audio modalities, where our method consistently outperformed prior arts across various target representations. Our benchmark's modality-agnostic design opens avenues for extending vulnerability analysis to less-explored modalities like IMU or tactile sensing, even in the absence of multimodal LLMs capable of processing these data types.

Limitations

Our work focused on short captions in exploring multimodal adversarial compositionality. Extending MAC (*i.e.*, deceiving pre-trained multimodal representations) to longer, detailed captions (Onoe et al., 2024; Chen et al., 2024) represents a distinct but promising research direction, as it would require more sophisticated attack strategies that consider long-range dependencies and contextual relationships throughout the caption to successfully deceive target representations.

Ethics Statement

Since our work uses language models to generate adversarial captions to reveal compositional vulnerabilities, they might potentially generate biased or toxic content. We encourage practitioners who wish to use generated captions to carefully monitor and filter outputs to prevent unintended harmful content.

For human evaluation, we worked with annotators primarily from the US, UK, Canada, New Zealand, and Australia, ensuring fair compensation above their local minimum wages (averaging $18 per hour). Please refer to Appendix A.5 for details.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No. RS-2019-II191082, RS-2021-II211343, No. RS-2022-II220156), the National Research Foundation of Korea (NRF) grant (No. 2023R1A2C2005573), and the IITP-ITRC (Information Technology Research Center) grant (IITP-2025-RS-2024-00437633) funded by the Korea government (MSIT). Gunhee Kim is the corresponding author.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv:2303.08774*.

Jaewoo Ahn, Yeda Song, Sangdoo Yun, and Gunhee Kim. 2023. MPCHAT: Towards multimodal persona-grounded conversation. In *ACL*.

Alexandr Andoni and Negev Shekel Nosatzki. 2020. Edit distance in near-linear time: It's a constant factor. In *FOCS*.

Eugene Bagdasaryan, Rishi Jha, Vitaly Shmatikov, and Tingwei Zhang. 2024. Adversarial illusions in {Multi-Modal} embeddings. In *USENIX Security*.

Hritik Bansal, Yonatan Bitton, Idan Szpektor, Kai-Wei Chang, and Aditya Grover. 2024. Videocon: Robust video-language alignment via contrast captions. In *CVPR*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2024. Sharegpt4v: Improving large multi-modal models with better captions. In *ECCV*.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *CVPR*.

Sivan Doveh, Assaf Arbelle, Sivan Harary, Eli Schwartz, Roei Herzig, Raja Giryes, Rogerio Feris, Rameswar Panda, Shimon Ullman, and Leonid Karlinsky. 2023. Teaching structured vision & language concepts to vision & language models. In *CVPR*.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. 2020. Clotho: An audio captioning dataset. In *ICASSP*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv:2407.21783*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal transformer for video retrieval. In *ECCV*.

Sensen Gao, Xiaojun Jia, Xuhong Ren, Ivor Tsang, and Qing Guo. 2024. Boosting transferability in vision-language attacks via diversification along the intersection region of adversarial trajectory. In *ECCV*.

Sreyan Ghosh, Ashish Seth, Sonal Kumar, Utkarsh Tyagi, Chandra Kiran Reddy Evuru, S Ramaneswaran, S Sakshi, Oriol Nieto, Ramani Duraiswami, and Dinesh Manocha. 2024. Compa: Addressing the gap in compositional reasoning in audio-language models. In *ICLR*.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *EMNLP*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *ICLR*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. 2023. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *NeurIPS*.

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 *ICLR*.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *EMNLP*.

Andrej Karpathy and Li Fei-Fei. 2017. Deep visual-semantic alignments for generating image descriptions. *TPAMI*.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. 2019. Audiocaps: Generating captions for audios in the wild. In *NAACL*.

Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *CVPR*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*.

Chun-Yi Kuan and Hung-yi Lee. 2025. Can large audio-language models truly hear? tackling hallucinations with multi-task assessment and stepwise audio reasoning. In *ICASSP*.

Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. 2021. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. 2024a. Naturalbench: Evaluating vision-language models on natural adversarial samples. In *NeurIPS Datasets and Benchmarks*.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and William B Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *NAACL*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*.

Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. 2023a. Unmasked teacher: Towards training-efficient video foundation models. In *ICCV*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. In *EMNLP*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *EMNLP*.

Yuheng Li, Haotian Liu, Mu Cai, Yijun Li, Eli Shechtman, Zhe Lin, Yong Jae Lee, and Krishna Kumar Singh. 2024b. Removing distributional discrepancies in captions improves image-text alignment. In *ECCV*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *CVPR*.

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

Jingzhou Liu, Wenhu Chen, Yu Cheng, Zhe Gan, Licheng Yu, Yiming Yang, and Jingjing Liu. 2020. Violin: A large-scale dataset for video-and-language inference. In *CVPR*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omar Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.

Dong Lu, Zhiqiang Wang, Teng Wang, Weili Guan, Hongchang Gao, and Feng Zheng. 2023. Set-level guidance attack: Boosting adversarial transferability of vision-language pre-training models. In *ICCV*.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2022. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*.

Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. 2023. Crepe: Can vision-language foundation models reason compositionally? In *CVPR*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2024. Tree of attacks: Jailbreaking black-box llms automatically. In *NeurIPS*.

Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. 2023. Verbs in action: Improving verb understanding in video-language models. In *ICCV*.

Youngtaek Oh, Jae Won Cho, Dong-Jin Kim, In So Kweon, and Junmo Kim. 2024. Preserving multi-modal capabilities of pre-trained vlms for improving vision-linguistic compositionality. In *EMNLP*.

Andreea-Maria Oncescu, A. Sophia Koepke, João F. Henriques, Zeynep Akata, and Samuel Albanie. 2021. Audio retrieval with natural language queries. In *INTERSPEECH*.

Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, Su Wang, and Jason Baldridge. 2024. Docci: Descriptions of connected and contrasting images. In *ECCV*.

Rafail Ostrovsky and Yuval Rabani. 2007. Low distortion embeddings for edit distance. *JACM*.

Jae Sung Park, Sheng Shen, Ali Farhadi, Trevor Darrell, Yejin Choi, and Anna Rohrbach. 2022. Exposing the limits of video-text models through contrast sets. In *NAACL*.

Seulki Park, Daeho Um, Hajung Yoon, Sanghyuk Chun, and Sangdoo Yun. 2024. Rococo: Robustness benchmark of ms-coco to stress-test image-text matching models. In *ECCV Workshop*.

Maitreya Patel, Sheng Cheng, Changhoon Kim, Tejas Gokhale, Chitta Baral, Yezhou Yang, et al. 2024. Tripletclip: Improving compositional reasoning of clip via synthetic vision-language negatives. In *NeurIPS*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv:2204.06125*.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. 2024. Vision-language models are zero-shot reward models for reinforcement learning. In *ICLR*.

Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. 2017. Movie description. *IJCV*.

Leo Schwinn, David Dobre, Stephan Günnemann, and Gauthier Gidel. 2023. Adversarial attacks and defenses in large language models: Old and new threats. In *NeurIPS Workshop*.

Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. 2023a. Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. In *ICLR*.

Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. 2023b. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv:2310.10844*.

Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurelie Herbelot, Moin Nabi, Enver Sangineto, Raffaella Bernardi, et al. 2017. Foil it! find one mismatch between image and language caption. In *ACL*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *NeurIPS*.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, et al. 2024. Beyond human data: Scaling self-training for problem-solving with language models. *TMLR*.

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR*.

Chia-Wei Tang, Ting-Chih Chen, Kiet A. Nguyen, Kazi Sajeed Mehrab, Alvi Md Ishmam, and Chris Thomas. 2024. M3D: MultiModal MultiDocument fine-grained inconsistency detection. In *EMNLP*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv:2408.00118*.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*.

Apostol Vassilev, Alina Oprea, Alie Fordyce, and Hyrum Anderson. 2024. Adversarial machine learning: A taxonomy and terminology of attacks and mitigations. Technical report, National Institute of Standards and Technology.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2023. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*.

Wenzhuo Xu, Kai Chen, Ziyi Gao, Zhipeng Wei, Jingjing Chen, and Yu-Gang Jiang. 2024. Highly transferable diffusion-based unrestricted adversarial attack on pre-trained vision-language models. In *ACM MM*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2. 5 technical report. *arXiv:2412.15115*.

Haozhe Yang, Yuhan Xiang, Ke Sun, Jianlong Hu, and Xianming Lin. 2024b. Towards video-text retrieval adversarial attack. In *ICASSP*.

Michal Yarom, Yonatan Bitton, Soravit Changpinyo, Roee Aharoni, Jonathan Herzig, Oran Lang, Eran Ofek, and Idan Szpektor. 2023. What you see is what you read? improving text-image alignment evaluation. *NeurIPS*.

Ziyi Yin, Muchao Ye, Tianrong Zhang, Tianyu Du, Jinguo Zhu, Han Liu, Jinghui Chen, Ting Wang, and Fenglong Ma. 2023. Vlattack: Multimodal adversarial attacks on vision-language tasks via pre-trained models. *NeurIPS*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*.

Youngjae Yu, Jiwan Chung, Heeseung Yun, Jack Hessel, Jae Sung Park, Ximing Lu, Rowan Zellers, Prithviraj Ammanabrolu, Ronan Le Bras, Gunhee Kim, and Yejin Choi. 2023a. Fusing pre-trained language models with multimodal prompts through reinforcement learning. In *CVPR*.

Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018. A joint sequence fusion model for video question answering and retrieval. In *ECCV*.

Zhen Yu, Zhou Qin, Zhenhua Chen, Meihui Lian, Haojun Fu, Weigao Wen, Hui Xue, and Kun He. 2023b. Sparse black-box multimodal attack for vision-language adversary generation. In *EMNLP Findings*.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2022. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STar: Bootstrapping reasoning with reasoning. In *NeurIPS*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *ICCV*.

Bowen Zhang, Hexiang Hu, and Fei Sha. 2018. Cross-modal and hierarchical modeling of video and text. In *ECCV*.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *HumEval*.

Jiaming Zhang, Qi Yi, and Jitao Sang. 2022. Towards adversarial attack on vision-language pre-training models. In *ACM MM*.

Jianrui Zhang, Mu Cai, and Yong Jae Lee. 2024. Vinoground: Scrutinizing lmms over dense temporal reasoning with short videos. *arXiv:2410.02763*.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM TIST*.

Tiancheng Zhao, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. 2022. Vl-checklist: Evaluating pre-trained vision-language models with objects, attributes and relations. In *EMNLP*.

Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. 2024. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *ICLR*.

A Experimental Details

A.1 Dataset

We used standard train and test sets commonly employed in multimodal retrieval tasks as follows.

For COCO (Lin et al., 2014), we adopt the Karpathy test split (Karpathy and Fei-Fei, 2017) as the test set, which consists of 5,000 images paired with 25,010 captions. The train set corresponds to the COCO 2014 train split, containing 83,287 images and 414,113 captions. For MSRVTT (Xu et al., 2016), we utilize the MSRVTT 1K-A split (Yu et al., 2018) as the test set, which includes 1,000 videos, each associated with a single caption. The train set corresponds to the MSRVTT 9K train split, containing 9,000 videos with 180,000 captions. For AudioCaps (Kim et al., 2019), we use the test split from Oncescu et al. (2021), which consists of 816 audio clips with 4,080 captions. The train set corresponds to the train split from Oncescu et al. (2021), which includes 49,291 audio clips, each paired with a single caption. All datasets contain English language captions and are publicly available, used in accordance with their respective licenses for research purposes.

Note that each train set (x_i, t_i) does not include a label for deceptive caption supervision. This absence of supervision serves as the primary motivation for our self-training approach, which aims to generate deceptive captions $\tilde{t}_i$.

A.2 Models

Target models. For target pre-trained multimodal representations for evaluating crossmodal criterion in Sec. 3.2, we utilize: CLIP[2], SigLIP[3], NegCLIP[4] BLIP[5], LanguageBind$_{\text{Video}}$[6], and LanguageBind$_{\text{Audio}}$[7].

NLI models. For NLI models for evaluating the unimodal criterion in Sec. 3.2, we utilize: RoBERTa[8], DeBERTa[9], and BART[10].

LLMs. For LLMs, we use: Llama-3.1-8B[11], Llama-3.1-70B (Q4_0)[12], Qwen-2.5-7B[13], Gemma-2-9B[14], and GPT-4o$_{\texttt{2024-08-06}}$. Here, Q4_0 denotes a 4-bit quantized version of the model.

A.3 Prompt Demonstration

Deceptive-General Prompt. The deceptive-general prompt is presented in Table 4.

Deceptive-Specific Prompt. The deceptive-specific prompts, tailored for different modification types, are presented as follows:

- **Replacement Prompts:**
  - Table 5: Replacing objects.
  - Table 6: Replacing attributes.
  - Table 7: Replacing relationships.
  - Table 8: Replacing numerical counts.
- **Addition Prompts:**
  - Table 9: Adding objects.
  - Table 10: Adding attributes.
- **Swap Prompts:**
  - Table 11: Swapping objects.
  - Table 12: Swapping attributes.

Deceptive-General Prompt
You will be given a caption describing the {contents_modality}. Your task is to generate a hard negative caption using the criteria below: *** [Generation Criteria] 1. Ensure the new caption has higher similarity to the {contents_modality} in {contents_modality}-text crossmodal model than the given caption. 2. Introduce a contradiction compared to the given caption, but avoid simple negations (e.g., using words like "no", "not", "empty", or "without"). 3. Make fewer than {max_word_distance_plus_one} word-level changes (add, delete, or substitute words) to the given caption without fully rewriting it to generate the new caption. [Given Caption] - {caption} *** Write only the new caption starting with "Generated Caption: ", without explanation.

Table 4: Deceptive-general prompt.

A.4 Implementation Details

For generating new captions with LLMs, we apply nucleus sampling (Holtzman et al., 2020) with $p = 0.95$ and a temperature of $\tau = 0.7$ across

[2] laion/CLIP-ViT-H-14-laion2B-s32B-b79K
[3] google/siglip-so400m-patch14-384
[4] https://github.com/mertyg/vision-language-models-are-bows
[5] Salesforce/blip-itm-base-coco
[6] LanguageBind/LanguageBind_Video_FT
[7] LanguageBind/LanguageBind_Audio_FT
[8] FacebookAI/roberta-large-mnli
[9] microsoft/deberta-xlarge-mnli
[10] facebook/bart-large-mnli
[11] meta-llama/Meta-Llama-3.1-8B-Instruct
[12] Ollama Llama-3.1-70B (Q4_0)
[13] Qwen/Qwen2.5-7B-Instruct
[14] google/gemma-2-9b-it

Deceptive-Specific Prompt (`replace-object`)

You will be given a caption describing the {contents_modality}. Your task is to generate a hard negative caption based on the "object replacement" scenario using the criteria below:

***
[Generation Criteria]
1. Replace a key object in the given caption with a new object that is not in the given caption.
2. Ensure the new caption has higher similarity to the {contents_modality} in {contents_modality}-text crossmodal model than the given caption.
3. Introduce a contradiction compared to the given caption, but avoid simple negations (e.g., using words like "no", "not", "empty", or "without").
4. Make fewer than {max_word_distance_plus_one} word-level changes (add, delete, or substitute words) to the given caption without fully rewriting it to generate the new caption.

[Given Caption]
- {caption}
***

Write only the new caption starting with "Generated Caption: ", without explanation.

Table 5: Deceptive-specific prompt (`replace-object`).

Deceptive-Specific Prompt (`replace-attribute`)

You will be given a caption describing the {contents_modality}. Your task is to generate a hard negative caption based on the "attribute replacement" scenario using the criteria below:

***
[Generation Criteria]
1. Replace an adjective word in the given caption with a new adjective word that is not in the given caption.
2. Ensure the new caption has higher similarity to the {contents_modality} in {contents_modality}-text crossmodal model than the given caption.
3. Introduce a contradiction compared to the given caption, but avoid simple negations (e.g., using words like "no", "not", "empty", or "without").
4. Make fewer than {max_word_distance_plus_one} word-level changes (add, delete, or substitute words) to the given caption without fully rewriting it to generate the new caption.

[Given Caption]
- {caption}
***

Write only the new caption starting with "Generated Caption: ", without explanation.

Table 6: Deceptive-specific prompt (`replace-attribute`).

Deceptive-Specific Prompt (`replace-relation`)

You will be given a caption describing the {contents_modality}. Your task is to generate a hard negative caption based on the "relation replacement" scenario using the criteria below:

***
[Generation Criteria]
1. Replace an action or a spatial relationship in the given caption with a new action or spatial relationship that is not in the given caption.
2. Ensure the new caption has higher similarity to the {contents_modality} in {contents_modality}-text crossmodal model than the given caption.
3. Introduce a contradiction compared to the given caption, but avoid simple negations (e.g., using words like "no", "not", "empty", or "without").
4. Make fewer than {max_word_distance_plus_one} word-level changes (add, delete, or substitute words) to the given caption without fully rewriting it to generate the new caption.

[Given Caption]
- {caption}
***

Write only the new caption starting with "Generated Caption: ", without explanation.

Table 7: Deceptive-specific prompt (`replace-relation`).

Deceptive-Specific Prompt (`replace-count`)

You will be given a caption describing the {contents_modality}. Your task is to generate a hard negative caption based on the "counting replacement" scenario using the criteria below:

***
[Generation Criteria]
1. Replace the numerical count of a key object in the given caption (e.g., from "two" to "three").
2. Ensure the new caption has higher similarity to the {contents_modality} in {contents_modality}-text crossmodal model than the given caption.
3. Introduce a contradiction compared to the given caption, but avoid simple negations (e.g., using words like "no", "not", "empty", or "without").
4. Make fewer than {max_word_distance_plus_one} word-level changes (add, delete, or substitute words) to the given caption without fully rewriting it to generate the new caption.

[Given Caption]
- {caption}
***

Write only the new caption starting with "Generated Caption: ", without explanation.

Table 8: Deceptive-specific prompt (`replace-count`).

Deceptive-Specific Prompt (`add-object`)

You will be given a caption describing the {contents_modality}. Your task is to generate a hard negative caption based on the "object addition" scenario using the criteria below:

***
[Generation Criteria]
1. Generate a new plausible but uncommon object that's not in the given caption, and then add the new object to make a new caption.
2. Ensure the new caption has higher similarity to the {contents_modality} in {contents_modality}-text crossmodal model than the given caption.
3. Introduce a contradiction compared to the given caption, but avoid simple negations (e.g., using words like "no", "not", "empty", or "without").
4. Make fewer than {max_word_distance_plus_one} word-level changes (add, delete, or substitute words) to the given caption without fully rewriting it to generate the new caption.

[Given Caption]
- {caption}
***

Write only the new caption starting with "Generated Caption: ", without explanation.

Table 9: Deceptive-specific prompt (`add-object`).

Deceptive-Specific Prompt (`add-attribute`)

You will be given a caption describing the {contents_modality}. Your task is to generate a hard negative caption based on the "attribute addition" scenario using the criteria below:

***
[Generation Criteria]
1. Add a new plausible but uncommon attribute for the object in the given caption.
2. Ensure the new caption has higher similarity to the {contents_modality} in {contents_modality}-text crossmodal model than the given caption.
3. Introduce a contradiction compared to the given caption, but avoid simple negations (e.g., using words like "no", "not", "empty", or "without").
4. Make fewer than {max_word_distance_plus_one} word-level changes (add, delete, or substitute words) to the given caption without fully rewriting it to generate the new caption.

[Given Caption]
- {caption}
***

Write only the new caption starting with "Generated Caption: ", without explanation.

Table 10: Deceptive-specific prompt (`add-attribute`).

Deceptive-Specific Prompt (`swap-object`)

You will be given a caption describing the {contents_modality}. Your task is to generate a hard negative caption based on the "object swapping" scenario using the criteria below:

***
[Generation Criteria]
1. First locate two swappable nouns in the given caption, and then swap them to make a new caption (e.g., from "woman looking at elephant" to "elephant looking at woman")
2. Ensure the new caption has higher similarity to the {contents_modality} in {contents_modality}-text crossmodal model than the given caption.
3. Introduce a contradiction compared to the given caption, but avoid simple negations (e.g., using words like "no", "not", "empty", or "without").
4. Make fewer than {max_word_distance_plus_one} word-level changes (add, delete, or substitute words) to the given caption without fully rewriting it to generate the new caption.

[Given Caption]
- {caption}
***

Write only the new caption starting with "Generated Caption: ", without explanation.

Table 11: Deceptive-specific prompt (`swap-object`).

Deceptive-Specific Prompt (`swap-attribute`)

You will be given a caption describing the {contents_modality}. Your task is to generate a hard negative caption based on the "attribute swapping" scenario using the criteria below:

***
[Generation Criteria]
1. First locate two swappable adjectives in the given caption describing different objects, and then swap them to make a new caption (e.g., from "a red apple and a purple grape" to "a purple apple and a red grape").
2. Ensure the new caption has higher similarity to the {contents_modality} in {contents_modality}-text crossmodal model than the given caption.
3. Introduce a contradiction compared to the given caption, but avoid simple negations (e.g., using words like "no", "not", "empty", or "without").
4. Make fewer than {max_word_distance_plus_one} word-level changes (add, delete, or substitute words) to the given caption without fully rewriting it to generate the new caption.

[Given Caption]
- {caption}
***

Write only the new caption starting with "Generated Caption: ", without explanation.

Table 12: Deceptive-specific prompt (`swap-attribute`).

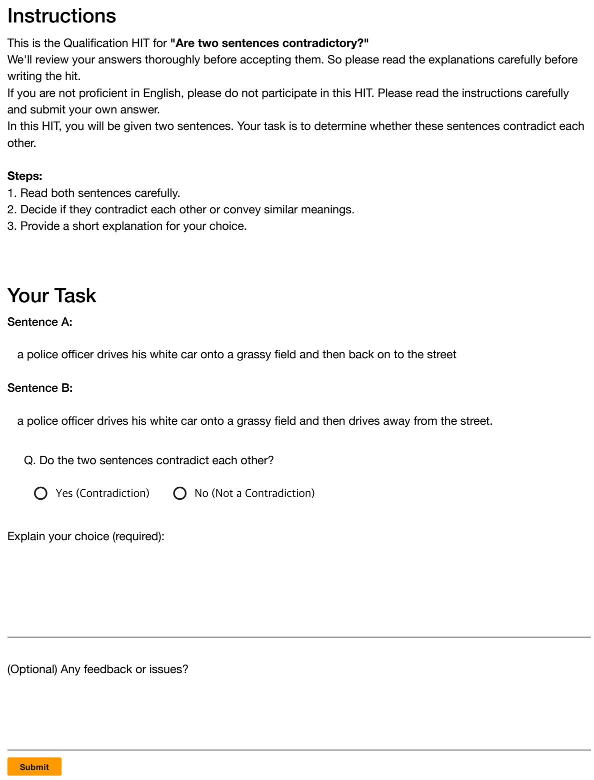


Figure 6: User interface for human evaluation: Task 1 (without video).

all LLMs, except for GPT-4o, where we use the default hyperparameters provided by the OpenAI API. For self-training LLMs, we use a batch size of 16, a LoRA (Hu et al., 2022) rank of 16, a LoRA alpha of 32, and a learning rate of 2×10^{-4}. Each LLM is trained for 3 epochs per round. During multi-round training, we reset the LLM to its original checkpoint at the start of each round, rather than continuing from the last checkpoint, to mitigate overfitting (Zelikman et al., 2022; Singh et al., 2024). All experiments are conducted on a single NVIDIA RTX A6000 GPU. All reported results are based on a single run per experiment.

A.5 Human Evaluation

We provide a detailed explanation of the human evaluation process described in Sec. 5.3 (Fig. 3-(c)). Two user interfaces were designed for evaluation on Amazon Mechanical Turk (AMT): one without video input (Fig. 6) and one with video input from MSRVTT (Fig. 7). For each data point, we collected five annotations to ensure reliability. To maintain annotation quality, annotators were required to provide a short explanation for their responses. Additionally, we ensured that AMT workers were fairly compensated at approximately \$18 per hour (\$0.5 per HIT).

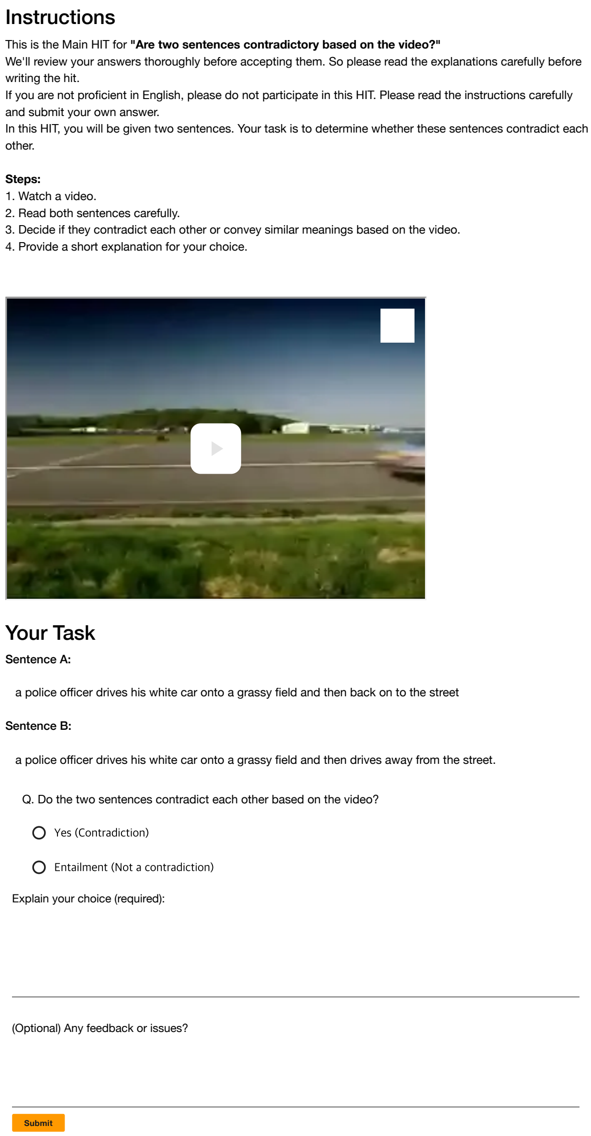


Figure 7: User interface for human evaluation: Task 2 (with video).

Method	ASR↑		Diversity↑	
	Cross	Total	H	D_1
Qwen-2.5-7B	18.80	4.50	6.454	0.538
Llama-3.1-8B	**32.20**	7.70	**6.809**	**0.638**
Gemma-2-9B	19.80	8.30	6.472	0.507
Llama-3.1-70B	20.80	9.10	6.416	0.520
GPT-4o$_{\texttt{2024-08-06}}$	21.10	**14.40**	6.440	0.502

Table 13: Attacking LanguageBind in MSRVTT test set with diverse LLMs (N=1). All LLMs use the deceptive-general prompt.

B Further Analyses

B.1 MAC Performance Across LLMs

We examine the applicability across different language models, such as Qwen 2.5 (Yang et al., 2024a) and Gemma 2 (Team et al., 2024), as well as GPT-4o (Achiam et al., 2023). As shown in Table 13, larger or proprietary models do not necessarily lead to more effective deception. For instance, while GPT-4o achieves the highest ASR, its diversity is lower than that of Llama-3.1-8B. Moreover, Llama-3.1-8B with $N = 4$ achieves a significantly higher ASR (24.80 in Table 2) compared to GPT-4o (14.40). This suggests that using a smaller model with a Best-of-$N(> 1)$ approach is more effective than relying on a proprietary model with a budget of $N = 1$.

Method		ASR↑		Diversity↑	
	Time	Cross	Total	H	D_1
$N = 4$					
Sequential	$O(N)$	38.50	20.10	**6.809**	0.658
Parallel	$O(1)$	42.40	24.80	6.808	0.626
$N = 8$					
Sequential	$O(N)$	45.40	28.50	6.764	**0.675**
Parallel	$O(1)$	**49.20**	**36.40**	6.773	0.601

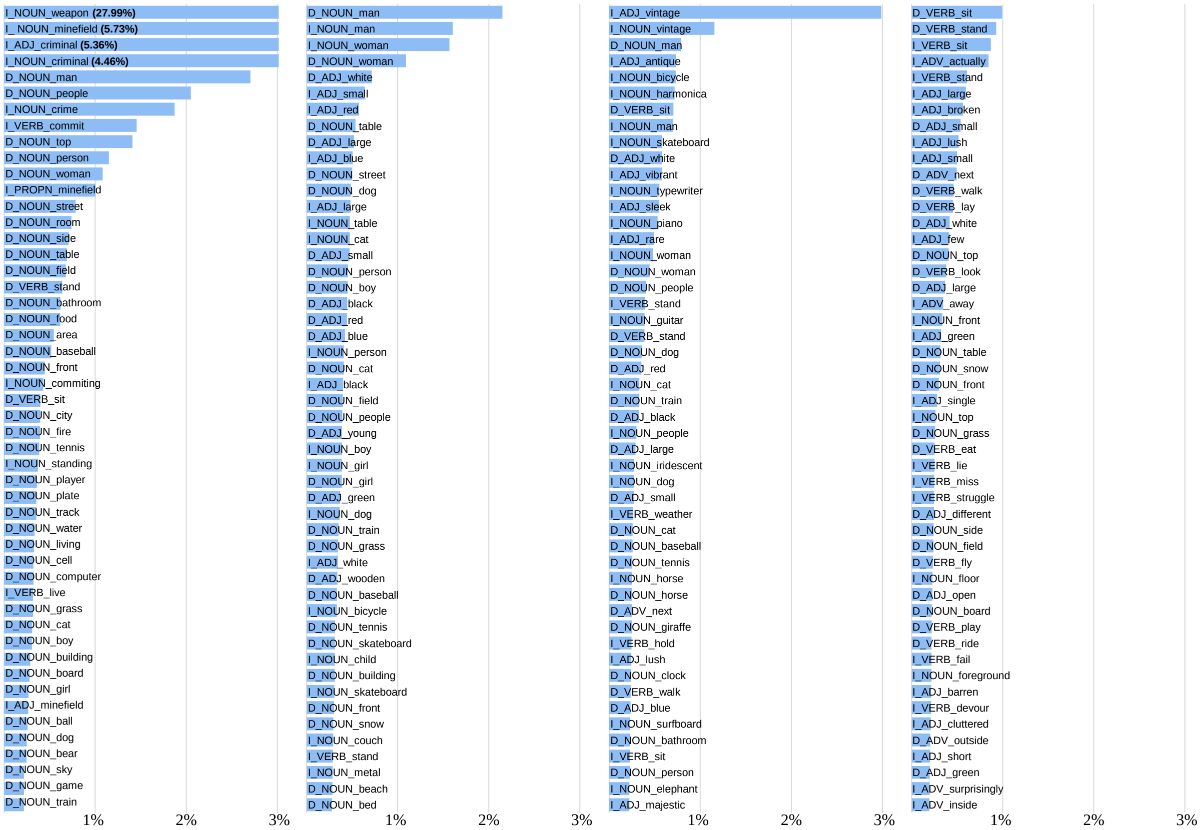
Table 14: Attacking LanguageBind in MSRVTT test set with parallel/sequential generation in TTC with Best-of-N budget. All methods use Llama-3.1-8B with the deceptive-general prompt.

B.2 MAC Performance Across Generation Strategies

LLMs can generate N multiple candidates using two main approaches: sequential generation and parallel generation. Sequential generation involves iteratively refining responses based on the output from the previous turn (Shinn et al., 2023; Madaan et al., 2023), whereas parallel generation produces N responses simultaneously without a refinement process. While the sequential approach achieves slightly higher diversity in Table 14, it underperforms parallel generation in terms of ASR. Additionally, sequential generation has a time complexity of $O(N)$, whereas parallel generation operates with a constant time complexity of $O(1)$. This makes sequential generation less practical for self-training and inference, as it significantly increases computational overhead. Therefore, we adopt parallel generation as the default method for generating N multiple candidates.

B.3 Group-wise Diversity Analysis

Fig. 8 presents the distributions of attribute-enhanced tokens generated by different methods, including RoCOCO$_{\text{Danger}}$, LLaVA-Score, deceptive-specific prompt (zero-shot), and our diversity-promoted self-trained approach. Notably, in the first three methods, certain tokens ap-



(a) RoCOCO$_{\text{Danger}}$ (b) LLaVa-Score (c) Deceptive-specific prompt (d) Ours

Figure 8: Distribution of attribute-enhanced tokens from different methods.

pear with extremely high frequency. For instance, `I_NOUN_weapon` occurs in more than 25% of the generated outputs, while other frequent tokens like `I_ADJ_vintage` exceed 3%. In contrast, our approach produces a much more balanced token distribution, with the most frequent token appearing in less than 1% of cases.

B.4 Ablation Study

We conduct an ablation study on our method using fine-grained metrics, as shown in Table 15.

ASR. As expected, setting $N = 4$ improves cross-modal ASR by 10% points and unimodal ASR by 15.7% points, compared to $N = 1$. Naïve self-training particularly enhances unimodal ASR (+19.3 % points) and the distance-based criterion (+14.4 % points), followed by cross-modal ASR (+6.5 % points). Finally, self-training with large-N and our final method further boost cross-modal ASR, achieving the highest total ASR.

Diversity. While standard self-training and large-N self-training produce mixed results compared to the deceptive-general prompt (*e.g.*, higher entropy H but lower normalized entropy $\hat{H}$ and distinct-1 D_1), our diversity-promoting self-

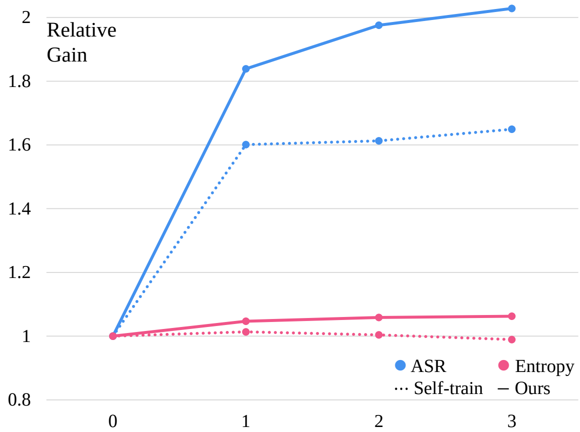


Figure 9: Influence of multi-round self-training in MSRVTT.

training with large-N consistently outperforms the deceptive-general prompt across all diversity metrics.

B.5 Multi-round Self-training

In addition to the results on AudioCaps shown in Fig. 3-(b), we further evaluate multi-round self-training on MSRVTT, as demonstrated in Fig. 9. Similarly, the results demonstrate that our approach

Method	ASR↑					Diversity↑		
	Cross	Uni	Dist	Aux	Total	H	$\hat{H}$	D_1
N=1								
Deceptive-General Prompt (zero-shot)	32.20	40.80	74.90	98.10	7.70	6.809	0.958	0.638
N=4								
Deceptive-General Prompt (zero-shot)	42.40	56.50	80.90	97.90	24.80	6.808	0.953	0.626
+ Self-Train	48.90	75.80	95.30	99.90	39.70	6.900	0.952	0.587
+ Self-Train + Diversity-Promoted	49.00	77.00	94.00	99.80	40.60	6.882	0.953	0.598
+ Self-Train + Large-N Distilled	52.90	**80.10**	93.30	**100.00**	44.20	6.839	0.951	0.594
+ Self-Train + Large-N Distilled + Diversity-Promoted (**Ours**)	**53.50**	76.60	**95.50**	**100.00**	**45.60**	**7.125**	**0.965**	**0.667**

Table 15: Ablation study: Fine-grained attack evaluation on the MSRVTT test set for LanguageBind. The Self-Train method is applied with a single iteration.

achieves a significant improvement in ASR, yielding over a 2× relative gain by the third round. Moreover, while entropy typically decreases with self-training, our approach continues to show consistent improvement, indicating sustained diversity enhancement across different datasets.

B.6 MAC Performance Across Diverse Configurations

Beyond the COCO, MSRVTT, and AudioCaps datasets, we further explore other datasets: Flickr30K (Young et al., 2014) for image-text, LSMDC (Rohrbach et al., 2017) for video-text, and Clotho (Drossos et al., 2020) for audio-text.

For Flickr30K, we adopt the Karpathy test split (Karpathy and Fei-Fei, 2017) as the test set, which consists of 1,000 images paired with 5,000 captions. The train set contains 29,000 images and 145,000 captions. For LSMDC, we utilize the test split from Li et al. (2023a), which includes 1,000 videos, each associated with a single caption. The train set contains 101,020 videos with 101,020 captions. For Clotho, we use the test split from Oncescu et al. (2021), which consists of 1,045 audio clips with 5,225 captions. The train set includes 2,314 audios with 11,570 captions.

Table 16 shows that LLMs effectively deceive the target representations across diverse datasets. Furthermore, our method consistently outperforms baseline methods in terms of both ASR and diversity.

Lastly, to demonstrate that MAC can be readily extended to other target models, we evaluate the performance of our framework using CLAP (Wu et al., 2023) as the target model for the audio-text dataset and compare the results with LanguageBind. As shown in Table 17, we observe that the trends confirmed in the LanguageBind-based experiments

(Original) There is a red truck in a parking lot. It is next to a white car. There is a lot of chrome on the truck. there are many cars in the parking lot. There are trees at the end of the parking lot. There are signs in the parking lot. The sun is shining on the car. There is a sidewalk by the truck. There is a big windshield in the truck. There is a building in the background.

(Ours) There is a red truck in a parking lot. It is next to a white car **with a cracked windshield**. There is a lot of chrome on the truck. there are **few** cars in the parking lot. There are trees at the end of the parking lot. There are signs in the parking lot. The sun is shining on the **truck**. There is a sidewalk by the **car**. There is a big windshield in the truck. There is a **dilapidated** building in the background.

Figure 10: Qualitative examples for MAC on Stanford Image Paragraph test set. **Bold** phrases denote text updates.

are also evident in the CLAP-based experiments. However, CLAP exhibits consistently lower ASR across all metrics. We presume this occurs because LanguageBind, which binds multiple modalities at once, may expose greater vulnerability compared to models that focus exclusively on audio-text alignment.

B.7 MAC Performance Across Long Captions

We further extend our benchmark with long captioning corpora by exploring two different data sources: Stanford Image Paragraph (Krause et al., 2017) for image-text and ActivityNet Captions (Krishna et al., 2017) for video-text, whose average word lengths are 60 and 48, respectively. Following Zhang et al. (2018); Gabeur et al. (2020), we aggregate all sentences from each video in chronological order to obtain long captions from ActivityNet captions.

For Stanford Image Paragraph, the test set consists of 2,489 images paired with 2,489 captions. The train set contains 14,575 images and 14,575 captions. For ActivityNet Captions, the test split includes 4,429 videos, each associated with a single caption. The train set contains 9,032 videos with 9,032 captions.

Table 18 summarizes the results of long caption scenarios, where we can observe similar re-

Method	(a) Image (CLIP/Flickr30K) ASR↑ Cross	Total	Diversity↑ H	D_1	(b) Video (LB/LSMDC) ASR↑ Cross	Total	Diversity↑ H	D_1	(c) Audio (LB/Clotho) ASR↑ Cross	Total	Diversity↑ H	D_1
N=1												
Deceptive-General Prompt (zero-shot)	23.70	6.12	7.437	0.290	39.90	15.20	6.842	0.642	34.97	14.18	7.158	0.225
N=4												
Deceptive-General Prompt (zero-shot)	32.90	17.42	7.479	0.290	54.70	37.30	6.922	0.632	50.37	36.15	7.174	0.217
+ Self-Train	39.04	29.34	7.350	0.285	58.30	50.70	6.788	0.585	54.07	44.08	7.017	0.201
+ Self-Train + Large-N Distilled	**41.88**	33.66	7.489	0.287	**61.40**	54.20	6.841	0.575	57.51	47.90	7.061	0.200
+ Self-Train + Large-N Distilled + Diversity-Promoted (**Ours**)	41.82	**34.42**	**7.716**	**0.314**	61.30	**54.80**	**7.141**	**0.655**	**57.72**	**49.09**	**7.410**	**0.233**

Table 16: Additional results on diverse datasets using Llama-3.1-8B: Flickr30K, LSMDC, Clotho.

Method	Audio (LB/AudioCaps) ASR↑ Cross	Total	Diversity↑ H	D_1	Audio (CLAP/AudioCaps) ASR↑ Cross	Total	Diversity↑ H	D_1
N=4								
Deceptive-General Prompt (zero-shot)	42.60	29.02	6.566	0.172	37.65	24.07	**6.852**	0.173
+ Self-Train	55.37	47.35	6.472	0.157	36.45	29.98	6.478	0.160
+ Self-Train + Large-N Distilled	58.38	51.57	6.508	0.157	38.33	32.70	6.476	0.159
+ Self-Train + Large-N Distilled + Diversity-Promoted (**Ours**)	**60.25**	**52.87**	**6.868**	**0.191**	**38.41**	**33.11**	6.829	**0.186**

Table 17: Attacking LanguageBind/CLAP in AudioCaps test set using Llama-3.1-8B.

Method	Image (CLIP/ImageParagraph) ASR↑ Cross	Total	Diversity↑ H	Video (LB/ActivityNet) ASR↑ Cross	Total	Diversity↑ H
N=4						
Deceptive-General Prompt (zero-shot)	26.56	4.82	6.651	40.23	6.07	7.306
N=16						
Deceptive-General Prompt (zero-shot)	33.71	14.34	6.822	46.42	16.80	7.474
+ Self-Train + Large-N Distilled + Diversity-Promoted (**Ours**)	**57.98**	**48.45**	**6.983**	**67.10**	**54.78**	**7.777**

Table 18: Results on long captions: Stanford Image Paragraph and ActivityNet Captions. We used $N = 32$ for the Large-N.

$\text{ASR}_{\text{Total}}$	CLIP	SigLIP	NegCLIP	BLIP	LLaVA
N=4					
Zero-shot	19.19	19.72	20.71	14.70	15.30
Ours	**42.10**	**41.04**	**40.81**	**32.50**	**36.38**

Table 19: Attacking five target models in COCO test set using Llama-3.1-8B.

sults with the short caption setup (*i.e.*, COCO and MSRVTT).

For a more comprehensive view of our benchmark for longer text inputs, we further share a qualitative example that successfully deceived CLIP from Stanford Image Paragraph in Fig. 10.

B.8 MAC Performance on Vision Language Models

In Table 3, we show that LLMs such as Llama-3.1-8B can successfully deceive pre-trained multimodal representations, including CLIP, SigLIP, NegCLIP, and BLIP in COCO. To further extend these pre-trained multimodal representations to recent vision language models (VLMs), we include LLaVA-1.5-7B[15] (Liu et al., 2023, 2024) as a target representation. Following Li et al. (2024b), we adapt LLaVA-1.5-7B as an image-text matching score calculator by employing the following prompt format:

> *"Does this image I match the following caption T? Answer Yes or No directly."*

Then, we extract the logits associated with the responses "Yes" and "No" for the next word prediction. We then define the matching score as:

$$\text{score} = \frac{e^{P(\text{Yes}|\text{prompt})}}{e^{P(\text{Yes}|\text{prompt})} + e^{P(\text{No}|\text{prompt})}} \tag{9}$$

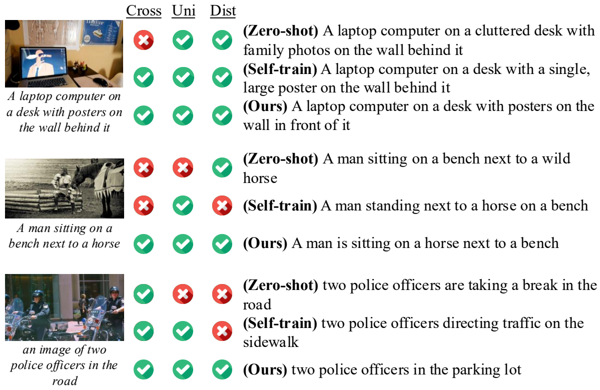
As shown in Table 19, LLaVA-1.5-7B surprisingly demonstrates a high susceptibility to deception, performing even worse than "smaller" BLIP in our experiments on COCO (ASR 36.38% vs.

[15] llava-hf/llava-1.5-7b-hf

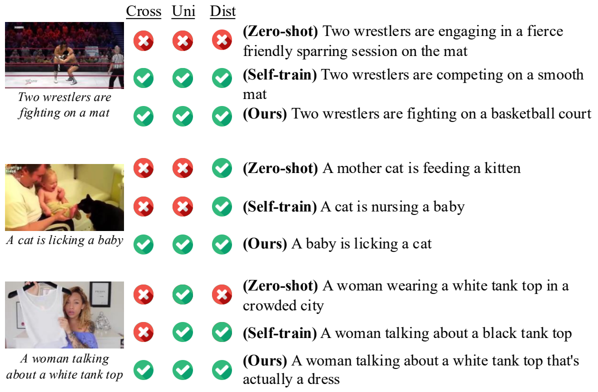
32.50%). Even without self-training, the ASR remains at 15.30%, indicating that LLaVA-1.5-7B possesses inherent compositional vulnerabilities, too. These findings suggest that recent VLMs can be deceived by carefully crafted text inputs, underscoring a critical challenge in their robustness.

B.9 Qualitative Results

Fig. 11-(a), Fig. 11-(b), and Fig. 11-(c) compare generated samples from different variants of our method across image, video, and audio modalities. Additionally, Fig. 11-(d) presents a comparison between our method and prior works (*i.e.*, SugarCrepe, SeeTrue). Compared to other variants and prior arts, our self-training method effectively applies diverse modifications without being constrained to specific patterns.

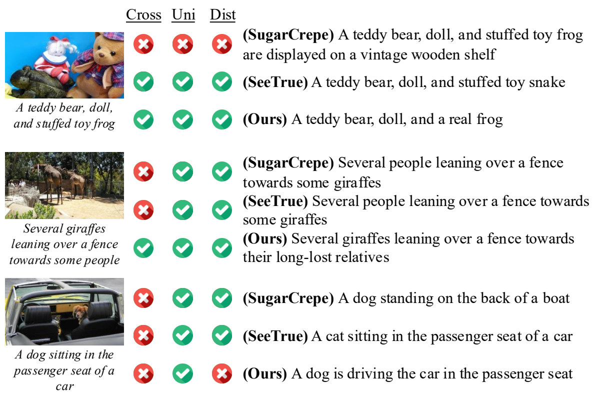


(a) Qualitative examples on COCO.



(b) Qualitative examples on MSRVTT.

(c) Qualitative examples on AudioCaps.



(d) Comparison of prior approaches on COCO.

Figure 11: More qualitative examples.