# Mitigating Image Captioning Hallucinations in Vision-Language Models

Fei Zhao, Chengcui Zhang, Tianyang Wang, Xi Li
Department of Computer Science
The University of Alabama at Birmingham, Birmingham, USA
{larry5, czhang02, tw2, xiliuab}@uab.edu

***Abstract*—Hallucinations in vision-language models (VLMs) hinder reliability and real-world applicability, usually stemming from distribution shifts between pretraining data and test samples. Existing solutions, such as retraining or fine-tuning on additional data, demand significant computational resources and labor-intensive data collection, while ensemble-based methods incur additional costs by introducing auxiliary VLMs. To address these challenges, we propose a novel test-time adaptation framework using reinforcement learning to mitigate hallucinations during inference without retraining or any auxiliary VLMs. By updating only the learnable parameters in the layer normalization of the language model (approximately 0.003% of the model parameters), our method reduces distribution shifts between test samples and pretraining samples. A CLIP-based hallucination evaluation model is proposed to provide dual rewards to VLMs. Experimental results demonstrate a 15.4% and 17.3% reduction in hallucination rates on LLaVA and InstructBLIP, respectively. Our approach outperforms state-of-the-art baselines with a 68.3% improvement in hallucination mitigation, demonstrating its effectiveness.**

***Index Terms*—Vision-Language Models, Reinforcement Learning, Hallucination Mitigation, Image Captioning**

## I. Introduction

VLMs have become foundational for tasks such as image captioning and visual question answering (VQA), demonstrating remarkable capabilities in aligning textual and visual modalities [1]. However, these models often suffer from a critical issue: hallucinations, where the generated output deviates from the input image content. Such hallucinations undermine the reliability and applicability of VLMs in real-world scenarios, particularly in high-stake domains including autonomous systems, medical image analysis, and surveillance, where factual accuracy is paramount [2].

The root cause of hallucinations often lies in distribution shifts between pretraining data and real-world test samples. Pretraining on noisy datasets introduces biases and reliance on unimodal priors, leading models to generate hallucinated content when presented with unseen or domain-specific data. Addressing this challenge requires methods that adapt models to new data distributions while ensuring computational efficiency.

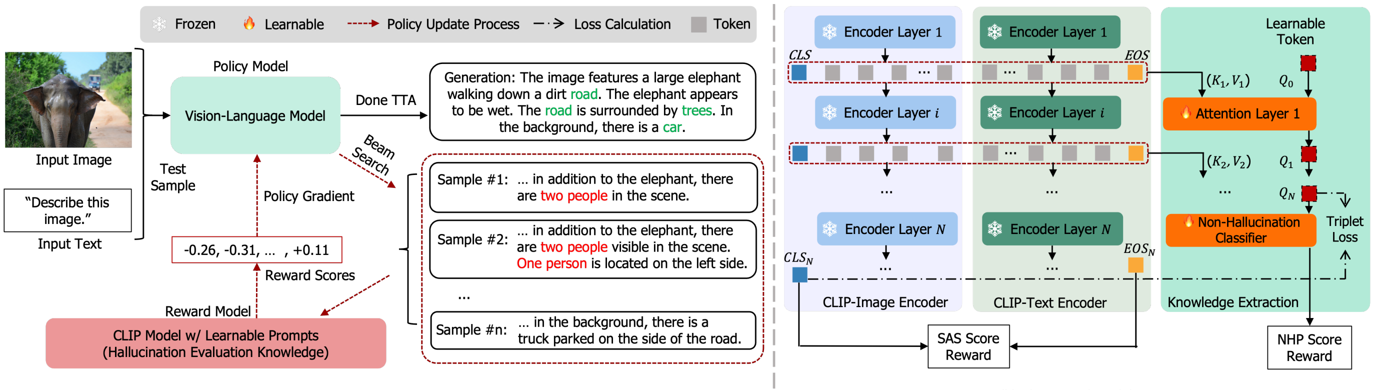
Existing methods to mitigate hallucinations, including retraining or fine-tuning, ensemble approaches, and logit manipulation, face limitations in scalability, computational efficiency, and generalizability (see Section II).

In this work, we propose a novel test-time adaptation (TTA) framework using reinforcement learning (RL) to dynamically mitigate hallucinations during inference. In our approach, the VLM itself acts as the policy model in an RL framework, allowing it to iteratively refine its output captions based on feedback from a hallucination evaluation model. RL is well-suited to this task as it enables the model to optimize its decisions (caption generation) based on feedback signals (rewards) rather than requiring extensive retraining. The policy model learns to adapt to each test sample individually, reducing distribution shifts on a per-sample basis.

The key contributions of this work are as follows:

1. **A Novel TTA Framework for Hallucination Mitigation:** We introduce a lightweight and efficient TTA method using RL to reduce the data distribution shift between test samples and pretraining data. It **mitigates hallucinations in VLMs during inference**, without requiring retraining or auxiliary VLMs.
2. **A Lightweight Hallucination Evaluation Model:** We propose a CLIP-based hallucination evaluation model equipped with a learnable query prompt to extract knowledge from the frozen CLIP encoders. This model is capable of independently classifying image-text pairs as non-hallucinated or hallucinated. Furthermore, it serves as a reward model in RL, providing dual rewards: *Semantic Alignment Score* (SAS) for image-text alignment and *Non-Hallucination Probability* (NHP) scores for the likelihood of non-hallucination. While SAS measures semantic alignment, it does not ensure the caption is free from hallucinations. NHP complements SAS by assessing factual consistency, addressing hallucination. Together, these dual rewards guide the policy model to dynamically refine the generated content, achieving superior performance in mitigating hallucinations.
3. **A Parameter-Efficient RL Approach for VLMs:** Our approach updates only the learnable parameters in layer normalization [3], accounting for approximately 0.003% of the model's total parameters. This is significantly more efficient than modifying entire VLMs or their cross-modal projection layers that have nearly 100 times more parameters.

**Scope**: This paper focuses on mitigating object hallucinations in VLMs, specifically addressing captions with incorrect


Frozen
Learnable
Policy Update Process
Loss Calculation
Token
Policy Model
Vision-Language Model
Input Image
"Describe this image."
Input Text
Test Sample
Done TTA
Generation: The image features a large elephant walking down a dirt road. The elephant appears to be wet. The road is surrounded by trees. In the background, there is a car.
Beam Search
Sample #1: ... in addition to the elephant, there are two people in the scene.
Sample #2: ... in addition to the elephant, there are two people visible in the scene. One person is located on the left side.
...
Sample #n: ... in the background, there is a truck parked on the side of the road.
Policy Gradient
-0.26, -0.31, ... , +0.11
Reward Scores
Reward Model
CLIP Model w/ Learnable Prompts (Hallucination Evaluation Knowledge)
Encoder Layer 1
$CLS$
$EOS$
Encoder Layer $i$
...
Encoder Layer $N$
$CLS_N$
$EOS_N$
CLIP-Image Encoder
CLIP-Text Encoder
Learnable Token
$(K_1, V_1)$
$Q_0$
Attention Layer 1
$(K_2, V_2)$
$Q_1$
$Q_N$
Non-Hallucination Classifier
Triplet Loss
Knowledge Extraction
SAS Score Reward
NHP Score Reward

(a) The overall framework

(b) The reward model

Fig. 1. (a) The overall framework for mitigating object hallucinations in VLMs using TTA. The red texts indicate object hallucinations in the generated captions before TTA, while the green text highlights accurate descriptions appearing after TTA. (b) The architecture of the hallucination evaluation model, which leverages a frozen CLIP model with learnable query prompts, multi-stage cross-attention layers, and a binary classifier. The model facilitates training through triplet loss to differentiate between non-hallucinated and hallucinated captions.

or non-existent objects. Attribute and relation hallucination mitigation is beyond the scope of our research for two reasons: first, there are too few benchmarks for evaluating these types of hallucinations, many of which rely on non-open-source APIs that lack accessible resources; second, our primary goal is to compare with the SOTA method VCD [4], which primarily targets object hallucination mitigation.

The remainder of this paper is organized as follows: Section II reviews related work on hallucination mitigation in VLMs. Section III details our proposed methodology, including the TTA framework, the RL setup, and the customized CLIP-based hallucination evaluation model. Section IV describes the datasets used for experiments, while Section V presents experimental results and offers ablation studies to further highlight the contributions of individual components. Finally, Section VI concludes the paper and discusses potential future directions.

## II. Related Work

VLMs have made significant progress in multimodal tasks, including image captioning and visual question answering (VQA). Proprietary state-of-the-art (SOTA) VLMs such as ChatGPT [5], Gemini [6], and Claude demonstrate impressive multimodal capabilities but are not open source, restricting their accessibility for research and customization. In contrast, open-source VLMs such as LLaVA [7] and InstructBLIP [8] offer competitive performance. Currently, VLMs are widely adopted for real-world applications. For instance, LLaVA has been effectively applied to extract financial information from check images [9]. Despite these advancements, VLMs are prone to hallucinations, where the generated output deviates from visual input, particular under distribution shifts between pretraining and real-world test data.

Current approaches for mitigating hallucinations can be grouped into four categories: **(1) Visual input enhancement**, focusing on improving the quality of visual features fed into VLMs. For example, methods like LLaVA-Next [10] and InternVL [11] adopt a multi-scale strategy: dividing the input image into smaller patches, resizing these patches to higher resolutions to emphasize local details, and downscaling the original image to capture global context. These processed images are then concatenated to create a richer visual representation. Similarly, Prismer [12] incorporates auxiliary vision experts/submodels to further diversify visual inputs and enhance performance. While effective, these methods require complete retraining of the VLM, which is computationally expensive and impractical for refining already-deployed models. **(2) Fine-tuning VLMs**, typically on domain-specific datasets, has shown promise in mitigating hallucinations. For instance, the authors in [13] fine-tuned LLaVA [7] for medical applications using newly collected domain-specific VQA data, significantly improving performance. Similarly, the authors in [14] apply RL to fine-tune VLMs such as CLIPCap [15], focusing on training the vision-language projector (**42 million parameters**). While effective, fine-tuning VLMs requires extensive computational resources and domain-specific data collection, making it resource-intensive and challenging to scale. **(3) Ensemble-based methods**, usually aggregating outputs from multiple VLMs or combining a weaker model with a stronger one. For instance, the authors in [16] propose a debate-style ensemble framework where two VLMs iteratively refine their outputs. Meanwhile, the authors in [17] leverage ChatGPT to provide detailed feedback for refining LLaVA outputs, reducing hallucination rates. While these techniques improve robustness, they are computationally costly. Furthermore, using a stronger model to enhance a weaker one often negates the practical benefits of deploying the less capable model. **(4) Logit manipulation methods**, modifying output distributions to mitigate hallucinations. The work [18] employs an external vision encoder to provide soft prompts, mitigating hallucinations without additional training. However, without fine-tuning, the external encoder may produce visual features that are misaligned with the vision-language projector, potentially introducing noise and degrading performance. Similarly, Visual Contrastive Decoding (VCD) [4] reduces hallucinations by contrasting logits generated from original and noise-distorted inputs, assuming that matching outputs across the two inputs indicate hallucination. While this approach shows promise for single-token prediction tasks, it faces significant limitations in auto-regressive generation. The assumption that identical output tokens from those two inputs are hallucinated breaks down in sequential tasks, as the distorted input cannot guarantee that all its outputs are incorrect or hallucinated. For instance, common words such as "this," "that," "is," or "are" might appear in both the original and distorted outputs, but they are often correct and contextually appropriate. Misclassifying these correct tokens as hallucinated can lead to unnecessary modifications, disrupting the generation process and introducing cascading errors that negatively affect subsequent tokens. These limitations make VCD less scalable and effective for auto-regressive applications.

**Algorithm 1** TTA with RL for VLMs Hallucination Mitigation

```
Require: Pretrained VLM, CLIP_{Prompts + Triplet}, Visual Input $v$,
    Textual Input $x$, Steps $num\_steps$, Beam Size $B$
Ensure: Refined Caption $y^*$
Initialize: Freeze VLM parameters except LayerNorm
    gamma. Load pretrained CLIP_{Prompts + Triplet}.
for $step = 1 \to num\_steps$ do
    Generate $B$ candidate captions using beam search.
    For each caption, compute:
        • SAS score for image-text alignment.
        • NHP score for non-hallucination likelihood.
    Aggregate and normalize reward scores. (see Section
        III-B)
    Update LayerNorm gamma using policy gradient optimization.
end for
Generate the refined caption $y^*$ using the updated VLM.
return $y^*$
```

## III. Methodology

As we mentioned in Section I, we treat the VLM as a policy model that iteratively refines its output captions based on feedback from a hallucination evaluation model, which serves as the reward provider. This framework operates entirely during inference, eliminating the need for retraining or auxiliary models, and dynamically adapts the VLM to test samples.

As shown in Fig. 1, the process begins with the policy model generating multiple candidate captions for a given test image and a prompt "Describe this image" using beam search [19]. Each candidate is combined with the input text image and evaluated by the reward model, which generates dual reward scores: a CLIP-based score for image-text alignment and a logit score for non-hallucination likelihood. These scores are aggregated to form a final reward signal. The policy model is then updated using policy gradient loss to improve its captioning decisions. This iterative process continues for a predefined number of steps, culminating in the generation of a refined caption. The entire process is outlined in Algorithm 1.

### A. Reinforcement Learning for Vision-Language Models

In our framework, the auto-regressive VLM acts as the *policy model*, parameterized by $\theta$, and the task of generating a caption for an input image is treated as a sequential decision-making problem. The key components are:

- **State** $s$: The state represents the context available to the model at each step $t$, which includes:
  - $v$: The visual input (e.g., images).
  - $x$: The textual input (e.g., "Describe this image").
  - $y_{<t}$: The sequence of tokens generated so far up to token generating step $t$.

  Thus, $s_t = \{v, x, y_{<t}\}$, encapsulating all information available for generating the next token.
- **Action** $a_t$: The next token $y_t$ is generated at step $t$.
- **Policy** $\pi(a|s;\theta)$: The VLM defines a probabilistic policy over actions given the state. At each step $t$, the probability of generating a token $y_t$ is:

  $$\pi(y_t|s_t;\theta) = softmax(f(v, x, y_{<t};\theta)),$$

  where $f$ is the VLM's output logits.
- **Reward** $r$: After generating the entire caption $y = \{y_1, y_2, \ldots, y_T\}$, a reward $r(y, v)$ is calculated by the hallucination evaluation model (see Section III-B).

The goal of RL is to maximize the expected reward $J(\theta)$, defined as:

$$J(\theta) = \mathbb{E}_{\pi_\theta}[r(y, v)],$$

where $r(y, v)$ is the reward assigned to the caption $y$. To optimize $J(\theta)$, we use the *policy gradient method* [20], which computes the gradient of the objective function with respect to the model parameters $\theta$:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi(y|s;\theta) \cdot r(y, v)\right].$$

This equation adjusts the model parameters to increase the likelihood of generating captions that maximize the reward.

During TTA, only the *gamma parameters in layer normalization* are updated, while the rest of the model is frozen. The updates are guided by the policy gradient loss, defined as:

$$\mathcal{L}_{\text{policy}} = -\mathbb{E}_{\pi_\theta}\left[\log \pi(y|s;\theta) \cdot r(y, v)\right].$$

This formulation converts the maximization problem into a minimization problem by introducing the negative sign, aligning with standard optimization procedures. The logarithm (log) plays a key role in shaping the model's behavior:

- **For positive rewards** ($r(y, v) > 0$): Minimizing the loss follows the $-\log$ curve, which pushes the logit higher, increasing the probability of generating this output.
- **For negative rewards** ($r(y, v) < 0$): Minimizing the loss moves the logit along the $\log$ curve to the left, reducing its probability.

This interplay between the reward and the $-\log$ curve allows the model to reinforce high-reward outputs and suppress low-reward ones, creating a self-correcting RL loop that refines captions effectively.

### *B. Reward Calculation*

In our framework, the reward signal combines two components to guide the VLM toward generating accurate and semantically aligned captions. The first component is the *Semantic Alignment Score* (SAS), which evaluates the cosine similarity between the visual input $v$ and the generated caption $y$ based on their embeddings from the hallucination evaluation model's frozen vision and text encoders, shown in Fig. 1. The second component is the *Non-Hallucination Probability* (NHP), derived from the hallucination evaluation model's classification subnetwork that predicts the likelihood of the caption being non-hallucinated. The reward values are normalized by subtracting the mean across all candidate captions. The final reward $r$ is computed as:

$$r(y, v) = \text{norm\_SAS} + \beta \cdot \text{norm\_NHP}$$

where $\beta$ is a weighting factor. This dual-reward system ensures that the VLM generates captions that are more semantically aligned with the input image with less hallucination, making it a crucial component of the RL framework.

### *C. Hallucination Evaluation Model*

The hallucination evaluation model is built upon the frozen CLIP architecture, enhanced with a single learnable query token $Q$. This token is updated iteratively through $N$-stage cross-attention [21] layers to extract alignment and semantic consistency features between image and text inputs. The final updated query token features ($Q_N$) serve as input to a classifier, composed of stacked fully connected layers, which predicts whether an image-text pair is non-hallucinated (positive) or hallucinated (negative). The classifier's output, a logit score referred to as the NHP score, serves as a crucial reward in the RL framework.

**Learnable Query Tokens:** The single learnable query token $Q$ interacts with image and text features through cross-attention. At each stage $i$, the tokens are updated as:

$$Q_i = \text{CrossAttention}(Q_{i-1}, K_i, V_i), \quad i \in \{1, 2, \dots, N\},$$

where $Q_{i-1}$ is the query token from the previous stage, and $K_i, V_i$ represent the keys and values derived from the concatenated image ($v$) and text ($y$) features at stage $i$.

The CLS token (classification token representing the image's global features) from the image encoder and the EOS token (End-of-Sequence token) from the text encoder are used to compute the SAS score via their cosine similarity, providing a quantitative measure of alignment between the image and the caption.

**Triplet Loss:** To guide the learnable query token $Q$ in extracting meaningful features from frozen vision and textual encoders, we incorporate a triplet loss [22] with a margin $\alpha$. The triplet comprises the image as the anchor, a ground truth (positive) caption, and a generated (negative) caption that contains hallucinations (details in Section IV). Let $CLS_N$ denote the final CLS token, and $Q_N^{\text{pos}}$ and $Q_N^{\text{neg}}$ be the updated query tokens for the positive and negative captions, respectively. The loss is defined as:

$$\mathcal{L}_{\text{triplet}} = \max\left(0, \cos\left(CLS_N, Q_N^{\text{neg}}\right) - \cos\left(CLS_N, Q_N^{\text{pos}}\right) + \alpha\right)$$

where $\cos$ represents cosine similarity. The triplet loss ensures that the query token extracts discriminative features for the subsequent binary classification.

## IV. Dataset and Experiments

This work focuses on mitigating object hallucinations in image captioning tasks. To achieve this, we utilize the AMBER [23] dataset for evaluation and a curated subset of the PixelProse [24] dataset for training our customized CLIP-based hallucination evaluation model.

The AMBER dataset contains 1,004 test samples designed specifically to benchmark generative tasks. The subset of PixelProse dataset, with approximately 30,000 samples, is used to train and test the hallucination evaluation model. Of these, 2,848 samples are used for testing the hallucination evaluation model's performance, and the remaining samples are for training. We use LLaMA3 [25] to generate negative image-caption pairs, including object hallucinations, as well as attribute and relation hallucinations. We apply our RL framework on two VLMs: LLaVA 7B [7] and InstructBLIP 7B [8] , with Vicuna backbone. The RL framework performs TTA on each AMBER test sample, leveraging feedback from the hallucination evaluation model. During RL-based inference, the learning rate is set to 2e-4 for InstructBLIP and 2e-3 for LLaVA. The beam size is set to 5, and the automatic adaptation process spans 5 steps per sample. After completing the TTA process for each test sample, the parameters of the VLM are reset to their initial state. This approach is crucial as it prevents the accumulation of sample-specific biases, maintains the generalizability of the pretrained model, and ensures that adaptations made for one sample do not negatively influence the performance on subsequent test samples.

The RL framework operates on a cluster of 4 A100 GPUs with 80 GB memory each, while the training of the hallucination evaluation model is performed on a single A100 GPU with 40 GB memory. The metrics are as follows [23]:

**CHAIR** (Cumulative Hallucination Rate) measures the frequency of hallucinated objects in captions. It is defined as:

$$\text{CHAIR}(R) = 1 - \frac{\text{len}(R'_{\text{obj}} \cap A_{\text{obj}})}{\text{len}(R'_{\text{obj}})},$$

where $R'_{\text{obj}}$ is the set of objects mentioned in the response, and $A_{\text{obj}}$ represents the ground truth objects.

**Cover** quantifies the proportion of ground-truth objects covered in the caption. It is calculated as:

$$\text{Cover}(R) = \frac{\text{len}(R'_{\text{obj}} \cap A_{\text{obj}})}{\text{len}(A_{\text{obj}})}.$$

**Hal** (Hallucination Rate) indicates whether a response contains hallucinated objects. It is defined as:

$$\text{Hal}(R) = \begin{cases} 1 & \text{if CHAIR}(R) \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

**Cog** (Cognitive Overlap) evaluates the overlap between hallucinated objects in the responses and human cognitive tendencies. It is computed as:

$$\text{Cog}(R) = \frac{\text{len}(R'_{\text{obj}} \cap H_{\text{obj}})}{\text{len}(R'_{\text{obj}})},$$

where $H_{\text{obj}}$ is the objects commonly hallucinated by humans.

## V. Results and Ablation Studies

As shown in Table I, our framework demonstrates strong performance on both InstructBLIP and LLaVA 7B, with most metrics showing significant improvement after TTA. While LLaVA's Cover metric drops slightly, likely due to the model narrowing the object range to improve accuracy and reduce hallucinations, this trade-off is acceptable given the overall gains in coherence and hallucination reduction. These results highlight the effectiveness of our approach in enhancing caption quality across diverse models.

TABLE I. Comparison Across Models (w/ and w/o TTA)

| **Model** | **Type** | **CHAIR (↓)** | **Cover (↑)** | **Hal (↓)** | **Cog (↓)** |
|---|---|---|---|---|---|
| Instruct BLIP | w/o TTA | 8.8 | 52.2 | 38.2 | 4.4 |
| | w/ TTA | 6.5 | 54.1 | 31.6 | 3.3 |
| | *Change* | **-26.1%** | **+3.6%** | **-17.3%** | **-25.0%** |
| LLaVA 7B | w/o TTA | 7.8 | 51.0 | 36.4 | 4.2 |
| | w/ TTA | 6.7 | 49.9 | 30.8 | 3.7 |
| | *Change* | **-14.1%** | **-2.2%** | **-15.4%** | **-11.9%** |

Table II highlights the efficiency and effectiveness of our framework. Both LLaVA 7B and InstructBLIP update only 0.0038% and 0.0034% of their total parameters, respectively, focusing exclusively on the layer normalization gamma parameters, and yet achieve significant performance gains. As shown in Table I, these minimal updates yield substantial reductions in hallucination, including CHAIR, Hal, and Cog for both models, with only a slight drop on Cover for LLaVA 7B. The improvements demonstrate how even a few learnable parameters, strategically optimized, can result in large-scale enhancements in caption accuracy and reliability, further underscoring the lightweight yet powerful design of our framework.

TABLE II. Learnable Parameters Across Models

| **Model** | **Total Params** | **Learnable Params** | **Percent** |
|---|---|---|---|
| CLIP$_{\text{Prompts + Triplet}}$ | 157M | 5.8M | 3.69 |
| LLaVA 7B | 7.06B | 0.27M | 0.0038 |
| InstructBLIP | 7.91B | 0.27M | 0.0034 |

As shown in Table III, the performance of LLaVA 7B with TTA demonstrates substantial improvements over the SOTA method Visual Contrastive Decoding (VCD) [4] across key metrics. Specifically, CHAIR and Hal scores, indicating hallucination levels, are reduced by 91.7% and 68.4%, respectively, while the Cover score, reflecting object coverage, increases by 384.5%. These significant gains demonstrate the capacity of TTA in mitigating hallucinations and improving object coverage, where **VCD struggles due to its limitations in addressing auto-regressive tasks that we have discussed in Section II**. Additionally, the lower Cog score achieved by TTA highlights its enhanced cognitive coherence, further establishing it as a superior method for hallucination mitigation.

TABLE III. Comparison Between LLaVA 7B w/ VCD and LLaVA 7B w/ TTA

| **Metric** | **w/ VCD** | **w/ TTA** | **Difference** |
|---|---|---|---|
| CHAIR (↓) | 80.5 | 6.7 | **-91.7%** |
| Cover (↑) | 10.3 | 49.9 | **+384.5%** |
| Hal (↓) | 97.5 | 30.8 | **-68.4%** |
| Cog (↓) | 9.2 | 3.7 | **-59.8%** |

Table IV highlights the impact of incorporating the NHP reward from the customized evaluation model. Both InstructBLIP and LLaVA 7B demonstrate consistent improvements across key metrics when using the NHP reward. CHAIR and Hal scores decrease, indicating reduced hallucination levels. Also, the decreased Cog scores reflects enhanced cognitive coherence. Although Cover scores slightly decrease for both models (-0.7% for InstructBLIP and -0.4% for LLaVA 7B), this trade-off is acceptable given the significant reduction in hallucinations. The decrease in Cover may result from the models narrowing their focus to align more strictly with factual content, thereby slightly limiting object coverage.

TABLE IV. Comparison between w/ logit and w/o logit Across InstructBLIP and LLaVA Models

| **Model** | **Logit** | **CHAIR (↓)** | **Cover (↑)** | **Hal (↓)** | **Cog (↓)** |
|---|---|---|---|---|---|
| Instruct BLIP | w/o | 7.0 | 54.5 | 33.2 | 3.7 |
| | w/ | 6.5 | 54.1 | 31.6 | 3.3 |
| | *Change* | **-7.1%** | **-0.7%** | **-4.8%** | **-10.8%** |
| LLaVA 7B | w/o | 7.3 | 50.1 | 31.6 | 3.9 |
| | w/ | 6.7 | 49.9 | 30.8 | 3.7 |
| | *Change* | **-8.2%** | **-0.4%** | **-2.5%** | **-5.1%** |

As shown in Table II, the CLIP$_{\text{Prompts + Triplet}}$ model comprises only 5.8M learnable parameters, emphasizing its lightweight design. Before adopting it in our RL framework, we explored two variants. In CLIP$_{\text{stacked}}$, CLS and EOS token features are combined and passed into stacked fully connected layers for classification. In contrast, CLIP$_{\text{Prompts}}$ introduces a learnable token that interacts with the frozen encoders, producing features then fed into stacked fully connected layers. Although both variants outperform the LLaVA models, neither surpasses our final CLIP$_{\text{Prompts + Triplet}}$ model, which builds upon CLIP$_{\text{Prompts}}$ by incorporating a triplet loss (see Section III-C) to further refine the representation space. As shown in Table V, CLIP$_{\text{Prompts + Triplet}}$ achieves the highest F1 scores for Object (86.5%), Attribute (80.7%), and Relation (75.5%), surpassing the other models. These results highlight the synergistic effect of combining learnable prompts with triplet loss, enabling the CLIP-based model to deliver superior accuracy and robustness in hallucination evaluation while maintaining efficiency.

TABLE V. Comparison of Models on Hallucination Detection (F1 Scores in %)

| Model | Object (↑) | Attribute (↑) | Relation (↑) |
|---|---|---|---|
| LLaVA 7B | 15.0 | 26.6 | 28.2 |
| LLaVA 13B [7] | 10.2 | 41.3 | 52.3 |
| CLIP$_{\text{stacked}}$ | 84.9 | 77.8 | 70.3 |
| CLIP$_{\text{Prompts}}$ | 85.5 | 80.2 | 71.8 |
| CLIP$_{\text{Prompts + Triplet}}$ | **86.5** | **80.7** | **75.5** |

In contrast, the LLaVA models, including LLaVA 13B, show significantly lower scores in hallucination detection, particularly for object hallucinations (15.0% for LLaVA 7B and 10.2% for LLaVA 13B), highlighting the inherent difficulty of this task. We then retrained and tested the LLaVA 7B model on the same dataset, achieving results closer to our CLIP$_{\text{Prompts + Triplet}}$ model. However, its large parameter count incurs high computational costs, and its inability to generate CLIP scores for reward calculations in our RL framework makes it less suitable for our approach. Therefore, we focused on the lightweight and efficient CLIP-based approach, which better aligns with the requirements of our framework.

Additionally, as the hallucination evaluation model was trained on three types of hallucinations: object, attribute, and relation, it has the potential to be extended for broader hallucination detection tasks beyond the current scope. This versatility positions it as a promising foundation for future research and applications focused on comprehensive hallucination mitigation across diverse scenarios.

## VI. Conclusion and Future Directions

This work tackles the critical challenge of mitigating hallucinations in VLMs by introducing a novel TTA framework using RL. By updating only the lightweight layer normalization gamma parameters and incorporating a customized CLIP-based hallucination evaluation model, our approach effectively reduces object hallucinations during inference. Experiments demonstrate substantial performance improvements across multiple metrics on SOTA VLMs, showcasing the framework's robustness and efficiency.

In future work, we aim to extend this framework to attribute and relationship hallucinations, leveraging the capabilities of the customized CLIP model for a more comprehensive solution. Additionally, we plan to explore its application in discriminative tasks and adversarial training setups, where the VLM acts as a generative model and the evaluation model serves as a discriminator. These advancements may further enhance hallucination mitigation in VLMs, broadening their real-world applicability.

## References

[1] F. Zhao, C. Zhang, and B. Geng, "Deep multimodal data fusion," *ACM Computing Surveys*, vol. 56, no. 9, pp. 1–36, 2024.

[2] H. Liu, W. Xue, Y. Chen, D. Chen, X. Zhao, K. Wang, L. Hou, R. Li, and W. Peng, "A survey on hallucination in large vision-language models," *arXiv preprint arXiv:2402.00253*, 2024.

[3] J. Xu, X. Sun, Z. Zhang, G. Zhao, and J. Lin, "Understanding and improving layer normalization," *Advances in neural information processing systems*, vol. 32, 2019.

[4] S. Leng, H. Zhang, G. Chen, X. Li, S. Lu, C. Miao, and L. Bing, "Mitigating object hallucinations in large vision-language models through visual contrastive decoding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13 872–13 882.

[5] OpenAI, "Chatgpt (mar 14 version)," 2023, large language model. [Online]. Available: https://chat.openai.com

[6] G. DeepMind, "Gemini (dec 1 version)," 2023, large language model. [Online]. Available: https://gemini.deepmind.com

[7] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," 2023.

[8] J. Huang, J. Zhang, K. Jiang, H. Qiu, and S. Lu, "Visual instruction tuning towards general-purpose multimodal model: A survey," *arXiv preprint arXiv:2312.16602*, 2023.

[9] F. Zhao, J. Chen, B. Huang, C. Zhang, and G. Warner, "Checkguard: Advancing stolen check detection with a cross-modal image-text benchmark dataset," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 5425–5429.

[10] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, "Llava-next: Improved reasoning, ocr, and world knowledge," January 2024. [Online]. Available: https://llava-vl.github.io/blog/2024-01-30-llava-next/

[11] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, B. Li, P. Luo, T. Lu, Y. Qiao, and J. Dai, "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," *arXiv preprint arXiv:2312.14238*, 2023.

[12] S. Liu, L. Fan, E. Johns, Z. Yu, C. Xiao, and A. Anandkumar, "Prismer: A vision-language model with an ensemble of experts," *arXiv preprint arXiv:2303.02506*, vol. 3, 2023.

[13] J. Chen, D. Yang, T. Wu, Y. Jiang, X. Hou, M. Li, S. Wang, D. Xiao, K. Li, and L. Zhang, "Detecting and evaluating medical hallucinations in large vision language models," *arXiv preprint arXiv:2406.10185*, 2024.

[14] S. Zhao, X. Wang, L. Zhu, and Y. Yang, "Test-time adaptation with clip reward for zero-shot generalization in vision-language models," *arXiv preprint arXiv:2305.18010*, 2023.

[15] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," *arXiv preprint arXiv:2111.09734*, 2021.

[16] C.-E. J. Yu, B. Jalaian, and N. D. Bastian, "Mitigating large vision-language model hallucination at post-hoc via multi-agent system," in *Proceedings of the AAAI Symposium Series*, vol. 4, no. 1, 2024, pp. 110–113.

[17] W. Xiao, Z. Huang, L. Gan, W. He, H. Li, Z. Yu, H. Jiang, F. Wu, and L. Zhu, "Detecting and mitigating hallucination in large vision language models via fine-grained ai feedback," *arXiv preprint arXiv:2404.14233*, 2024.

[18] L. Zhao, Y. Deng, W. Zhang, and Q. Gu, "Mitigating object hallucination in large vision-language models via classifier-free guidance," *arXiv preprint arXiv:2402.08680*, 2024.

[19] C. Meister, T. Vieira, and R. Cotterell, "Best-first beam search," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 795–809, 2020. [Online]. Available: https://aclanthology.org/2020.tacl-1.51

[20] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[21] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[23] J. Wang, Y. Wang, G. Xu, J. Zhang, Y. Gu, H. Jia, M. Yan, J. Zhang, and J. Sang, “An llm-free multi-dimensional benchmark for mllms hallucination evaluation,” *arXiv preprint arXiv:2311.07397*, 2023.

[24] V. Singla, K. Yue, S. Paul, R. Shirkavand, M. Jayawardhana, A. Ganjdanesh, H. Huang, A. Bhatele, G. Somepalli, and T. Goldstein, “From pixels to prose: A large dataset of dense image captions,” *arXiv preprint arXiv:2406.10328*, 2024.

[25] M. AI, “Llama 3.2 multimodal (version 2023),” 2023, large language model. [Online]. Available: https://ai.meta.com/llama