Hiding-in-Plain-Sight (HiPS) Attack on CLIP for Targetted Object Removal from Images

Arka Daw[1]* **Megan Hong-Thanh Chung**[1]* **Maria Mahbub**[1] **Amir Sadovnik**[1]
[1] Oak Ridge National Laboratory (ORNL)

Abstract

Machine learning models are known to be vulnerable to adversarial attacks, but traditional attacks have mostly focused on single-modalities. With the rise of large multi-modal models (LMMs) like CLIP, which combine vision and language capabilities, new vulnerabilities have emerged. However, prior work in multimodal targeted attacks aim to completely change the model's output to what the adversary wants. In many realistic scenarios, an adversary might seek to make *only subtle modifications to the output*, so that the changes go unnoticed by downstream models or even by humans. We introduce *Hiding-in-Plain-Sight (HiPS)* attacks, a novel class of adversarial attacks that subtly modifies model predictions by selectively concealing target object(s), as if the target object was absent from the scene. We propose two HiPS attack variants, HiPS-cls and HiPS-cap, and demonstrate their effectiveness in transferring to downstream image captioning models, such as CLIP-Cap, for targeted object removal from image captions.

1 Introduction

The vulnerability of machine learning (ML) models to adversarial attacks—small perturbations in input data that lead to incorrect predictions—has been extensively studied [1, 2] across various domains, including image classification [3, 4] and biometrics [5, 6]. However, most existing adversarial attacks have been designed for single modalities, primarily focusing on the image or, less frequently, the text domain [7, 8]. The advent of large foundational models, such as large language models (LLMs) [9] and large multi-modal models (LMMs) [10] (e.g., Chat-GPT [11], Gemini [12]), which have shown great promise across a diverse range of tasks [13] (such as zero-shot classification, visual question answering, and image captioning) has revolutionized the ML community and led to their widespread adoption. Many of these models [14] integrate a pre-trained LLM with a large vision encoder, such as CLIP [15] which is a foundational multimodal model trained on 400M image-text pairs via contrastive learning. Typically, the vision encoder of such LMMs remains frozen during training, and the vision embeddings are mapped into the shared embedding space of the LLM using a simple projection layer. However, this introduces an obvious vulnerability [16]: adversarial attacks developed against these open-source vision encoders (such as CLIP) can be directly transferred to LMMs, compromising their integrity.

While generating adversarial attacks on LMMs (such as CLIP) [16, 17] has been explored, generally termed as *jailbreaking LMMs* [18, 19], they have primarily focused on 'completely' changing the output of the LMM to what the adversary wants, which is typically very different from the original outputs (without any perturbation). However, in many real-world scenarios, an adversary might seek to make only 'subtle' modifications to the output, so that the changes go unnoticed by downstream models or even by humans. To this end, we introduce a novel class of adversarial attacks on images, termed *Hiding-in-Plain-Sight (HiPS)* attacks. The primary goal of a HiPS attack is to generate an adversarial image that subtly modifies the model's predictions by selectively concealing a specific target object while leaving the rest of the model's functionality intact. For example, a HiPS adversarial image designed to hide a particular object should cause an image captioning model to generate a

[*] Equal Contribution

AdvML-Frontiers'24: The 3nd Workshop on New Frontiers in Adversarial Machine Learning@NeurIPS'24, Vancouver, CA.

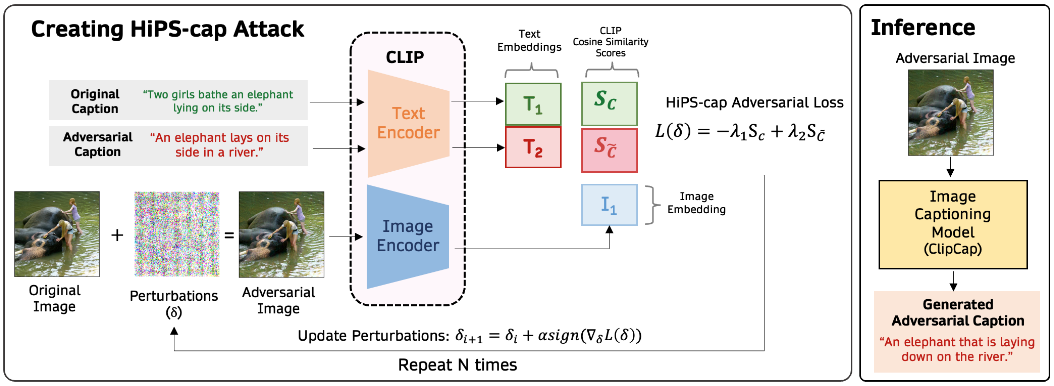

Creating HiPS-cap Attack
CLIP
Text Embeddings
CLIP Cosine Similarity Scores
Original Caption
"Two girls bathe an elephant lying on its side."
Adversarial Caption
"An elephant lays on its side in a river."
Text Encoder
Image Encoder
T_1
T_2
S_C
$S_{\tilde{C}}$
I_1
Image Embedding
HiPS-cap Adversarial Loss
$L(\delta) = -\lambda_1 S_c + \lambda_2 S_{\tilde{C}}$
Original Image
+
Perturbations (δ)
=
Adversarial Image
Update Perturbations: $\delta_{i+1} = \delta_i + \alpha sign(\nabla_\delta L(\delta))$
Repeat N times
Inference
Adversarial Image
Image Captioning Model (ClipCap)
Generated Adversarial Caption
"An elephant that is laying down on the river."

Figure 1: A schematic illustration of the *Hiding-in-Plain-Sight* (HiPS-cap) Attack.

caption as if the target object(s) was never present, while the rest of the image content should stay in tact. We propose two distinct types of HiPS attacks using the CLIP vision encoder: (1) HiPS-cls, which generates the attack by leveraging only the class label information, and (2) HiPS-cap, which utilizes the original image caption and a target caption to craft the attack. We demonstrate that our HiPS attacks can effectively transfer to downstream image captioning models, such as CLIP-Cap [20], enabling selective removal of target objects from image captions. Additionally, we introduce several novel evaluation metrics to assess the performance of our proposed HiPS attacks in targeted object removal.

2 Background and Related Works

Adversarial Robustness: One of the seminal methods for generating adversarial attacks is the Fast Gradient Sign Method (FGSM) [1], a simple, single-step L_∞-bounded attack, defined as: $I_{\text{adv}} = I + \epsilon \text{sign}(\nabla_I \mathcal{L}(I, y))$, where I_{adv} is the adversarial image, I is the original image, ϵ is the attack budget, and $\mathcal{L}$ is the loss function to be maximized for the attack. For an untargeted attack, $\mathcal{L}$ is typically the cross-entropy loss with respect to the correct class y, and in a targeted attack, the objective shifts to minimizing the loss with respect to a target class $\tilde{y}$, making $\mathcal{L}$ the negative cross-entropy loss for the target class. Another widely used technique is the Projected Gradient Descent (PGD) attack [2], which is the strongest first-order attack. PGD is an iterative, first-order optimization-based attack, defined as: $I^{t+1} = \mathcal{P}_{I+\mathcal{S}}(I^t + \alpha \text{sign}(\nabla_I \mathcal{L}(I, y)))$, where t denotes the iteration number, $\mathcal{P}$ is a projection operation that maps the perturbed input back onto a L_p ball with radius ϵ, with $\mathcal{S}$ representing the region defined by the L_p ball, and α is the step size.

Multi-modal Models: CLIP [15] is one of the seminal works in multi-modal modeling due to its exceptional performance in zero-shot tasks. Recently, there has been growing popularity in developing large multi-modal models (LMMs) [10] (GPT-4V [11], Gemini [12], LLaVA [14]) driven by their impressive capabilities across a wide range of tasks and domains [13]. Many of these models integrate a pre-trained large language model (LLM), such as Llama [21] or Vicuna [22], with a large vision encoder like CLIP. For LLaVA, the vision encoder remains frozen during training, with a simple projection layer mapping the vision embeddings to the shared embedding space of the LLM.

Adversarial Robustness of LMMs: With the advent of LMMs, investigating their vulnerabilities has become an important research focus in AdvML, often referred to as *jailbreaking* LLMs and LMMs [19, 18]. While previous studies have demonstrated that jailbreaking LLMs is feasible with full access to model parameters, recent findings highlight that LMMs are particularly susceptible to adversarial attacks targeting the vision modality [23]. In particular, even with access solely to the vision encoder, such as the open-sourced CLIP model–adversaries can exploit these vulnerabilities to jailbreak LMMs like LLaVA and OpenFlamingo[24], which rely on the frozen CLIP vision encoder.

3 Hiding-in-Plain-Sight (HiPS) Attack

Traditional 'targeted' adversarial attacks on images are designed to drastically alter the behavior of a downstream ML models (such as a Large Vision Language Model or an image classifier), forcing them to produce outputs that align with the adversary's objectives. In contrast, we introduce a novel class of adversarial attacks on images, termed *Hiding-in-Plain-Sight (HiPS)* attacks. The primary goal of a HiPS attack is to generate an adversarial image that can *'subtly'* modify the model(s) predictions by selectively concealing a specific 'target' object while leaving the rest of the model's functionality intact. For instance, a HiPS adversarial image designed to conceal (or 'target') a particular object should cause an image captioning model to generate a caption as if the target object(s) was never present while the rest of the image content should stay in tact. Similarly, when a HiPS adversarial image is processed by a LMM, the model should respond to queries about the image as if the target object were absent. Ideally, the adversarial images generated using the HiPS attacks should be universal and transferable across a variety of downstream ML models. Therefore, generating the HiPS attack using a *'foundation'* multi-modal model that is already universally used for a variety of downstream tasks is necessary for transferability. For simplicity, in this paper, we focus on investigating the transferability of HiPS attacks specifically on image captioning models [25].

3.1 Problem Formulation

In this section, we introduce the formal notations used throughout this paper. Let I represent an input image containing n different object classes, and let $\mathcal{T}_I = \{T_1, T_2, \dots, T_n\}$ denote the set of objects present in the image I, where T_i corresponds to the textual description (or simply, the class labels) of the i-th object. The target object to be removed is denoted as $T_{\text{target}} = T_j$ for some $j \in \{1, 2, \dots, n\}$. We will utilize the CLIP model to generate the HiPS attack, which consists of an image encoder, $f_{\texttt{Image}} : I \rightarrow \mathbf{Z}_{\texttt{Image}}$, and a text encoder, $f_{\texttt{Text}} : T \rightarrow \mathbf{Z}_{\texttt{Text}}$, where T is a textual input, $\mathbf{Z}_{\texttt{Image}} \in \mathbb{R}^D$ is the image embedding, $\mathbf{Z}_{\texttt{Text}} \in \mathbb{R}^D$ is the text embedding, and D is the embedding dimension.

In the context of image captioning, the objective of the HiPS attack is to generate an adversarial image I_{adv} that is nearly indistinguishable from I. However, when this adversarial image I_{adv} is processed by the downstream image captioning model, $f_{\text{caption}} : I \rightarrow T$, the generated caption $\hat{C}_{\text{adv}} = f_{\text{caption}}(I_{\text{adv}})$ should omit the target object T_{target} while accurately describing all other objects in the image. In other words, the adversarially generated caption, $\hat{C}_{\text{adv}}$, should *closely resemble* the caption $\hat{C}_{\text{orig}}$ produced from the original (unperturbed) image, with the exception that T_{target} is not mentioned. This approach contrasts sharply with traditional targeted attacks, where typically the goal is to produce an output that is significantly different from the correct one. While a traditional adversarial attack's goal is to make the perturbation imperceptible in the *input space*, in the HiPS attack, we want the difference in the output space to also be minimal - the only difference should be the omission of the target class. In this paper, we propose two different variants of the HiPS attack in the context of image captioning, which are detailed below (See Figure 1 for a schematic representation of HiPS Attack).

3.2 HiPS-cls Attack using Class Labels

In this variant of HiPS attack, termed *HiPS-cls*, we utilize only the textual class labels $\mathcal{T}_I$ to obtain the adversarial image. Given an image I and its corresponding set of class labels $\mathcal{T}_I$, we compute the cosine similarity scores S_i for each class label T_i as follows:

$$S_i = \cos(f_{\texttt{Image}}(I), f_{\texttt{Text}}(T_i)) = \left\langle \frac{f_{\texttt{Image}}(I)}{\|f_{\texttt{Image}}(I)\|_2}, \frac{f_{\texttt{Text}}(T_i)}{\|f_{\texttt{Text}}(T_i)\|_2} \right\rangle \quad (1)$$

The cosine similarity S_i between the image I and class label T_i measures the alignment between their respective image and text embeddings. A higher score S_i indicates that the object with class label T_i is likely present in the image I, while a lower score suggests its absence. Since the objective of the HiPS attack is to remove the target object $T_{\text{target}} = T_j$, our goal is to perturb the image I in such a way that the cosine similarity score for the target object, S_j, is reduced (as if it is absent), while the scores for all other objects T_i (for all $i \neq j$) are either increased or remain unchanged. To formalize this, we define the HiPS-cls adversarial loss function as follows: $\mathcal{L}_{\text{HiPS-cls}} = -\lambda_1 S_j + \lambda_2 \sum_{i \neq j} S_i$.

3.3 HiPS-cap Attack using Adversarial Captions

In this variant of the HiPS attack, termed *HiPS-cap*, rather than using class labels, we generate the attack on CLIP by utilizing the original caption C and a target caption $\tilde{C}$. The target caption $\tilde{C}$ is designed to be similar to C, but as if the target object T_{target} were not present in the image. In other words, $\tilde{C}$ represents an ideal adversarial caption that a successful HiPS attack on a captioning model should produce. Similar to the HiPS-cls approach, we calculate the cosine similarities between the image I and both the original caption C and the target caption $\tilde{C}$ as follows:

$$S_C = \cos(f_{\texttt{Image}}(I), f_{\texttt{Text}}(C)); \qquad S_{\tilde{C}} = \cos(f_{\texttt{Image}}(I), f_{\texttt{Text}}(\tilde{C})) \tag{2}$$

The corresponding adversarial loss can be computed as $\mathcal{L}_{\text{HiPS-cap}} = -\lambda_1 S_C + \lambda_2 S_{\tilde{C}}$. $\mathcal{L}_{\text{HiPS-cap}}$ aims to reduce the score for the original caption S_C while increase the score for the target caption $S_{\tilde{C}}$, where the target object is missing. The adversarial loss $\mathcal{L}_{\text{HiPS-cls}}$ and $\mathcal{L}_{\text{HiPS-cap}}$ can be optimized using existing adversarial attacks such as FGSM and PGD attacks (See Section 2).

4 Experimental Setup

Setting: We develop HiPS-cls and HiPS-cap attacks using the CLIP model, where the vision encoder is based on Vision Transformer architecture (ViT-B/32) [26]. To generate adversarial images for the HiPS attack, we employ established techniques, including FGSM and PGD with L_∞, L_1, and L_2 norms. For simplicity, we focus on images containing only two foreground objects: one serving as the target object to be removed, and the other as the object to be retained in the adversarial caption. We manually sampled 50 such images from the MS COCO dataset to test our two HiPS attack variants (cap vs. cls). For the HiPS-cap attack, we use the original COCO captions as C, and manually generate two target (adversarial) captions, one used for training ($\tilde{C}$), while the other one is reserved for evaluation. For the downstream captioning model, we utilize the CLIP-Cap [20] model. CLIP-Cap uses the vision encoder from CLIP and a mapping network to project the image embeddings into a shared representation space, where a language model (GPT-2) [27] generates the captions.

Evaluation Metrics: In the context of assessing the success of HiPS attack, we introduce several novel metric to measure attack success, where we consider two main criterions. First, the ability to successfully remove references of the target object from the generated textual caption. We propose a metric called *Target Object Removal Rate (TORR)* to capture this using similarity-based assessments and string-matching comparisons between words. Second, the ability to measure if the remaining objects are intact to ensure that perturbation does not inadvertently affect or remove references to the objects other than the targeted one. We propose another metric called *Remaining Objects Retention Rate (RORR)* for this purpose. Next, we utilize Attack Success Rate (ASR) that measures if both of these criterions (TORR and RORR) are satisfied. We additionally utilize Caption Semantic Similarity (CSS) which is essentially the cosine similarity between the ground truth adversarial caption, and the generated adversarial caption ($\cos(\tilde{C}_{gt}, \hat{C}_{\text{adv}})$). CSS measures if the two are semantically close to each other in the text embedding space. Additional details of computation of TORR, RORR and ASR are provided in the Appendix. The image quality is another important metric to measure the imperceptibility of the attack. We use standard metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Signal-to-Noise Ratio (PSNR), and Structural Similarity metric (SSIM).

Baselines: We compare against two PGD (L_∞) based attacks: targeted and untargeted. For the class-labels variant, in the PGD (targeted) setting, we set $\lambda_1 = 1$, $\lambda_2 = 0$, focusing solely on removing the target object. In the PGD (untargeted) setting, we set $\lambda_1 = 0$, $\lambda_2 = 1$, prioritizing the retention of all other objects in the image. For adversarial captions variant, we only use PGD (targetted) setting, where we optimize to maximize the similarity with target caption ($\lambda_1 = 0$, $\lambda_2 = 1$).

5 Results

Quantitative Evaluation of HiPS-cls and HiPS-cap: In Tables 1 and 2, we compare the attack success and image quality metrics of the two HiPS variants, using FGSM and PGD under L_∞, L_1, and L_2 norm constraints. We report results for the best-performing model in each case (see hyper-parameter settings in Appendix). FGSM performs poorly across both HiPS variants, achieving an ASR of only 36-38%. In contrast, the various PGD attacks demonstrate strong performance across both variants, with the L_∞ variant slightly outperforming the L_1 and L_2 norms. Specifically, for the HiPS-cls attack, PGD achieves 100% RORR, indicating that the adversarial captions consistently retain the non-target objects, and TORR of 90% or higher, demonstrating effective removal of the target object from the caption. However, the TORR, RORR, and ASR metrics for HiPS-cap are slightly lower than those for HiPS-cls (across all PGD attack norms), while the CSS is significantly higher. This indicates that while HiPS-cls is more effective at retaining non-target objects and removing target objects, but it produces lower-quality captions, often resulting in grammatical errors and introducing unnatural artifacts that negatively impact CSS (See Figure 2 for a qualitative comparison of different methods). Further, we observe that the baseline method PGD (targeted) achieves a high TORR of 90% but a relatively low RORR of 78%, indicating that it primarily optimizes for the removal of the target object. In contrast, PGD (untargeted) exhibits the opposite trend, prioritizing the retention of non-target objects, achieving a 100% RORR and 76% TORR. For PGD (targeted) using the adversarial caption, we observe that solely optimizing for similarity to the adversarial caption results in a lower 66% ASR. This occurs because the adversarial caption is, by definition, already similar to the original caption, making it less effective at removing the target object. To achieve a higher TORR, it is necessary to move away from the original caption (i.e., $\lambda_1 > 0$). Additionally, we compare the image quality metrics in Table 2. As expected, FGSM utilizes the entire attack budget and introduces significantly larger perturbations compared to the iterative PGD attack. We also observe that the perturbations introduced by PGD under the L_1 norm are slightly smaller when compared to L_2 and L_∞, as the L_1 norm favors localized perturbations with minimal overall change.

Table 1: Comparison of attack success metrics for HiPS-cls and HiPS-cap attacks, optimized using FGSM and PGD under L_∞, L_1, and L_2 norm constraints.

	HiPS-cls (Class Labels)				HiPS-cap (Adv. Caption)			
	TORR ↑	RORR ↑	ASR ↑	CSS ↑	TORR ↑	RORR ↑	ASR ↑	CSS ↑
FGSM	38.0	98.0	36.0	0.6907	40.0	96.0	38.0	0.7066
PGD (L_1)	88.0	**100.0**	88.0	0.6898	84.0	**98.0**	84.0	0.7578
PGD (L_2)	90.0	**100.0**	90.0	0.6701	88.0	96.0	86.0	0.7546
PGD (L_∞)	**94.0**	**100.0**	**94.0**	**0.6901**	**90.0**	**98.0**	**90.0**	**0.7673**
PGD (untarget)	76.0	**100.0**	76.0	0.6790	**-**	**-**	**-**	**-**
PGD (target)	90.0	78.0	72.0	0.6111	66.0	100.0	66.0	0.7499

Table 2: Comparison of image quality metrics for HiPS-cls and HiPS-cap attacks, optimized using FGSM and PGD under L_∞, L_1, and L_2 norm constraints.

	HiPS-cls (Class Labels)				HiPS-cap (Adv. Caption)			
	MSE ↓	MAE ↓	PSNR ↑	SSIM ↑	MSE ↓	MAE ↓	PSNR ↑	SSIM ↑
FGSM	61.56	7.81	30.25	79.39	61.56	7.81	30.25	79.46
PGD (L_∞)	12.32	3.03	37.2	94.84	23.04	3.69	34.62	92.23
PGD (L_1)	**6.67**	**1.73**	**39.91**	**97.02**	**6.46**	**1.73**	**40.05**	**97.14**
PGD (L_2)	11.33	2.23	37.60	95.32	35.36	4.18	32.67	88.93

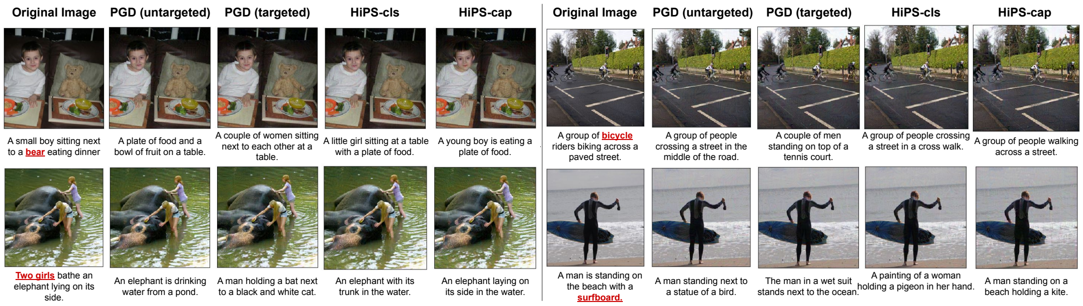

Original Image
PGD (untargeted)
PGD (targeted)
HiPS-cls
HiPS-cap
A small boy sitting next to a bear eating dinner
A plate of food and a bowl of fruit on a table.
A couple of women sitting next to each other at a table.
A little girl sitting at a table with a plate of food.
A young boy is eating a plate of food.
Two girls bathe an elephant lying on its side.
An elephant is drinking water from a pond.
A man holding a bat next to a black and white cat.
An elephant with its trunk in the water.
An elephant laying on its side in the water.
Original Image
PGD (untargeted)
PGD (targeted)
HiPS-cls
HiPS-cap
A group of bicycle riders biking across a paved street.
A group of people crossing a street in the middle of the road.
A couple of men standing on top of a tennis court.
A group of people crossing a street in a cross walk.
A group of people walking across a street.
A man is standing on the beach with a surfboard.
A man standing next to a statue of a bird.
The man in a wet suit stands next to the ocean.
A painting of a woman holding a pigeon in her hand.
A man standing on a beach holding a kite.

Figure 2: Qualitative Results comparing various methods with target shown as red words of caption.

Effect of Attack Budget: Figure 3 illustrates the impact of the attack budget ϵ on various attack success metrics for HiPS-cls and HiPS-cap attacks using FGSM and PGD with the L_∞ norm. Consistent with previous observations, FGSM performs significantly worse than PGD across all attack budgets for both HiPS variants. For PGD attacks, as expected, increasing the ϵ value leads to an improvement in the ASR up to $\epsilon = 0.05$, after which the ASR gradually saturates, with HiPS-cls

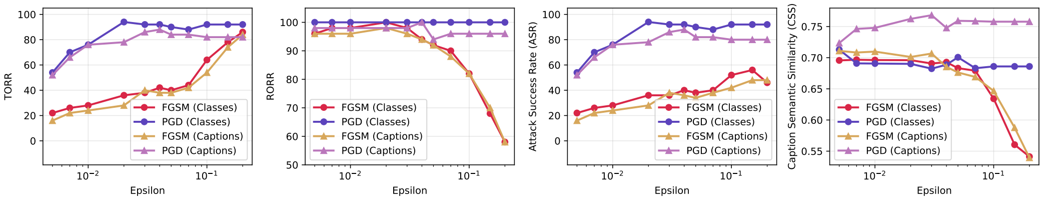
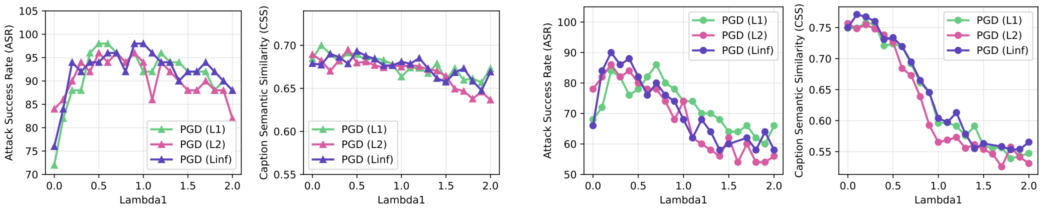


Figure 3: Comparing the effect of attack budget ϵ on the different attack success metrics for HiPS-cls and HiPS-cap attacks using FGSM and PGD with L_∞ norm.

showing slightly better performance than HiPS-cap. However, it is noteworthy that as ϵ increases, the CSS for HiPS-cls drops sharply, whereas HiPS-cap maintains a relatively stable CSS around 0.75. Image quality metrics for the various attack budgets are presented in Appendix Figure 5.



(a) HiPS-cls (Class Labels)

(b) HiPS-cap (Adversarial Captions)

Figure 4: Comparing the sensitivity of hyperparameter λ_1 on HiPS-cls and HiPS-cap attacks.

Sensitivity to Lambda: Figure 4 illustrates the effect of the hyperparameter λ_1 on the performance of different HiPS attacks while keeping λ_2 fixed at 1.0 for different HiPS PGD variants. We observe that as the magnitude of λ_1 increases in HiPS-cls, both the ASR and CSS remain relatively high (greater than 90%) across a wide range of λ_1 values, from 0.3 to 1.8, across all PGD variants. This stability is due to the increased emphasis on removing the target object as λ_1 increases. In contrast, for HiPS-cap, increasing λ_1 places greater focus on reducing the score of the original caption while maintaining the weight of the adversarial caption. As a result, when λ_1 exceeds 0.5, both ASR and CSS decline rapidly. See Appendix Figure 6 and 7 for TORR, RORR and image quality metrics.

6 Limitations, Discussion and Conclusion

In this work, we demonstrate promising results for the HiPS attack, showing that it is possible to generate small perturbations which cause subtle differences in the output of a downstream task. However, we recognize a few current limitations and future work needed to overcome them. First, the metrics we use to measure success are not always correct. For example, the rule based metrics (TORR and RORR) are biased towards only detecting the presence and absence of an object(s) from a text caption, and does not consider if the sentence is grammatically correct or if additional objects were added to the caption even though they do no exist in the image. In our small dataset this occurs infrequently but can skew the results more on a large dataset. In addition, we find that the cosine similarity metric is not precise enough to measure the small differences between cosine similarities of our caption since they are all very close to each other (by design). In the future, we plan on using a LMM to evaluate the results in a more accurate manner using custom prompts. In addition, in this work we our experiment was restricted to 50 images due to the manual annotations required for generating two adversarial captions for each image. We plan to conduct much larger experiments in the future by automating this process using LMMs, prompting the model to generate a caption with the target object missing. Finally, while in this work we focused on a single image captioning model, we believe that this attack can be used for other multimodal models (LLaVa, OpenFlamingo) as well as other downstream tasks (object detection, action recognition). The transferability of the attack can be improved using an ensemble of multimodal models to generate the attack.

Acknowledgments

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (https://www.energy.gov/doe-public-access-plan).

References

[1] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[2] Aleksander Madry. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[3] Gabriel Resende Machado, Eugênio Silva, and Ronaldo Ribeiro Goldschmidt. Adversarial machine learning in image classification: A survey toward the defender's perspective. *ACM Computing Surveys (CSUR)*, 55(1):1–38, 2021.

[4] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

[5] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[6] Shuai Jia, Bangjie Yin, Taiping Yao, Shouhong Ding, Chunhua Shen, Xiaokang Yang, and Chao Ma. Adv-attribute: Inconspicuous and transferable adversarial attack on face recognition. *Advances in Neural Information Processing Systems*, 35:34136–34147, 2022.

[7] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.

[8] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*, 2017.

[9] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[10] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023.

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

[12] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[13] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[14] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[16] Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. Plug and pray: Exploiting off-the-shelf components of multi-modal models. *arXiv preprint arXiv:2307.14539*, 2023.

[17] Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. Robust clip: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. *arXiv preprint arXiv:2402.12336*, 2024.

[18] Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. Cold-attack: Jailbreaking llms with stealthiness and controllability. *arXiv preprint arXiv:2402.08679*, 2024.

[19] Weikang Zhou, Xiao Wang, Limao Xiong, Han Xia, Yingshuang Gu, Mingxu Chai, Fukang Zhu, Caishuang Huang, Shihan Dou, Zhiheng Xi, et al. Easyjailbreak: A unified framework for jailbreaking large language models. *arXiv preprint arXiv:2403.12171*, 2024.

[20] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[21] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[22] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

[23] Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[24] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

[25] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: A survey on deep learning-based image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):539–559, 2022.

[26] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[28] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spacy: Industrial-strength natural language processing in python, 2020.

[29] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

[30] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[31] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

A Additional Details of Evaluation Metrics

Attack Success Rate (ASR) metric is an aggregated evaluation of the success of modifying textual captions to remove references to a target object while preserving mentions of remaining objects. The aggregation constitutes of two measures: (1) Target Object Removal Rate (TORR) and (2) Remaining Objects Retention Rate (RORR). TORR assesses whether references to a specific target object T_{target} are effectively removed from the caption generated after the HiPS attack $\tilde{C}$, measuring how well the perturbations obscure T_{target} from the model's point-of-view. RORR, on the other hand, evaluates whether references to the remaining objects T_i are preserved in $\tilde{C}$. This ensures that the perturbation does not inadvertently affect or remove references to the objects other than the targeted one.

We calculate the TORR and RORR metrics in two steps: (i) word segmentation and cleaning, (ii) semantic presence validation.

(i) Word segmentation and cleaning: To identify object references within $\tilde{C}$, we tokenize words using $spaCy$ [28], yielding a list of words $W_{\tilde{C}}$. We assume that specific words in $W_{\tilde{C}}$ correspond to object representations. We remove stop words and punctuation as they do not contribute to our evaluation schema and help streamline the analysis by focusing on meaningful words that are critical to understanding the content of $\tilde{C}$. We convert plurals in $W_{\tilde{C}}$ to singular forms using the $inflect$ engine in Python. This normalization aids in matching terms more effectively, when compared to T_{target} and T_i, during semantic presence validation. Finally, we filter $W_{\tilde{C}}$ based on Part-of-Speech tags using $spaCy$. We exclude determiners (DET) and pronouns (PRON) from our analysis as they do not bear any significance to our analysis. This extensive processing within this step is integral to transforming $\tilde{C}$ into a refined set of lexical units that accurately represent its meaningful content. It allows the ASR metric to perform precise evaluations of object presence, enhancing the accuracy and validity of the analysis and reducing the risk of misinterpretations due to irrelevant or misleading text components.

(ii) Semantic presence validation: Provided $W_{\tilde{C}}$ from step (i), we verify the absence of T_{target} and the presence of T_i. This step involves both direct presence checks and similarity-based assessments. For direct presence check, we perform string-matching comparisons between $W_{\tilde{C}}$ and T_{target}, as well as between $W_{\tilde{C}}$ and T_i. If the direct presence check does not yield a clear result, we employ cosine similarity between word embeddings to further validate the success of T_{target} removal and T_i retention. Using an empirically established similarity threshold (0.7 in this case), we determine the boundary for distinguishing between successful and unsuccessful removal/retention of the objects. Our ASR metric offers multiple options for obtaining word embedding, including Word2Vec [29], GloVe [30], FastText [31], and BERT [32], during semantic presence validation, enhancing its adaptability and effectiveness across various downstream tasks and models. For our specific application, we found GloVe to be the most effective choice.

The ASR metric, while robust for many scenarios, encounters challenges when dealing with multi-word objects, such as "teddy bear." In these cases, the metric may struggle to effectively assess the presence or removal of the entire phrase because it traditionally operates on individual word embeddings. Our workaround for this limitation involves averaging the embeddings for each word within the multi-word phrase. Furthermore, the metric's reliance on cosine similarity thresholds may not fully account for the nuanced differences between conceptually related but distinct objects and vice-verse. For example, while "hills" and "mountains" are closely related, they are not interchangeable. Our experiments show that the ASR metric might fail to recognize this subtle distinction, leading to rare but incorrect assessments of object removal success.

B Hyperparameter Details

We have used $\lambda_2 = 1$, in all of our experiments. Additional details of other hyperparameters are provided in Table 3.

C Additional Results

We present some additional results on image quality, TORR, and RORR for the effect of attack budget and hyperparameter sensititivity.

Table 3: Hyperparameter Details for Best Performing Models from Table 1

	HiPS-cls (Class Labels)		HiPS-cap (Adv. Caption)	
	α	ϵ	α	ϵ
FGSM	2/255	0.03	2/255	0.03
PGD (L_1)	500	1000	500	1000
PGD (L_2)	5	5	5	10
PGD (L_∞)	2/255	0.02	2/255	0.06

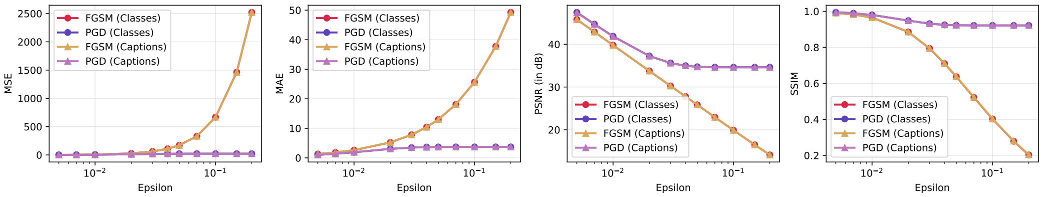
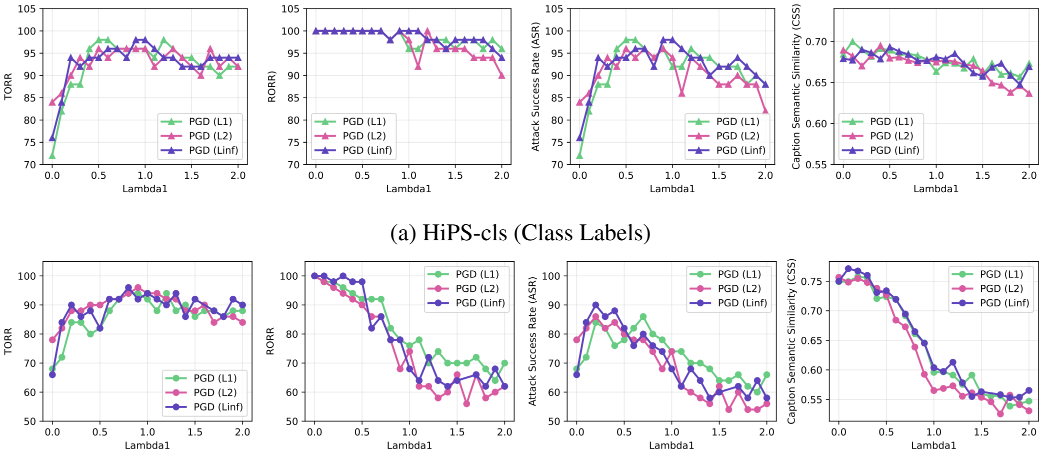


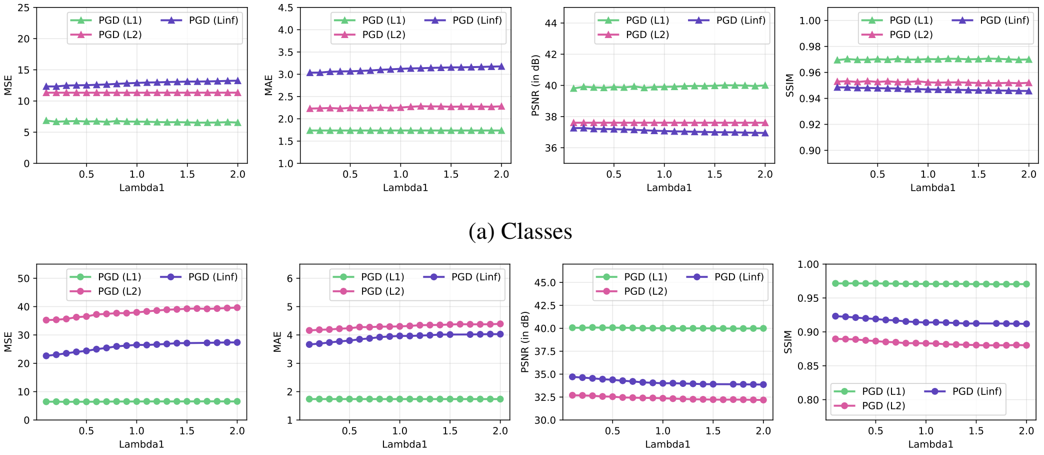
Figure 5: Comparing the effect of attack budget ϵ on the different image quality metrics for HiPS-cls and HiPS-cap attacks using FGSM and PGD with L_∞ norm.



(a) HiPS-cls (Class Labels)

(b) HiPS-cap (Adversarial Captions)

Figure 6: Comparing the sensitivity of hyperparameter λ_1 on HiPS-cls and HiPS-cap attacks.



(a) Classes

(b) Adversarial Captions

Figure 7: Comparing image quality metrics on the sensitivity of hyperparameter λ_1 on HiPS-cls and HiPS-cap attacks.