Language as its own Adversary for Vision-Language Models

Jameel Hassan, Sanoojan Baliah, Adham Ibrahim
22010301, 22010233, 22010323
{jameel.hassan, sanoojan.baliah, adham.ibrahim}@mbzuai.ac.ae

Abstract

Vision-language models have taken up a prominent place as pretrained models given their robustness to zero-shot performance. One such model is the CLIP model, which has been adopted as a foundation model for numerous downstream tasks. This renders the study of the susceptibilities and pitfalls of CLIP an indispensable topic. In this work, we show the bias on language(text) by the CLIP model by naively writing on the image, and extend this idea to an adversarial attack. Specifically, we design a generator model that creates an adversarial image conditioned on a corruption text to create visually indistinguishable adversarial samples. The adversarial attack is shown to reduce the model performance across different datasets and shows promise in targetted attacks on CIFAR10 and CIFAR100. This gives rise to the question, *Is language itself the downfall of vision-language models?*

1 Introduction

Pre-training methods from raw text, task-agnostic objectives such as autoregressive and masked language modelling have proven effective in the domain of NLP. This showed the possibility of aggregating supervision from web-scaled text over crowd sourced datasets in NLP. CLIP (1) was one of the pioneering efforts in extending this web-scaled collection of data in the domain of computer vision. Here, they trained a model using 400 million image and text caption pairs, manifesting text labels are strong supervision signals for computer vision tasks.

Earlier works in the domain of combining vision and text stemmed from using image meta data as a bag of words for classification (2). The research in (3) extended this by using n-gram models instead of individual words. However, the large scale training with a contrastive learning methodology gave CLIP excellent zero-shot performance on a wide range of datasets. This has made CLIP takeover as a foundation model for many downstream tasks.

The advent of CLIP as a foundation model poses certain crucial questions. How robust is CLIP? Are there weaknesses to it? If so, what kind of weaknesses are they? These questions are critical to address the possibility of adversarial attacks, which have been shown prevalent in deep learning models. Various forms of adversarial attacks such as on CNNs (4) Fig. 1, and more recently on the use of CLIP itself to design an adversarial attack (5) have been studied.

In this work, we design an adversarial attack targeted on the CLIP model. We specifically explore the use of language itself to craft the adversarial attack on CLIP. We make use of an encoder-decoder generator architecture that generates an adversary given a corruption text label, such that the adversary is visually indistinguishable from the original image. The key contributions are as follows:

1. The susceptibility of CLIP model for adversarial attack using pure text superimposed on the image.

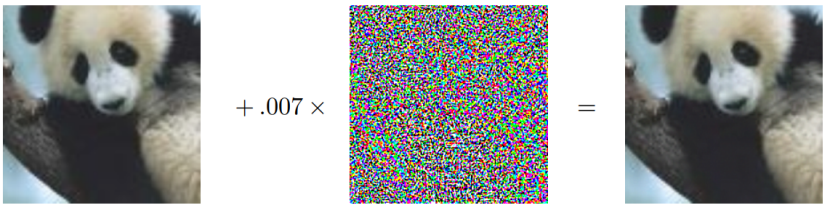

Figure 1: Adversarial attack example on CNNs (4)

2. A generator model designed using an encoder-decoder architecture, to generate visually indistinguishable adversarial images, using text as corruption input.
3. Extensive experiments to show that the designed generator can create targetted attack on the CLIP model, especially on the CIFAR-10 dataset.

2 Related works

Adversarial attacks on classifiers: Several works have explored the creation of targeted perturbations to fool models with image inputs. The work (6) noticed that neural networks misled using malicious perturbations, which remain imperceptible to humans. Many follow up works (4; 7; 8; 9) explore gradient ascent in pixel domain, solving complex optimizations to create adversarial attacks. However, these methods were data-dependent and the adversaries were crafted for each sample exclusively, rendering them inefficient at inference time.

Universal Adversarial Perturbations: The seminal work of (10) introduced the existence of Universal Adversarial Perturbation (UAP). This single noise vector was capable of deviating a model prediction from the correct prediction when added to the original input. While, this adversary was universal, the performance of the UAP was proportional to the number of training samples used for designing the UAP. A more robust data-agnostic approach that crafts adversarial samples directly from a generator is proposed by (11).

Generator based adversaries: Another branch of attacks utilize generator modesl to create adversarial attacks. (12) apply generative adversarial networks to craft visually realistic perturbations and build distilled network to perform black-box attack. Similarly, (13; 14) train generators to create adversaries to launch attacks; the former uses target data directly and the latter relies on class impressions.

Vision-and-Language models and adversarial attacks: Due to the robustness of zero-shot performance, vision and text pretrained models have been adopted to various downstream tasks (). V-L models provide high-quality aligned visual and textual representations learn from large scale image-text pairs. (5) utilizes a prominent such model - CLIP - in its design of an adversarial attack. On the contrary, we explore the same attribute of alignment between vision and text, to attack the CLIP model using text using a generative model.

3 Motivation

The robust nature of the CLIP model has led to an extensive use of CLIP in various tasks and projects. One such was "rclip" (15), that conducted an MS paint-style art competition. Providing various text prompts such as *"cute raccoon using a computer"* or *"world's most fabulous monster"*, players were asked to draw a painting. The paintings were scored based on the similarity of the CLIP model image embedding and the text prompt embedding. An interesting observation that arose was that paintings with text related to the prompt such as "raccoon" or "monster", written on them, boosted the score.

This observation was attributed towards the vision-language pretraining of CLIP, enabling it to understand the text. However, we pursue this observation under a different lens. We explore the

possibility of using such text additions as corruption (using misleading text) to flip the model predictions. The idea of this is shown in Figure. 2. Given the original image of the dog, the prediction of the model is the class "DOG" with over 99% confidence. But with the text "airplane" written on it, which is quite faint in the background, still leads the model to predict the class "AIRPLANE" with confidence of nearly 70%. We thus pose the question, *is language itself the downfall of CLIP, a vision-language model?*

Further analysis of this observation is described in Section 5. Hence, our hypothesis of creating a targeted attack using language itself can be shown to be valid. However, since this cannot be exploited as an adversarial attack as the image has been visually changed, we extend this idea through a generator model, to create an adversarial image that is indistinguishable from the original image visually but still is able to fool the CLIP model.

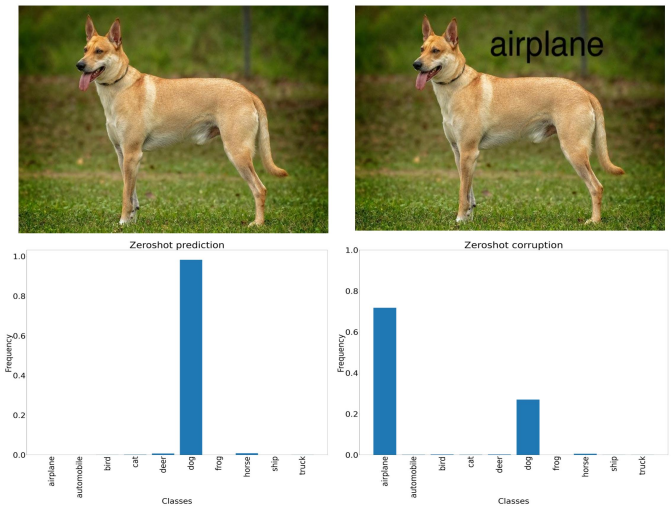

Figure 2: Comparison between original dog image and corrupted dog image with text of airplane. The CLIP model gets biased towards the airplane class after text corruption.

4 Methodology

In this section, we first introduce the CLIP Vision-Language model and its zero-shot prediction, and subsequently we introduce our language written adversary generation approach for the targetted white-box attack on CLIP models.

4.1 CLIP

Vision-language model CLIP is trained on large-scale image-text pairs to learn semantics-driven visual features. Importantly, these models provide “zero-shot" knowledge transfer. We represent CLIP visual and text encoders as f_v and f_t, respectively. Given the CLIP model design, we are able to use CLIP’s zero shot prediction since the class information can be fed to the network in the form of text. This is done by employing the template prompt (1), where text inputs are generated as $S = \{This\ is\ a\ photo\ of\ a\ \{class_i\}\}_{i=1}^{i=C}$, where C is the number of classes. We can then obtain the zero-shot outputs by computing the cosine similarity between the visual and textual features. The textual features of a class with highest similarity with visual output $f_v(x)$ is selected as the model’s final prediction. Formally, based on $\hat{y} = argmax(\langle f_v(\boldsymbol{x}), f_t(S)\rangle)$, we get the zero shot prediction.

4.2 Language written adversary generation

From our motivation in Section. 3, we show that the image created by writing a different label than the original ground truth i.e., the targeted corruption label, makes the prediction of CLIP-VIT-B/16 get biased towards the corruption label. However, such *language written image* is distinguishable as

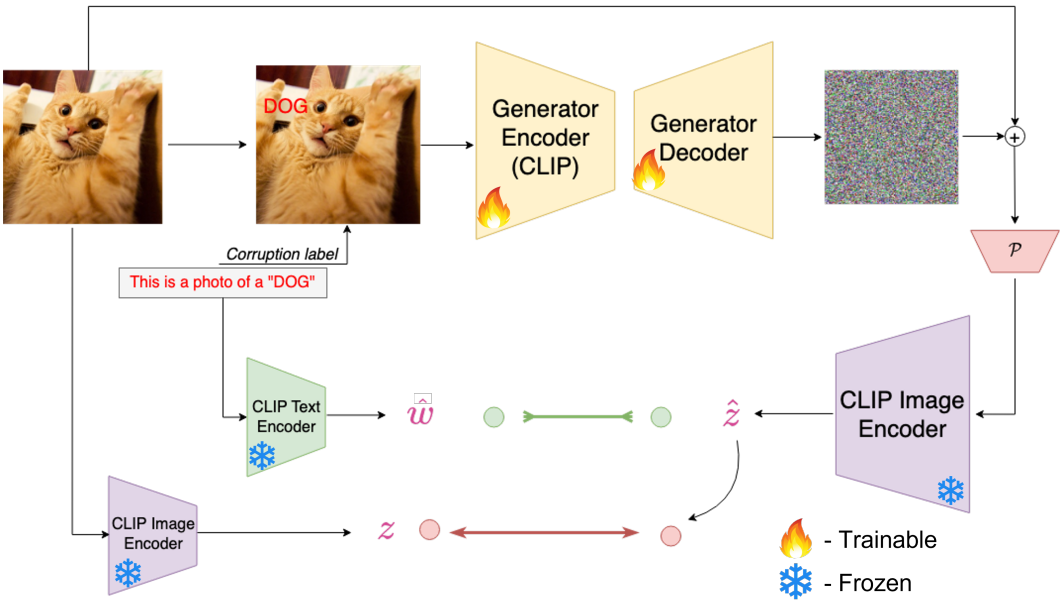

Figure 3: Adversarial image generation from language written image

an adversary by humans. To serve the purpose of an adversarial image from an attacker's point of view, which is indistinguishable from the original image by humans, we create the adversarial image from the *language written image*. We call this approach, "Language written adversary generation" and is described in this section.

As the initial step, the corrupted image $\tilde{x}$ is generated from the original image by writing the randomly chosen corruption label name $\tilde{l}$, which is different from the true label name l. We place the text with the text size s_l in a randomly chosen position and a randomly chosen color from the preset list of positions and colors respectively. The corrupted image is then fed to the generator model $\mathcal{G}$ with the encoder-decoder architecture. Then an L_∞ projection $\mathcal{P}$, with a perceptual budget ϵ, is applied on the addition of the original image and the noise $\tilde{n}$, generated from the generator model, to create the adversarial image $\hat{x}$. Mathematically, the adversarial image can be written as $\hat{x} = \mathcal{P}(\mathcal{G}(\tilde{x}) + x)$ and the corresponding adversarial text $\hat{s}$ is created from the template prompt discussed in the Section. 4.1 as $\hat{s} = \mathit{This\ is\ a\ photo\ of\ a}\ \{\tilde{l}\}$. We train only the generator model with the contrastive loss such that the adversarial visual feature $f_v(\hat{x})$ attracts corrupt label text feature $f_t(\hat{s})$ and repels the original image visual feature $f_v(x)$, where f is the CLIP-ViT-B/16 model - the white-box model that we attack. Formally the loss is formulated as,

$$\mathcal{L} = \mathcal{L}_{pos} + \mathcal{L}_{neg} \quad \texttt{where}, \tag{1}$$

$$\mathcal{L}_{pos} = 1 - (\langle f_v(\hat{x}), f_t(\hat{s}) \rangle) \tag{2}$$

$$\mathcal{L}_{neg} = max(0, (\langle f_v(\hat{x}), f_v(x) \rangle - \psi)), \quad \psi = \texttt{margin} \tag{3}$$

Inference: Our approach facilitates, the targeted adversarial attack from the single-trained generator model $\mathcal{G}$. At inference time, the target label is chosen and the adversarial image is created in the same manner as in the training steps. To our knowledge, there is little to no work that performs targeted adversarial attacks with a single-trained model. With our approach, we show this is feasible, especially on the highly generalizable CLIP model.

5 Experiments and Results

Datasets: We conduct experiments on CIFAR-10 (16), CIFAR-100, Caltech-101 (17). CIFAR-10 contains 10 classes, a total of 50000 training images, and 10000 test images with a resolution of 32×32. CIFAR-100 contains 100 classes, with a total of 50000 training images, and 10000 test

images with a resolution of 32×32. Caltech-101 consists of 101 classes with around 9000 images with the size of around 300×200.

Experiment protocols: We train the generator model with the AdamW optimizer with the default parameters except the learning rate chosen either $1e-3$ or $1e-5$. The font size s_l, is set such that $(5/32) \times h$, where h is the height of the image. We use 32 as batch size and trained the models for 20 epochs. We set the perceptual budget ϵ in the L_∞ projection as 0.1 and the margin $\psi = 0.2$ in our loss formulation.

5.1 Extended Results of Motivation

We extend the observation shown through Figure 2 with further analysis. We obtain the zero-shot predictions of the CLIP model and then add the text label of each of the CIFAR-10 classes on all of the CIFAR-10 images and obtain the zero-shot predictions after corruption. We thus obtain the zero-shot performance of CLIP with text addition for each of the 10 classes.

The results as seen in Figure. 4 and 5 show that the model now predicts a significantly higher number of images for the corresponding added text label 2 and 5. Confusion matrix shows prediction for zero shot is considerably good. But, after corrupting CIFAR-10 dataset with labels of class 2 and class 5, more of the other images are now predicted as the corrupted classes which leads to the brighter columns 2 and 5 in Figure. 4. Similarly, Figure. 5 shows how the frequency increases towards a class by adding a text of a specific class to the images.

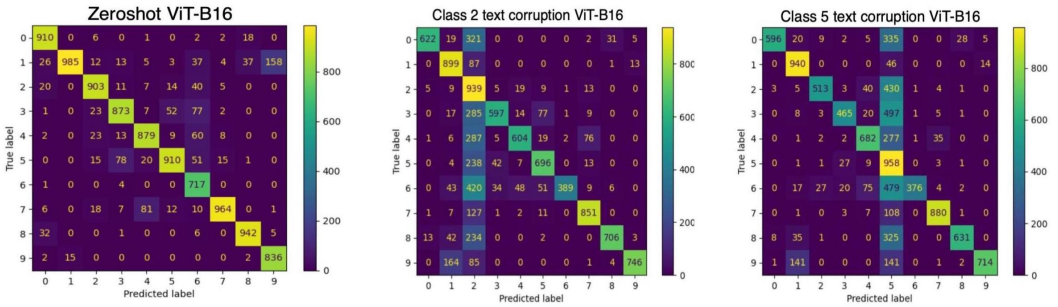

Figure 4: Confusion matrix comparison between zero shot predictions and corruption by adding class 2 and class 5 label as text. The confusion matrix gets biased towards the class when the specific class label is added as text.

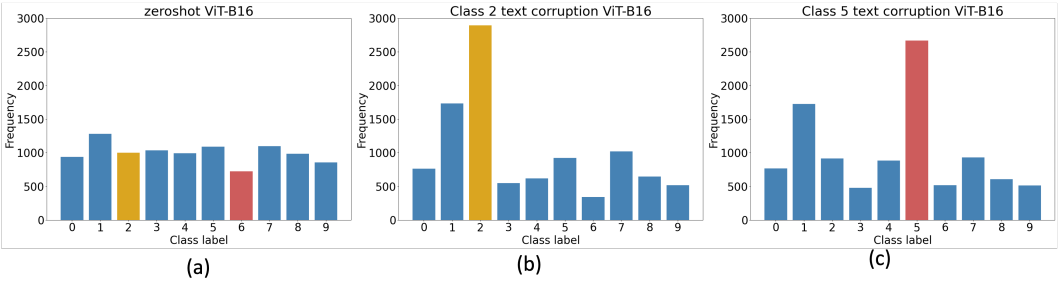

Figure 5: Comparison of prediction between (a) CLIP zero shot on original CIFAR-10, (b) CIFAR-10 images corrupted with class 2 text (shown by yellow bar), and (c) CIFAR-10 images corrupted with class 5 text (shown by red bar).

We extend the observation shown in CIFAR-10 to CIFAR-100. The results as seen in Figure. 6 shows that the model predicts a significantly higher number of images for the corresponding added text label 2 and 90. The bar plots shows how the frequency increases towards a class by adding a text of a specific class to the images.

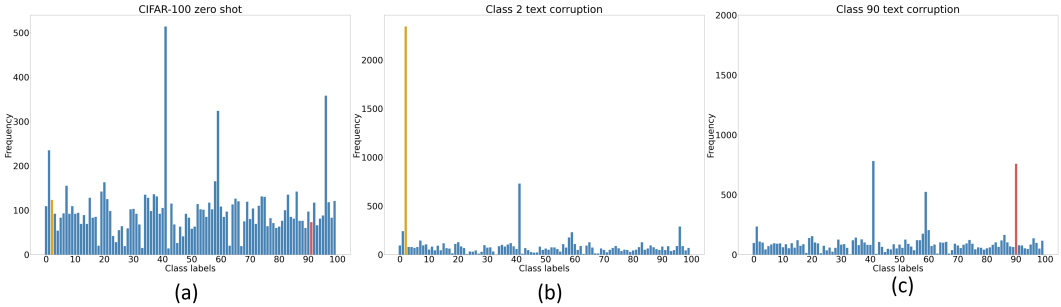

Figure 6: Comparison between the (a) CLIP zero shot on original CIFAR-100 , (b) corrupted CIFAR-100 images with class 2 text (shown in yellow) and (c) corrupted CIFAR-100 images with class 90 text.

5.2 Adversarial attack results

Comparison with the zero-shot: In Table. 5.2, we show the effectiveness of our LWAG approach by comparing the Top-1 and Top-5 accuracy with the zero-shot prediction on the original images. Our approach is able to make adversarial images such that the prediction with the CLIP-VIT-B/16 yields a top-1 accuracy of 10.61% from 89.16%, dropping by 78.55% on CIFAR-10 dataset. On CIFAR-100 and Caltech-101 the top-1 accuracy is dropped by 61.77% and 37.12%respectively. Further, we show the attack accuracy of LWAG approach, which produces adversarial images according to the target class that is randomly selected. Attack accuracy is the accuracy if the model predicts the label as the target adversarial label that is used in the targeted attack, i.e., the "target" is now the corruption text label. We obtain the top-1 attack accuracy of 97.81% on CIFAR-10 dataset with a single trained generator model.

Transferability to other datasets: We analyse the transferability of the generator model that is trained with CIFAR-10 dataset and tested on CIFAR-100 and Caltech-101 in Table. 5.2. To observe the attack accuracy, the target dataset classes are extended with the CIFAR-10 classes. The top-1 accuracy is reduced from 64.40% to 0.96% and the top-1 attack accuracy is 80.06% on the CIFAR-100. Although in the Caltech-101 the top-1 accuracy is getting reduced slightly by 11.29%, the top-1 attack accuracy is only 1.03% even with only 10 attack classes, possibly due to the distribution change.

Table 1: Performance evaluation of adversarial attack with CLIP zero-shot as baseline. The adversarial attack is better for a higher Attack accuracy and lower Top X accuracy.

Generator Encoder	Learning rate	Top 1 Acc.	Top 5 Acc.	Top 1 Attack Acc.	Top 5 Attack Acc.
CLIP Zero shot	CIFAR-10	89.16	99.08	-	-
	CIFAR-100	64.40	86.55	-	-
	Caltech-101	83.21	96.06	-	-
LWAG Attack (Ours)	CIFAR-10	10.61	52.2	82.67	97.81
	CIFAR-100	2.63	8.81	1.13	5.11
	Caltech-101	46.09	70.04	1.26	5.74

Table 2: Transferability of the model across datasets

Transfer	Top 1 Acc.	Top 5 Acc.	Top 1 Attack Acc.	Top 5 Attack Acc.
CIFAR-100 zero shot	64.40	86.55	-	-
Caltech-101 zero shot	83.21	96.06	-	-
CIFAR-10 to CIFAR-100	0.96	6.39	80.06	94.82
CIFAR-10 to Caltech-101	71.92	95.60	1.03	8.86

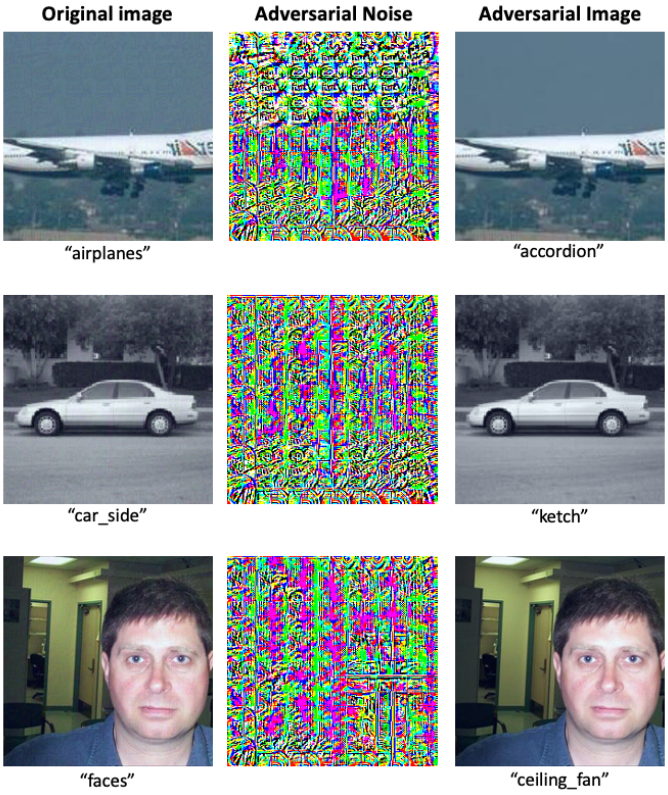

Figure 7: The Original image(left column), the generated structured noise(center column) and the adversarial image generated(right column). The true label and the predicted label are mentioned beneath each image.

5.3 Analysis and Ablations

Effect of encoder: Table. 5.3 describes the performance of different encoder choices for the generator model in the architecture design. The generator with native ResNet50 encoder is able to generate adversaries to decrease the model performance. However, the attack accuracy does not improve in this setting. Using the CLIP encoder with frozen weights in the generator increases the attack accuracy while dropping the model performance further. However, the best setting is obtained by finetuning the CLIP encoder in the generator which yields a Top-1 attack accuracy of 82.67% and drops the model Top-1 accuracy to 10.61%.

Visualization of generated noise: We show the generated noise and the adversarial image generated for each input image in Figure. 7. We observe the obtained adversarial image looks almost same as the original image yet it can fool the CLIP model.

Attention visualization: We also analyse the impact of the adversarial attacks in terms of the attention maps of the CLIP encoder. Figure. 8 shows the attention maps of the CLIP ViT-b/16 model on zero-shot performance, with text corruption (corrupted image) and on the adversarial image generated using LWAG attack (Ours). As it can be observed, the zero-shot attention map attends to

Table 3: Encoder choice of Generator model

Generator Encoder	Learning rate	Top 1 Acc.	Top 5 Acc.	Top 1 Attack Acc.	Top 5 Attack Acc.
Native ResNet50	$1e-3$	57.45	50.07	1.03	8.86
Frozen CLIP ResNet50	$1e-3$	13.70	52.78	15.94	64.05
Unfrozen ResNet50	$1e-5$	11.12	51.68	70.68	94.82
	$1e-3$	10.61	52.2	82.67	97.81

the object pretty well. However, when corrupted with text, the attention now shifts towards the text that has been added (second column). Meanwhile, the attention on the adversarially generated images can be observed to be scattered and lesser than the original zero-shot attentions.

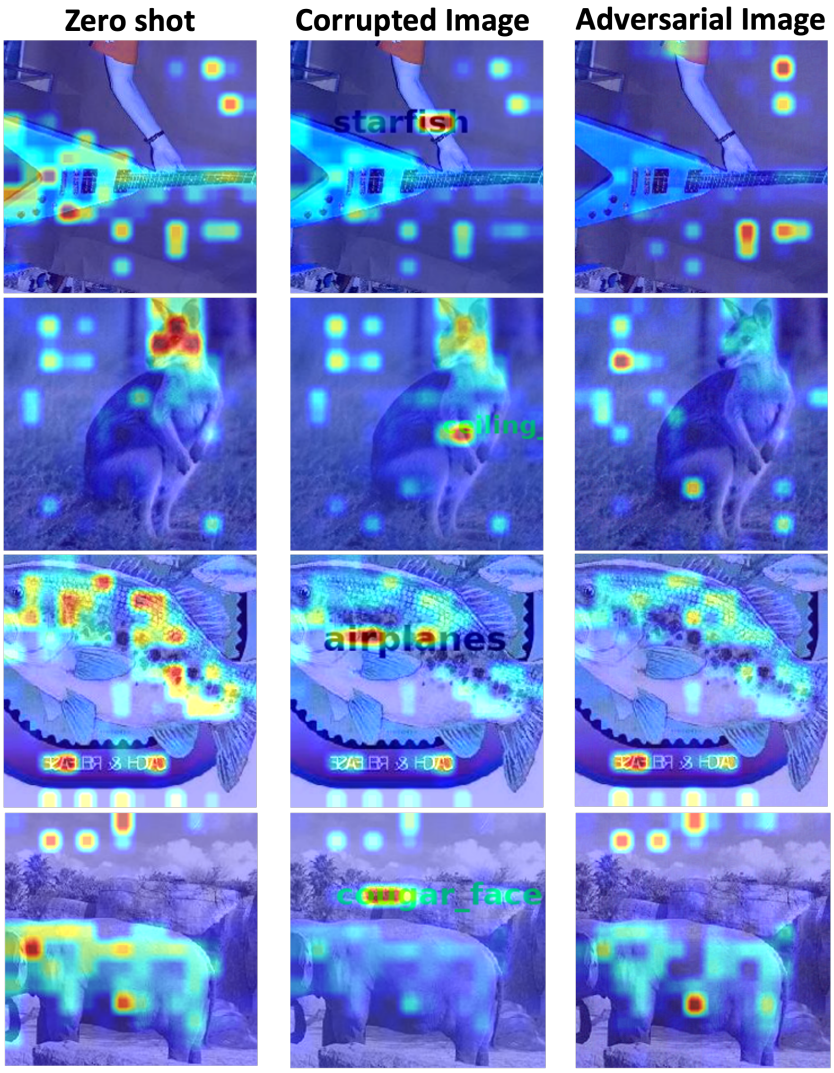

Figure 8: Attention comparison between the CLIP zero shot on the original image, corrupted image, and adversarial image. We get attention from the last self-attention layer of the ViT-B/16 architecture. The zero shot attention on the original image shows much stronger attention on the object of interest while the attention goes towards the text in the corrupted image. In the adversarial image, attention strength becomes less compared to the original image.

6 Conclusion

In this work we show the bias of CLIP - a vision-language model - towards text on image, and design a targeted adversarial attack using text itself for the CLIP model. We propose a generator model that can create adversarial samples using text as corruption, which are indistinguishable from the original image for humans, but capable of fooling the CLIP model. We demonstrate the effective use of language as an adversary for CLIP using extensive experiments with multiple datasets through the drop in model accuracy and increase in attack accuracy, especially on CIFAR-10 and CIFAR-100. As part of future works, we will explore the robustness of the adversarial generator across datasets, perhaps through longer training and stronger conditioning of the generator on the text corruption.

7 Acknowledgements

We are grateful for Hashmat Malik and Muzammal Naseer for their fruitful thoughts and suggestions. We would also like to thank Osama Afzal for his suggestion in code infrastructure.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, pp. 8748–8763, PMLR, 2021.

[2] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[3] J. Li, A. H. Miller, S. Chopra, M. Ranzato, and J. Weston, "Learning through dialogue interactions by asking questions," *arXiv preprint arXiv:1612.04936*, 2016.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[5] A. Aich, C. Khang-Ta, A. Gupta, C. Song, S. V. Krishnamurthy, M. S. Asif, and A. K. Roy-Chowdhury, "Gama: Generative adversarial multi-object scene attacks," *arXiv preprint arXiv:2209.09502*, 2022.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[7] A. Fawzi, O. Fawzi, and P. Frossard, "Analysis of classifiers' robustness to adversarial perturbations," *Machine learning*, vol. 107, no. 3, pp. 481–508, 2018.

[8] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "Robustness of classifiers: from adversarial to random noise," *Advances in neural information processing systems*, vol. 29, 2016.

[9] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.

[11] M. M. Naseer, S. H. Khan, M. H. Khan, F. Shahbaz Khan, and F. Porikli, "Cross-domain transferability of adversarial perturbations," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[12] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," *arXiv preprint arXiv:1801.02610*, 2018.

[13] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie, "Generative adversarial perturbations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4422–4431, 2018.

[14] K. R. Mopuri, P. K. Uppala, and R. V. Babu, "Ask, acquire, and attack: Data-free uap generation using class impressions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 19–34, 2018.

[15] "Roboflow, paint.wtf." `https://paint.wtf/`. Accessed: 2022-10-19.

[16] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[17] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Pattern Recognition Workshop*, 2004.