Web Artifact Attacks Disrupt Vision Language Models

Maan Qraitem, Piotr Teterwak, Kate Saenko, Bryan A. Plummer
Boston University
{mqraitem, piotrt, saenko, bplum}@bu.edu

Abstract

Vision-language models (VLMs) (e.g., CLIP, LLaVA) are trained on large-scale, lightly curated web datasets, leading them to learn unintended correlations between semantic concepts and unrelated visual signals. These associations degrade model accuracy by causing predictions to rely on incidental patterns rather than genuine visual understanding. Prior work has weaponized these correlations as an attack vector to manipulate model predictions, such as inserting a deceiving class text onto the image in a "typographic" attack. These attacks succeed due to VLMs' text-heavy bias—a result of captions that echo visible words rather than describing content. However, this attack has focused solely on text that matches the target class exactly, overlooking a broader range of correlations, including non-matching text and graphical symbols, which arise from the abundance of branding content in web-scale data. To address this gap, we introduce "artifact-based" attacks: a novel class of manipulations that mislead models using both non-matching text and graphical elements. Unlike typographic attacks, these artifacts are not predefined, making them simultaneously harder to defend against and more challenging to find. We address this by framing artifact attacks as a search problem and demonstrate their effectiveness across five datasets, with some artifacts reinforcing each other to reach 100% attack success rates. These attacks transfer across models with up to 90% effectiveness, making it possible to attack unseen models. To defend against these attacks, we extend prior work's artifact aware prompting to the graphical setting. We see a moderate reduction of success rates of up to 15% relative to standard prompts, suggesting a promising direction for enhancing model robustness. Code: https://github.com/mqraitem/Web-Artifact-Attacks

1. Introduction

Vision-language models (VLMs), such as CLIP [27] and LLaVA [19], are trained on large-scale, lightly curated web datasets [30, 32]. Since web captions often describe what is present rather than what is important, models learn spurious patterns—unintended correlations that arise from frequent but unrelated co-occurrences [1, 12, 14, 16, 18, 23, 33, 34, 37]. For example, prior work found that Chinese characters frequently co-occur with the word "carton" in ImageNet [16], leading the model to misclassify images containing Chinese characters as a "carton," regardless of its actual content. Moreover, these correlations can be weaponized by attackers to manipulate model predictions. Fig. 1(a) shows a well-documented example, typographic attacks [3, 26], where inserting the text of the wrong category into an image tricks the model into predicting that category. This attack takes advantage of the model's over-reliance on text, a bias reinforced by pretraining datasets in which captions frequently parrot text visible in the images [18].

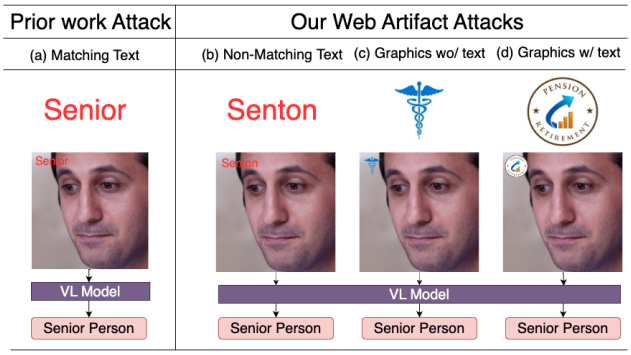

Figure 1. We propose a novel class of **Web Artifact Attacks** that expand the attack surface of Vision-Language Models beyond prior typographic attacks. For a sample task of predicting a person's age, (a) illustrates Typographic Attacks from prior work [3] which manipulate predictions using class-matching text. In contrast, our proposed web artifact attacks, consisting of (b) non-matching text, (c) standalone graphics, and (d) graphics with embedded text, can also mislead the model.

A shortcoming of prior work is that they focused only on explicit class-matching text, overlooking a broader set of vulnerabilities, including non-matching text and graphical artifacts such as logos. These elements, which are common in web-scale datasets due to branding and advertisements, may introduce unintended visual cues that models may rely on for classification. To address this gap, in Fig. 1, we introduce Web Artifact Attacks, a broader class of attacks that exploit

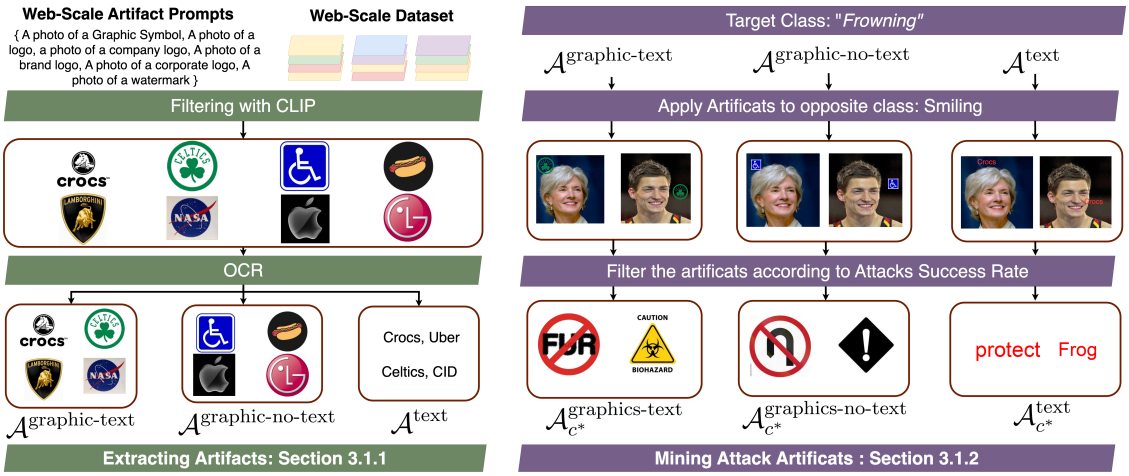

Figure 2. **Overview of the Web Artifacts Search and Selection**. The process involves two steps: Step 1: We filter an image-text dataset using CLIP-based retrieval and a set of artifact prompts. Retrieved artifacts are categorized into three groups: graphics with text ($\mathcal{A}^{\text{graphics-text}}$), graphics without text ($\mathcal{A}^{\text{graphics-no-text}}$), and unrelated text ($\mathcal{A}^{\text{text}}$) using OCR-based text extraction (Sec. 3.1.1). Step 2: artifacts are applied to opposing-class images (e.g., "Smiling" images when targeting "Frowning"), and their effectiveness is evaluated based on their ability to mislead model predictions. Artifacts are ranked according to their attack success rate, resulting in a refined set of high-impact artifacts for each category ($\mathcal{A}^{\text{graphics-text}}_{c^*}$, $\mathcal{A}^{\text{graphics-no-text}}_{c^*}$, and $\mathcal{A}^{\text{text}}_{c^*}$) (Sec. 3.1.2).

both non-matching text (Fig. 1(b)) and graphics (Fig. 1(c) and (d)) to mislead VLM predictions. We categorize graphical artifacts into two types: those with embedded text, which may reinforce textual biases, and those without text, isolating the impact of purely visual cues.

Unlike typographic attacks [3, 26], where the adversary inserts class-matching text to deliberately exploit the model's reliance on words, Web Artifact Attacks pose a greater threat because the defender does not know in advance the attacking artifacts. Thus, it is not obvious how to identify effective attacking artifacts or defend against them. To find attacks, we frame Web Artifact Attacks as a search problem, scanning image-text datasets [6, 29] to uncover text and graphics that can distort model predictions. The simplicity of these attacks makes them a notable security concern for deployed VLMs, as they only require dataset queries using similarity retrieval, making them highly accessible.

To that end, our search process consists of three stages: (1) Fig. 2 Green: Artifact Retrieval, where we extract candidate artifacts by scanning image-text datasets for images with text or graphics (2) Fig. 2 Purple: Effect Estimation, where we apply artifacts to images from opposing classes and evaluate their influence on model predictions, and (3) placement optimization, where we refine the positioning of artifacts to maximize attack success.

We evaluate the resulting attack artifacts across different VLM training objectives, visual encoder architectures, and pretraining dataset curation. Our findings demonstrate that Web Artifact Attacks are highly effective across various VLM training paradigms, including contrastive learning [27], sigmoid loss [36], and Large Language Model (LLM) fusion with visual features [19, 38]. Furthermore, these vulnerabilities persist regardless of the visual encoder size (ViT-B-32, ViT-B-16, ViT-L-14), and attempts at stronger pretraining dataset curation (DataComp [11]).

A key property of these attacks is high transferability—artifacts identified for one model retain up to 90% effectiveness when applied to another, indicating shared failure modes across vision language models. Additionally, attacks remain effective even when artifacts are small or transparent, revealing persistent vulnerabilities even in cases where artifacts are harder to detect. Furthermore, combining different artifact types significantly amplifies attack success, with text and graphical symbols reinforcing each other to reach nearly 100% misclassification rates. While text-based artifacts and graphics with embedded text are the most effective, graphics without text provide a stealthier attack vector, making them harder to detect while still influencing predictions.

Finally, we explore mitigation strategies by building on prior work [8], which showed that incorporating typographic attacks into the prompt can reduce their effectiveness. For example, if an image of a youth is attacked with the text "senior," the prompt "an image of a senior with the word 'youth' written on it" can weaken the attack. However, unlike typographic attacks, our attacks include graphical elements that cannot be easily incorporated into text. To address this, we propose augmenting the prompt with descriptions of graphical artifacts obtained through a captioning model. For LLM-based VLMs, we prompt the model to identify and describe these artifacts in its response. This approach reduces success rates by up to 15% relative to standard prompts, suggesting a promising direction for enhancing model robustness.

To summarize our contributions:

- We expose Web Artifact Attacks, which extend beyond typographic attacks by leveraging non-matching text, standalone graphics, and graphics with embedded text to mislead VLM predictions.
- We develop a scalable attack pipeline to search, evaluate, and optimize artifacts from pretraining datasets, leveraging their natural abundance in branding and advertisements.
- We show that our attacks are highly transferable, effective even when artifacts are small or transparent, and significantly stronger when combining multiple artifacts, reaching up to 100% success rates.
- We explore mitigation strategies that incorporate descriptions of the artifacts into the VLM(s) prompt which successfully mitigate some of their effect.

2. Related Work

Bias due to Artifacts. Prior work [16] has shown that models, from ResNet-50 [13] to large-scale VLMs [27, 36], can mistakenly associate Chinese characters watermark with the carton class in Imagenet [9]. Bykov et al. [5] further demonstrated that multilingual text artifacts (*e.g.*, Arabic, Latin, and Hindi) influence model predictions in unintended ways. Unlike these studies, which primarily observe such correlations, our work weaponizes these artifacts as an attack vector. Additionally, while prior work focused on manually discovered watermarks, we develop an automated mining pipeline that extracts a diverse set of non-matching text, graphical symbols with and without text at scale.

Typographic Attacks. Prior work [3, 8, 22, 26] has demonstrated that VLMs, including CLIP [27] and LLaVA [19], are highly susceptible to typographic attacks, where inserting class-matching text into an image directly manipulates predictions. These attacks exploit VLMs' over-reliance on textual cues, a bias reinforced by pretraining datasets in which captions often echo the visible text within images [18]. However, typographic attacks are inherently predefined, as they rely on explicitly matching class names, making them relatively easy to detect. In contrast, we show that non-matching text, graphical symbols, and mixed artifacts—elements not known a priori—can also mislead VLMs, enabling more flexible and covert attack strategies.

Relation to Adversarial Attacks. Adversarial attacks [7, 10, 28, 35, 39] learn an imperceptible noise, that disrupts the model visual recognition capabilities when added to the model. More recently, adversarial techniques have been weaponized for jailbreaking multimodal models to bypass safety mechanisms and generating restricted outputs [4, 24, 31]. While our attacks share a similar end goal to adversarial attacks, namely, disrupting model behavior, adversarial and artifact attacks are fundamentally different from each other. Our attacks originate as a result of spurious correlation in uncurated pretraining datasets, while adversarial attacks result from an artificial noise uniquely designed through the use of model intrinsics (*e.g.*, gradients) to disrupt the fundamental mechanics of neural nets. Therefore, to develop such attacks, an attacker needs a sophisticated understanding of the intrinsic mechanics of neural nets such as *e.g.*, Gradient [28, 35, 39], Meta Learning [10] or Attention maps [7] to name a few. However, the attacker, in our setup, only needs to know how to query the model.

3. Web Artifact Attacks Against Vision-Language Models

Vision-language models (VLMs) [19, 27, 36] are trained on large, uncurated web-scale datasets, leading them to form unintended correlations between visual elements and semantic categories [16, 18, 23, 33, 34]. Prior work has demonstrated that typographic attacks, which insert explicit class-matching text into images, exploit these correlations to manipulate model predictions [3, 26]. However, typographic attacks rely on direct textual matches, making them predictable and relatively easy to detect.

We propose Web Artifact Attacks, a new vector of unpredictable class of attacks that leverage artifacts from image-text datasets [6, 29]. Unlike typographic attacks, these artifacts—such as non-matching text (text that contains partial overlaps, phonetic similarities, or visually similar characters but does not explicitly spell out the target class), graphical symbols—do not need to explicitly match the target class yet can still mislead model predictions.

Formally, our artifact-based attack introduces an artifact a correlated with a target class c^* into an image x, causing the model f_θ to misclassify it as c^*:

$$\arg\max_c f_\theta(x) = \hat{c} \quad \Rightarrow \quad \arg\max_c f_\theta(x \oplus a) = c^*,$$

where $\oplus$ denotes artifact insertion and $\hat{c}$ is the correct class.

Since these artifacts are not known a priori, they are harder to defend against while also making it challenging to determine in advance which artifacts will be effective. Therefore, we frame finding our attack artifacts as a search problem: systematically identifying artifacts within large-scale image-text datasets that can trigger misclassification.

To that end, Sec. 3.1 describes our proposed the search pipeline, Sec. 3.2 analyzes the extracted artifacts to reveal broader insights into model vulnerabilities, Sec. 3.3 introduces an evaluation metric that accounts for occlusion effects, and finally, Sec. 5 proposes a mitigation strategy.

3.1. Attack Pipeline

This section describes our search pipeline to systematically find attack artifacts. First, in Sec. 3.1.1, we extract artifacts from pretraining datasets by filtering images containing either text or graphical elements, categorizing them into non-matching text, graphics with text, and graphics without text.

Next, in Sec. 3.1.2, we evaluate how these artifacts influence model predictions by applying them to opposing-class images and selecting those that consistently increase misclassification rates. Finally, in Sec. 3.1.3, we optimize artifact placement to maximize attack effectiveness.

3.1.1. Extracting Artifacts from a Pretraining Dataset

To search for our artifacts, we leverage the fact that graphics (*e.g.* a logo of a company) frequently appear in web-scale datasets as standalone images due to the abundance of branding content. Using a CLIP-based retrieval mechanism, we apply broad text prompts $\mathcal{P}$ (*e.g.*, "a photo of a graphic") to extract images from an image-text dataset $\mathcal{D}$ (*e.g.*, CC12M [6]). We then rank the retrieved images by their similarity scores and retain the top 1% of the highest-scoring images, ensuring that the selected artifacts closely match the intended query while filtering out irrelevant content. To verify the quality of the retrieved artifacts, we conduct a human study by randomly sampling 1,000 images from the extracted set and manually assessing their noise level, defined as the percentage of non-graphic or irrelevant images. We find that the noise level remains below 2%, indicating that our retrieval process effectively isolates graphical artifacts while minimizing unrelated content.

Next, as shown in Fig. 2, we categorize artifacts into three types Green : unrelated text, graphics with embedded text, and graphics without text. To achieve this categorization, we apply OCR-based text detection (We use off-the-shelf models from EasyOCR [2]) to identify artifacts containing text and separate them from purely graphical artifacts. Additionally, to ensure that the extracted text does not merely replicate typographic attacks from prior work [3, 26], we filter out any text that explicitly matches class names in the downstream dataset. This structured approach allows us to systematically evaluate how VLMs respond to textual and graphical cues, both independently and in combination, providing deeper insight into their learned vulnerabilities.

3.1.2. Finding Artifacts That Influence a Target Class

After obtaining our set of artifacts in Sec. 3.1.1, the next step is to determine which ones are most effective at misleading model predictions. Not all artifacts significantly influence classification—some may be ignored by the model, while others may reinforce correct predictions. To systematically identify high-impact artifacts, we evaluate their ability to shift model predictions toward a target class.

To that end, as Fig. 2 Purple illustrates, we apply each artifact to opposing-class images (*e.g.* if the target class is "car images" then we use every image that is not a car) and measure its effectiveness in shifting predictions toward the target class If an artifact consistently causes images from unrelated classes to be classified as the target class, it reveals a strong learned correlation in the model that can be exploited for adversarial purposes [25]. Formally, given a downstream dataset $\mathcal{D}_{\text{downstream}}$, we define a subset of opposing-class images as:

$$\mathcal{D}^{\text{train}}_{\neg c^*} \subset \{x \mid x \in \mathcal{D}_{\text{downstream}}, y \neq c^*\},$$

where y is the ground-truth label. As we show in our experiments, $\mathcal{D}^{\text{train}}_{\neg c^*}$ can be as small as 10 samples per class while still uncovering the most effective artifacts. We then apply each artifact a from the three previously defined categories to this subset.

To ensure efficient evaluation, artifacts are initially placed at random locations. This allows us to identify artifacts that demonstrate adversarial effects across a variety of placements, rather than optimizing for a single position prematurely. By deferring location-specific optimization, we reduce the computational complexity from $\mathcal{O}(|\mathcal{A}||\mathcal{D}^{\text{train}}_{\neg c^*}||\mathcal{L}|)$ to $\mathcal{O}(|\mathcal{A}||\mathcal{D}^{\text{train}}_{\neg c^*}|)$ during this stage (where $\mathcal{L}$ refers to the set of target locations), ensuring that we focus on selecting artifacts with strong generalizable effects before fine-tuning their positioning.

Artifact impact score a is defined per-artifact as the proportion of modified images classified as the target class:

$$P_{c^*}(a) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[f_\theta(x_i^a) = c^*],$$

where higher values indicate a stronger influence on misclassification. We compute this score for each artifact in our three categories: text artifacts ($\mathcal{A}^{\text{text}}$), graphics with embedded text ($\mathcal{A}^{\text{graphics-text}}$), and graphics without text ($\mathcal{A}^{\text{graphics-no-text}}$), as defined in Sec. 3.1.1. For each artifact category, we rank artifacts by $P_{c^*}(a)$ in descending order and select the top s most effective ones. This results in the sets $\mathcal{A}^{\text{text}}_{c^*}$, $\mathcal{A}^{\text{graphics-text}}_{c^*}$, and $\mathcal{A}^{\text{graphics-no-text}}_{c^*}$, where c^* denotes the target class.

3.1.3. Optimizing Artifact Placement

While Sec. 3.1.2 identified artifacts that are consistently effective across different placements, their positioning within an image can further enhance misclassification rates.

To address this, for each artifact $a \in \mathcal{A}^{\text{top-text}}_{c^*} \cup \mathcal{A}^{\text{top-graphics-text}}_{c^*} \cup \mathcal{A}^{\text{top-graphics-no-text}}_{c^*}$, we evaluate a set of candidate locations $\mathcal{L}$ to determine the most effective positioning. For each artifact, we pick the location that obtains the highest success rate. This step fine-tunes the effect of already strong artifacts, maximizing misclassification rates. Refer to the Supp. B for an ablation of location effect. In short, we find that the top border of the image (especially top middle) to be the most effective.

3.2. Insights from the Artifacts

After extracting and ranking artifacts based on their influence on model predictions, we examine the highest-impact artifacts for different target classes. Fig. 3 presents examples of these artifacts, categorized into **graphics without text** ($\mathcal{A}^{\text{graphics-no-text}}_{c^*}$), **graphics with embedded text**

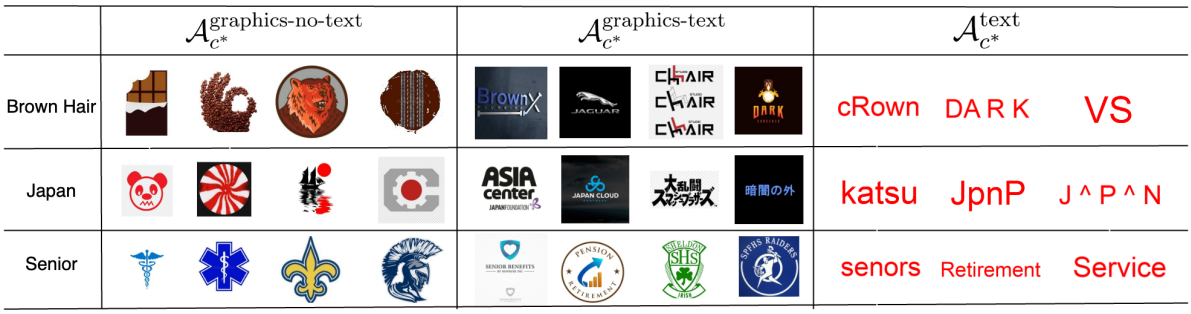

Figure 3. **Examples of Web Artifacts** split into: graphics without text ($A_{c^*}^{\text{graphics-no-text}}$), graphics with embedded text ($A_{c^*}^{\text{graphics-text}}$), and unrelated text ($A_{c^*}^{\text{text}}$). Each row corresponds to a different target class (*Brown Hair*, *Japan*, *Adult*), that models have learned to associate with the artifacts. Notably, text artifacts need not match the class exactly, while graphical symbols can represent indirect but learned associations. These findings highlight the diverse range of artifacts that can manipulate model predictions. Refer to Supp. A for more examples.

($\mathcal{A}_{c^*}^{\text{graphics-text}}$), and **unrelated text artifacts** ($\mathcal{A}_{c^*}^{\text{text}}$). These examples reveal distinct patterns in how different artifact types induce misclassification.

Non-textual graphics often share visual attributes loosely related to the target class. For instance, brown-colored logos and animal symbols appear frequently for "Brown Hair," while "Japan" is associated with red-and-white designs reminiscent of its national flag. Similarly, the "Senior" class is linked to medical symbols and retirement-related imagery, suggesting that models rely on broad visual associations learned from pretraining data.

Graphics with embedded text frequently contain contextually related words, even when the complete phrase is irrelevant. Examples include "Brown" from unrelated brand names appearing in the "Brown Hair" category and "Japan Cloud" for the "Japan" category. This demonstrates that models are highly sensitive to the presence of text fragments within graphical elements, further reinforcing the text-heavy biases introduced during pretraining.

Text-only artifacts, we observe frequent misspellings, phonetic approximations, and visually similar character patterns rather than exact class matches. Words like "senors" for "Senior" or "J^P^N" for "Japan" illustrate how models generalize text-based associations beyond strict class labels, making these artifacts more difficult to detect than conventional typographic attacks.

Refer to Supplementary A for more examples.

3.3. Attack Success Evaluation

Prior evaluations of typographic attacks [3, 8, 22, 26] measure attack success by checking whether inserting class-matching text misleads VLMs. However, they overlook occlusion effects—misclassification may occur due to the added text obscuring important image features. To ensure that attack success is solely due to the artifact rather than occlusion, we introduce an evaluation framework that isolates its effect while preserving the image's classifiability.

Given an input image x and an artifact a placed at location l, we define the attack-modified image as $x^a = x \oplus a$, where $\oplus$ denotes artifact insertion. To verify that misclassification stems from the artifact itself, we introduce a masking-based validation step. We generate a masked version of the image, $x^{\neg l}$, where the region l is occluded using a mask M_l.

To ensure that the image remains classifiable after artifact removal, we retain only samples where $x^{\neg l}$ is still correctly classified as its original class c. Let f_θ be the model's classifier, and let c be the ground-truth class of x. We discard any image where $f_\theta(x^{\neg l}) \neq c$, ensuring that attack success cannot be attributed to the removal of critical visual features.

The Attack Success Rate (ASR) is then defined as the proportion of cases where the model predicts the target class c^* for x^a but maintains correct classification for $x^{\neg l}$:

$$ASR = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[f_\theta(x_i^a) = c^* \wedge f_\theta(x_i^{\neg l}) = c\right],$$

where $\mathbf{1}[\cdot]$ is the indicator function, and N is the number of samples passing the masking validation step. This stricter evaluation ensures that attack success is genuinely due to the artifact rather than incidental occlusion effects. To ensure that the attack generalizes beyond the images used during search, the final ASR is evaluated on a separate test set disjoint from the training set $\mathcal{D}_{\neg c^*}^{\text{train}}$ used for artifact mining.

4. Experiments

Datasets. We use CC12M [6] as our primary dataset for searching and extracting artifacts. Our extraction process yields 87k artifacts, but this approach is dataset-agnostic and can be applied to any large-scale image-text dataset, such as LAION [29, 30]. To evaluate the effectiveness of these artifacts, we conduct experiments across five distinct tasks spanning both human-related attributes and object/geography classification. For human-related tasks, we predict Gender (Male/Female) and Age (Child, Teen, Adult, Elderly) using FairFace [15], as well as Smiling using CelebA [20]. For non-human classification tasks, we predict aircraft models using FGVC Aircraft [21] and countries using Country211 [27]. For each downstream dataset, we use only 32 images

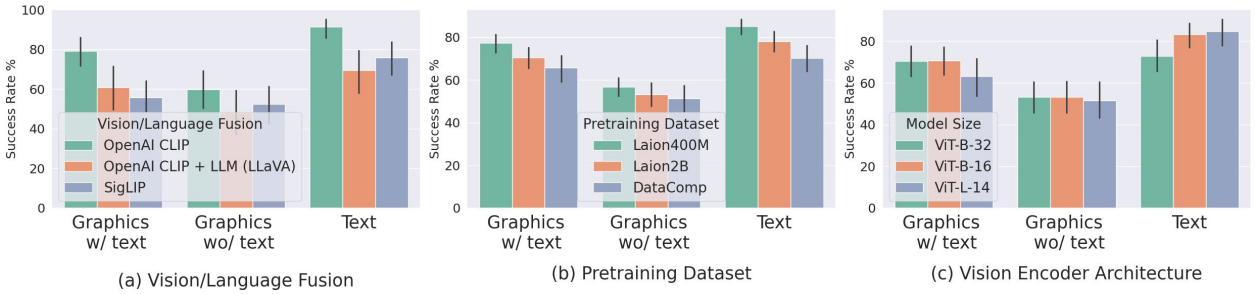

Figure 4. **Breakdown of Artifact Attacks Performance.** Our Web Artifact attacks are highly effective across different configurations: namely (a) Vision/Language Fusion, (b) Pretraining Dataset, and (c) Vision Encoder Architecture. Refer to Sec. 4.1 for further discussion..

per class to estimate the Artifact Success Rate (ASR), which as we demonstrate in Sec. 4.1, is sufficient to estimate ASR.

Metrics. We use the success rate as outlined in Sec. 3.3. In short, given a target class c^*, we measure attack effectiveness by evaluating whether the model misclassifies an image after the artifact that correlates with c^* is introduced while ensuring that the image remains correctly classified when the artifact is masked. We also report the change of the confidence score in target class c^* after pasting the artifact.

Models. We evaluate the robustness of VLMs to Web Artifact Attacks by varying vision-language fusion strategy, vision encoder architecture, and pretraining dataset size and curation. For fusion, we test contrastive learning (CLIP) [27], sigmoid-based alignment (SigLIP) [36], and LLM-based fusion (LLaVA) [19]. To assess the impact of architecture size, we evaluate ViT-B-32, ViT-B-16, ViT-L-14 while we fix the pretraining dataset to LAION 2B [29]. Finally, we compare models pretrained on LAION 400M, LAION-2B [29], and DataComp [11] while fixing the pretraining architecture to ViT-B-32 and ViT-B-16 to examine whether dataset curation affects attack vulnerability.

Artifact Setting. We fix the artifact size to 10th of the image size. Refer to Supplementary. D for a visual example and ablation of artifact size and transparency.

4.1. Results

We benchmark the effect of Web Artifact Attacks on Vision-Language Models (VLMs) across different vision-language fusion methods, model architectures, and pretraining datasets. We make the following observations:

Artifact Attacks are Effective across Diverse Model Configurations. As seen in Fig. 4, Web Artifact Attacks consistently achieve high success rates across different vision-language fusion strategies, pretraining datasets, and model architectures, reaching up to 90% success. Comparing the artifacts, text-based attacks are the most effective, followed by graphics with embedded text, and finally graphics without text, suggesting that models are most vulnerable to explicit textual cues but still susceptible to purely visual artifacts.

CLIP is most vulnerable to Artifact Attacks. In Fig. 4(a), models trained with standard contrastive learning (OpenAI CLIP) exhibit the highest attack success rates across all artifact types, particularly for text-based artifacts, reinforcing CLIP's strong reliance on textual cues. The LLM-based fusion model (LLaVA) shows a moderate reduction in vulnerability, likely due to its alignment on a smaller, curated dataset such as COCO [17], which likely have much fewer graphics and logos. Similarly, SigLIP, which employs a different training objective and a pretraining dataset, achieves moderate reduction in vulnerability too on par with LLaVA.

Pretraining Dataset Curation has Minimal Effect on Success Rate. Fig. 4(b) shows that models pretrained DataComp [11] exhibit a slightly improved robusteness to artifact attacks than uncurated Laion400M and LAION 2B. Despite DataComp's filtering, it does not majorly reduce susceptibility, suggesting that its filtering strategy may be ill-suited for mitigating spurious correlations. DataComp primarily optimizes for image quality and semantic similarity to high-quality benchmarks, rather than explicitly removing textual artifacts or graphical elements that reinforce spurious correlations. As a result, while the dataset curation improves image-text alignment for downstream tasks, it fails to eliminate the kind of incidental patterns that make models vulnerable to Web Artifact Attacks. This highlights a critical limitation: filtering strategies that focus on dataset relevance may not address Web Artifact Attacks.

Larger Architecture is More Vulnerable to Textual Attacks. As seen in Fig. 4(c), larger models, such as ViT-L-14, exhibit higher attack success rates on text-based attacks. This can be attributed to their increased capacity for optical character recognition, enabling them to more effectively detect and leverage textual elements in images. As a result, they become more susceptible to attacks that exploit text-based spurious correlations present in web-scale datasets.

Artifacts Based Attacks vs Prior Work's Typographic Attacks. Fig. 5 presents the average performance of our web-scale Web Artifact Attacks compared to Typographic Attacks [3]. The results demonstrate that our text-based attacks, derived from mined logo datasets, are more effective than typographic attacks, which are typically limited to using text from the opposing class. This suggests that optimizing attacks beyond simple class-matching text—by leveraging

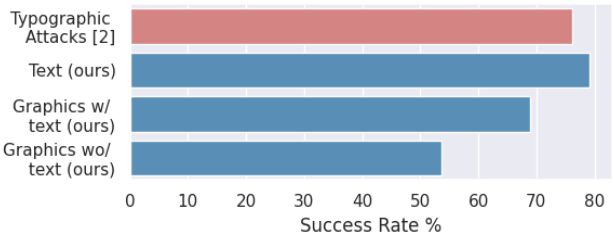

Figure 5. **Comparison to Prior Work Attacks**. Success rate of Web Artifact attacks vs prior work Typographic Attacks [3].

Figure 6. **Confidence in Target Class** when different types of artifacts—graphics with text (orange), graphics without text (green), and text (red)—are introduced, compared to the baseline with no artifacts (blue). Results demonstrate that artifacts significantly increase model confidence in the target class.

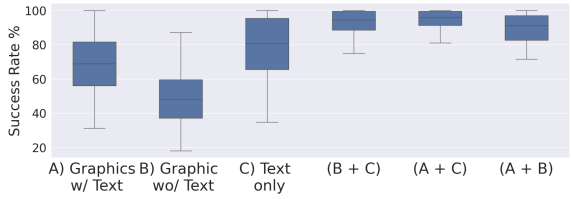

Figure 7. **Success Rate of Combining artifacts**. Results indicate that combining artifact types leads to higher attack success.

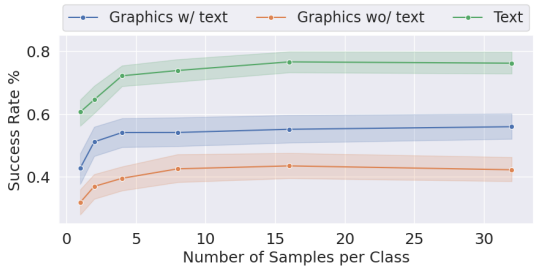

Figure 8. **Sample Size vs Success Rate**. Success rate of artifact-based attacks vs the number of samples used from the downstream dataset to estimate their effect. Results indicate that a relatively small sample size is sufficient to assess attack effectiveness.

a broader range of textual configurations—can significantly enhance their effectiveness. In contrast, graphical attacks tend to be less effective overall. However, they open up a novel attack surface that, as discussed in Section 5, proves more challenging to defend against.

Artifacts Increase Model Confidence in the Target Class. Fig. 6 shows that the introduction of textual and graphical artifacts significantly increases the model's confidence in the target class compared to the no-artifact baseline. Text-based artifacts lead to the highest confidence scores, followed by graphics with embedded text and graphics without text. This increased confidence correlates with the attack success rates observed in prior experiments, suggesting that artifacts not only influence predictions but also enhance model certainty in incorrect classifications.

The Compounding Effect of Different Types of Artifacts Fig. 7 presents the attack success rate across different artifact types and their combinations. The results highlight a clear synergistic effect when multiple artifacts are used together, significantly increasing misclassification rates compared to individual artifacts. While text-only artifacts remain highly effective, their impact is amplified when combined with graphical symbols. The highest success rates are observed in (A + C) and (A + B) cases, where text is paired with either graphical symbols with or without text, reaching near 100% attack success in some instances. This suggests that models do not treat textual and graphical cues independently but rather reinforce their reliance on both when they co-occur, exacerbating spurious correlations learned during pretraining. Additionally, graphical artifacts without text (B) alone exhibit moderate attack success, but when combined with text (B + C), their effectiveness increases substantially.

A Small Sample Size is Enough for Estimating Artifact Effect. A key question for our attacks is how many samples are needed per class to reliably estimate an artifact's attack effect. Fig. 8 shows that across all artifact types, the attack success rate stabilizes quickly as the number of samples increases, indicating that only a small sample size is required to obtain a reliable estimate. This result is significant as it suggests that evaluating Web Artifact Attacks does not require extensive sampling, which greatly reduces computational overhead. By using only a fraction of the dataset, we can efficiently assess the impact of artifacts while preserving computational resources. This finding is particularly important for large-scale evaluations, where repeatedly testing artifacts on full datasets would be prohibitively expensive.

Web Artifact Attacks are Highly Transferable. Fig. 9 illustrates that Web Artifact Attacks are highly transferable across different models. Across all artifact types, success rates remain consistently high when applying artifacts mined from one model to another. The most significant drop in success rate occurs when transferring attacks to a model trained with SigLIP [36], which employs a different training objective. However, even in this case, the attack success rates remain substantial, indicating that spurious correlations exploited by these attacks persist across diverse model architectures and training paradigms. This suggests that mitigating these vulnerabilities will require more than simply altering the training objective.

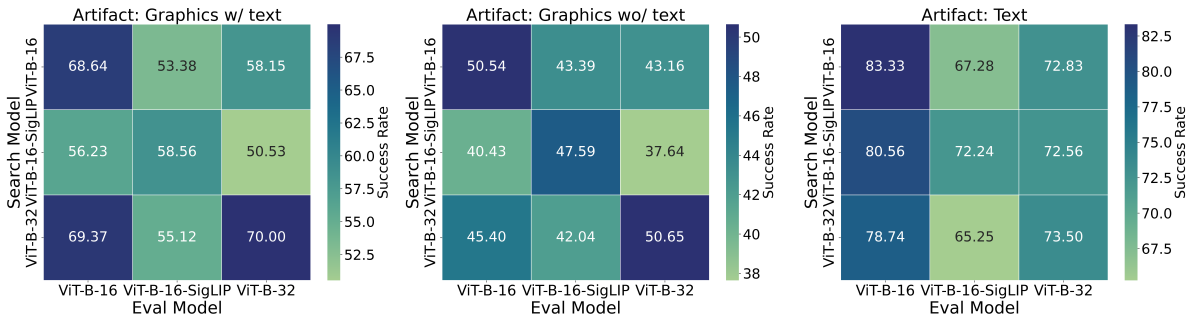

Figure 9. **Transferability of Web Artifact Attacks Across Different Vision Encoder Architectures.** The heatmap shows the success rate (%) of artifacts search on a specific model (y-axis) and evaluated on another model (x-axis). Higher success rates indicate greater transferability of spurious correlations. Results indicate that artifacts effect is highly transferable across architectures.

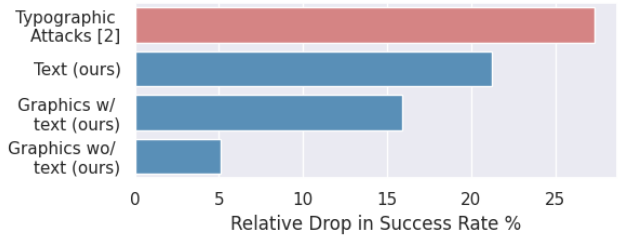

Figure 10. **Artifact Aware Prompting**. Results indicate that the attack effect can be mitigated with more informative prompts.

5. Mitigation w/ Artifact Aware Prompting

Recent work [8] has uncovered an important insight: VLMs can be guided to adjust their attention toward specific visual elements when provided with more informative prompts. This finding was leveraged as a mitigation strategy for typographic attacks by explicitly incorporating the attack text into the prompt. For example, assume the an image of a dog is attacked with the text of "a cat," then rather than using "a photo of a dog," the authors use: "a photo of a dog with cat written on it." While this approach is effective for textual artifacts, it does not generalize to graphical artifacts, which lack an explicit textual form and cannot be directly incorporated into the language input.

To address this limitation, we propose a mitigation approach inspired by [8] for the two types of models. For **Contrastive VLM(s)**: we use structured descriptions of graphical symbols to make these artifacts explicit to the model. Specifically, we employ a captioning model to transform the graphics into a textual description which then are appended to the input prompt, ensuring that the model processes them explicitly. Refer to Supp. E for an example. For **Large VLM(s)**: following [8], we leverage the generative capabilities for LVLMs like LLaVA [19] to integrate artifact awareness directly. We prompt LLaVA to describe any graphical or textual artifacts present in the image, ensuring that these features are explicitly considered during inference. Note that our approach does not require model retraining or architectural modifications. The only overhead is the cost of executing the captioning model.

Results. Fig. 10 illustrates the effectiveness of our approach in reducing attack success rates. Across all artifact types, artifact-aware prompting consistently lowers the success rate, indicating that explicitly guiding the model to consider artifacts can diminish their impact. However, this defense is less effective against our attacks compared to prior typographic attacks [3], with the largest performance gap observed for graphical attacks. This highlights the increased difficulty posed by our attacks and suggests promising directions for future research in developing more robust defenses.

6. Conclusion

In this work, we introduced **Web Artifact Attacks**, a novel vector of attacks that make use of non-matching text and graphical artifacts—both with and without embedded text—to mislead model predictions, revealing vulnerabilities beyond traditional typographic attacks. Our results show that these attacks not only increase model confidence in incorrect classifications but also persist across different model architectures, vision-language fusion strategies, and pretraining datasets. Moreover, we showed how our search process for attack artifacts is efficient only requiring a handful of samples to approximate attack effect. While mitigation strategies such as artifact-aware prompting reduce attack success rates, they are insufficient to eliminate the threat, particularly for non-textual graphical artifacts.

Future Work. Our findings highlight the need for more robust defenses against our attacks. A promising direction is dataset curation that extends beyond simple CLIP-based filtering; developing more informative captioning strategies that better contextualize images containing logos and graphics could help models learn more meaningful associations rather than relying on spurious correlations. Additionally, while our artifact discovery process is effective, it could be further optimized for efficiency. Future improvements could explore faster, more scalable methods for identifying artifacts in large-scale datasets, reducing the computational cost of attack evaluations without sacrificing accuracy.

Acknowledgments This material is based upon work supported, in part, by DARPA under agreement number HR00112020054. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the supporting agencies.

References

[1] Sandhini Agarwal, Gretchen Krueger, Jack Clark, Alec Radford, Jong Wook Kim, and Miles Brundage. Evaluating clip: towards characterization of broader capabilities and downstream implications. *arXiv preprint arXiv:2108.02818*, 2021. 1

[2] Jaided AI. EasyOCR: Ready-to-use ocr with 80+ supported languages and growing. https://github.com/JaidedAI/EasyOCR, 2020. 4

[3] Hiroki Azuma and Yusuke Matsui. Defense-prefix for preventing typographic attacks on clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3644–3653, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[4] Eugene Bagdasaryan, Tsung-Yin Hsieh, Ben Nassi, and Vitaly Shmatikov. Abusing images and sounds for indirect instruction injection in multi-modal llms. *arXiv preprint arXiv:2307.10490*, 2023. 3

[5] Kirill Bykov, Klaus-Robert Müller, and Marina M-C Höhne. Mark my words: Dangers of watermarked images in imagenet. In *European Conference on Artificial Intelligence*, pages 426–434. Springer, 2023. 3

[6] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3558–3568, 2021. 2, 3, 4, 5

[7] Sizhe Chen, Zhengbao He, Chengjin Sun, Jie Yang, and Xiaolin Huang. Universal adversarial attack on attention and the resulting dataset damagenet. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2188–2197, 2020. 3

[8] Hao Cheng, Erjia Xiao, Jindong Gu, Le Yang, Jinhao Duan, Jize Zhang, Jiahang Cao, Kaidi Xu, and Renjing Xu. Unveiling typographic deceptions: Insights of the typographic vulnerability in large vision-language model. *ECCV*, 2024. 2, 3, 5, 8

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[10] Jiawei Du, Hu Zhang, Joey Tianyi Zhou, Yi Yang, and Jiashi Feng. Query-efficient meta attack to deep neural networks. *arXiv preprint arXiv:1906.02398*, 2019. 3

[11] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36: 27092–27112, 2023. 2, 6

[12] Melissa Hall, Laura Gustafson, Aaron Adcock, Ishan Misra, and Candace Ross. Vision-language models performing zero-shot tasks exhibit gender-based disparities. *arXiv preprint arXiv:2301.11100*, 2023. 1

[13] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[14] Sepehr Janghorbani and Gerard De Melo. Multimodal bias: Introducing a framework for stereotypical bias assessment beyond gender and race in vision language models. *arXiv preprint arXiv:2303.12734*, 2023. 1

[15] Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1548–1558, 2021. 5

[16] Zhiheng Li, Ivan Evtimov, Albert Gordo, Caner Hazirbas, Tal Hassner, Cristian Canton Ferrer, Chenliang Xu, and Mark Ibrahim. A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20071–20082, 2023. 1, 3

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014. 6

[18] Yiqi Lin, Conghui He, Alex Jinpeng Wang, Bin Wang, Weijia Li, and Mike Zheng Shou. Parrot captions teach clip to spot text. In *European Conference on Computer Vision*, pages 368–385. Springer, 2024. 1, 3

[19] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 3, 6, 8

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015. 5

[21] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5

[22] Joanna Materzyńska, Antonio Torralba, and David Bau. Disentangling visual and written concepts in clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16410–16419, 2022. 3, 5

[23] Yannic Neuhaus, Maximilian Augustin, Valentyn Boreiko, and Matthias Hein. Spurious features everywhere - large-scale detection of harmful spurious features in imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20235–20246, 2023. 1, 3

[24] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI conference on artificial intelligence*, pages 21527–21536, 2024. 3

[25] Maan Qraitem, Kate Saenko, and Bryan A Plummer. Bias mimicking: A simple sampling approach for bias mitigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20311–20320, 2023. 4

[26] Maan Qraitem, Nazia Tasnim, Piotr Teterwak, Kate Saenko, and Bryan A Plummer. Vision-llms can fool themselves with self-generated typographic attacks. *arXiv preprint arXiv:2402.00626*, 2024. 1, 2, 3, 4, 5

[27] 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. 1, 2, 3, 5, 6

[28] Jérôme Rony, Eric Granger, Marco Pedersoli, and Ismail Ben Ayed. Augmented lagrangian adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7738–7747, 2021. 3

[29] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 2, 3, 5, 6

[30] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 1, 5

[31] Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. Jailbreak in pieces: Compositional adversarial attacks on multimodal language models. *arXiv preprint arXiv:2307.14539*, 2023. 3

[32] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 1

[33] Maya Varma, Jean-Benoit Delbrouck, Zhihong Chen, Akshay Chaudhari, and Curtis Langlotz. Ravl: Discovering and mitigating spurious correlations in fine-tuned vision-language models. *arXiv preprint arXiv:2411.04097*, 2024. 1, 3

[34] Qizhou Wang, Yong Lin, Yongqiang Chen, Ludwig Schmidt, Bo Han, and Tong Zhang. A sober look at the robustness of clips to spurious features. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1, 3

[35] Zheng Yuan, Jie Zhang, Yunpei Jia, Chuanqi Tan, Tao Xue, and Shiguang Shan. Meta gradient adversarial attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7748–7757, 2021. 3

[36] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 2, 3, 6, 7

[37] Kankan Zhou, Yibin LAI, and Jing Jiang. Vlstereoset: A study of stereotypical bias in pre-trained vision-language models. In *Association for Computational Linguistics*, 2022. 1

[38] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 2

[39] Mingkang Zhu, Tianlong Chen, and Zhangyang Wang. Sparse and imperceptible adversarial attack via a homotopy algorithm. In *International Conference on Machine Learning*, pages 12868–12877. PMLR, 2021. 3