

One Object, Multiple Lies: A Benchmark for Cross-task Adversarial Attack on Unified Vision-Language Models

Supplementary Material

7. Dataset Construction Details

7.1. Comparison with Other Adversarial Attack Benchmarks

Table 4 compares CrossVLAD with existing adversarial attack benchmarks for vision-language models. While some prior works have considered unified VLMs or multiple tasks, CrossVLAD uniquely combines comprehensive multi-task coverage with explicit cross-task evaluation metrics. Additionally, our benchmark contains a substantial dataset size of 3,000 carefully curated samples, enabling statistically robust evaluation.

Table 4. Comparison with existing adversarial attack benchmarks for vision-language models.

Benchmark	Unified VLMs	Multi-tasks	Cross-task Evaluation	Attack Type	Dataset Size
CrossVLAD (Ours)	✓	✓	✓	target	3k
vllm-safety-bench[31]	✓	✗	✗	untarget	2k
ROZ-benchmark[32]	✗	✗	✗	untarget	✗
MultiTrust [42]	✓	✗	✗	both	200

7.2. Selection Criteria

The CrossVLAD benchmark was constructed using following filtering approach:

- Excluded images containing potential target categories to avoid pre-existing confusion
- Verified source categories existed in the image
- Limited the maximum number of object instances per image to 5
- Ensured category uniqueness within each image to avoid ambiguity
- Object size constraints: Selected objects occupying between 10% and 50% of the image area
- Caption verification: Required objects to appear in at least 3 of the 5 MSCOCO captions

7.3. Annotation Process

For each selected image, we:

- Preserved original MSCOCO annotations (bounding boxes, category labels)
- Randomly selected one qualified source object per image
- Identified appropriate target category from our predefined change-pairs
- Used GPT-4 to generate target captions with the following prompt:

You are given a picture with a primary object called "[SOURCE_CATEGORY]".

Below are 5 captions that describe the image including this object:

1. [CAPTION_1]
2. [CAPTION_2]
- ...
5. [CAPTION_5]

Task: Imagine replacing the primary object "[SOURCE_CATEGORY]" with a new object "[TARGET_CATEGORY]". Create a caption describing the scene with this replacement.

We implemented quality control by verifying that each generated caption: (1) explicitly mentioned the target category, (2) excluded the source category, and (3) maintained coherence with the original image context. Multiple generation attempts were made when necessary to ensure quality standards were met.

7.4. Complete Change-Pair List

The complete list of 79 change-pairs used in CrossVLAD is provided in table 5

8. Pseudocode

Algorithm 1 presents the detailed procedure of our CRAFT method. The procedure begins by initializing the adversarial example and locating the token indices corresponding to the source object region. In each iteration, we extract image features from the current adversarial example and isolate the features of the target region. We then obtain text embeddings for both positive (target) and negative (source and other) categories. The contrastive loss is computed to align the region features with the target category while pushing them away from negative categories. Finally, we update the adversarial example using the PGD algorithm with the computed gradient.

9. Experimental Details

9.1. Implementation Details of Compared Methods

We provide detailed implementation information for all compared methods to ensure reproducibility and fair comparison.

Attack-Bard We adopt the text description attack from Attack-Bard [6], which maximizes the log-likelihood of predicting a target sentence. For our evaluation, we use

Table 5. Complete source-target object change pairs in CrossVLAD.

Category	Number of Pairs	Source → Target Examples
Vehicle	8 pairs	bicycle → motorcycle, motorcycle → bicycle, car → bus, bus → truck, train → airplane, truck → car, airplane → bus, boat → train
Outdoor	5 pairs	traffic light → stop sign, fire hydrant → stop sign, stop sign → traffic light, parking meter → bench, bench → parking meter
Animal	10 pairs	bird → cat, cat → dog, dog → cat, horse → sheep, sheep → cow, cow → horse, elephant → bear, bear → elephant, zebra → giraffe, giraffe → zebra
Accessory	5 pairs	backpack → handbag, umbrella → handbag, handbag → suitcase, tie → handbag, suitcase → backpack
Sports	10 pairs	frisbee → sports ball, skis → snowboard, snowboard → skateboard, sports ball → kite, kite → baseball bat, baseball bat → baseball glove, baseball glove → tennis racket, skateboard → surfboard, surfboard → skis, tennis racket → frisbee
Kitchen	7 pairs	bottle → wine glass, wine glass → cup, cup → fork, fork → knife, knife → spoon, spoon → bowl, bowl → bottle
Food	10 pairs	banana → apple, apple → orange, sandwich → hot dog, orange → banana, broccoli → carrot, carrot → hot dog, hot dog → pizza, pizza → donut, donut → cake, cake → apple
Furniture	6 pairs	chair → couch, couch → potted plant, potted plant → bed, bed → dining table, dining table → toilet, toilet → chair
Electronic	6 pairs	tv → laptop, laptop → mouse, mouse → remote, remote → keyboard, keyboard → cell phone, cell phone → tv
Appliance	5 pairs	microwave → oven, oven → toaster, toaster → sink, sink → refrigerator, refrigerator → microwave
Indoor	7 pairs	book → clock, clock → vase, vase → scissors, scissors → teddy bear, teddy bear → hair drier, hair drier → toothbrush, toothbrush → book

GPT-generated captions that include the target object category while excluding the source object category. The attack is formulated as:

$$\max_x \sum_{i=1}^N \sum_{t=1}^L \log p_{\theta_i}(y_t | x, p, y_{<t}), \quad \text{s.t.} \quad \|x - x_{nat}\|_{\infty} \leq \epsilon \quad (13)$$

where y_t represents tokens in the target caption, p is the prompt, and θ_i denotes model parameters. We optimize this objective using the PGD algorithm with the same hyperparameters as our primary method.

Mix.Attack Since the original Mix.Attack [31] was designed for untargeted attacks, we modified it for our targeted setting. Our adaptation aligns the adversarial image with the target text while pushing it away from the original descriptions. Specifically, we use three text references: two

from the original MSCOCO caption annotations and one from our generated target caption labels. The optimization objective encourages similarity between the image and target caption representations while reducing similarity with the original captions. All image and text features are extracted using the attacked model’s own encoders to ensure alignment with the model’s internal representations.

MF-it For MF-it [43], we directly compute the similarity between image features and target caption text features, then minimize this similarity through adversarial optimization. This approach attempts to align the perturbed image’s feature representation with the textual representation of the target category. The optimization is performed using PGD with the same constraints as our main experiments.

Algorithm 1 CRAFT: Cross-task Region-based Attack Framework with Token-alignment

Require: Input image I , source object bounding box b_s , source category c_s , target category c_t , perturbation budget ϵ , iteration number T , step size α

Ensure: Adversarial example I_{adv}

- 1: $I_{adv} \leftarrow I$ {Initialize adversarial example}
 - 2: $\mathcal{R} \leftarrow \text{RegionTokenLocalization}(b_s)$ {Localize tokens corresponding to source object}
 - 3: **for** $t = 1$ to T **do**
 - 4: $F_I \leftarrow \text{ImageEncoder}(I_{adv})$ {Extract image feature tokens}
 - 5: $F_R \leftarrow F_I[\mathcal{R}]$ {Extract region feature tokens}
 - 6: $E_{pos} \leftarrow \text{TextEncoder}(c_t)$ {Encode target category}
 - 7: $E_{neg} \leftarrow \text{TextEncoder}(c_s, \text{other categories})$ {Encode negative categories}
 - 8: $\mathcal{L} \leftarrow \text{ContrastiveLoss}(F_R, E_{pos}, E_{neg})$ {Compute alignment loss}
 - 9: $g \leftarrow \nabla_{I_{adv}} \mathcal{L}$ {Compute gradient}
 - 10: $I_{adv} \leftarrow \text{Clip}(I_{adv} + \alpha \cdot \text{sign}(g), I - \epsilon, I + \epsilon)$ {Update with PGD}
 - 11: **end for**
 - 12: **return** I_{adv}
-

MF-ii For MF-ii[43], we first generate a target image using Stable Diffusion [28] with our target caption label as the prompt. We then minimize the feature distance between the adversarial image and this generated target image. This approach attempts to make the adversarial image perceptually similar to an image of the target category while maintaining the ℓ_∞ perturbation constraint.

9.2. Additional Experimental Results

Method	Tasks				Evaluate Metrics
	IC	OD	RC	OL	CTSR-3
TLM-IC	0.935	0.241	0.346	0.296	0.305
TLM-OD	0.451	0.827	0.683	0.518	0.54
TLM-RC	0.523	0.548	0.703	0.724	0.568
TLM-OL	0.244	0.257	0.277	0.736	0.252
CRAFT(ours)	0.765	0.565	0.849	0.649	0.609

Table 6. Performance comparison of CTSR-3 between task-specific Training Loss Minimization (TLM) attacks and our CRAFT method. TLM-IC, TLM-OD, TLM-RC, and TLM-OL represent attacks optimized specifically for Image Captioning, Object Detection, Region Categorization, and Object Location tasks, respectively. Bold values indicate the best performance for each column.

9.2.1. Cross-task Transferability of Task-specific Methods Results with CTSR-3

Table 6 presents the transferability comparison between task-specific TLM attacks and our CRAFT method using

the CTSR-3 metric. These results complement those in Section 5.3 by showing performance when success on at least three tasks is required rather than all four. While the overall trends remain consistent with the CTSR-4 results, the higher CTSR-3 values provide additional insights into partial transferability. Notably, the gap between task-specific attacks and CRAFT is smaller under this more relaxed evaluation criterion, though CRAFT still maintains its advantage, particularly for tasks with spatial components.

9.2.2. Ablation Study with CTSR-3

Figure 6 shows the impact of perturbation budget (ϵ) and iteration count on CTSR-3 performance. The trends broadly mirror those observed for CTSR-4 in Section 5.5, but with higher overall success rates as expected from the more lenient evaluation criterion. The CTSR-3 results further support our choice of $\epsilon = 16/255$ and 100 iterations as the optimal configuration, offering a favorable balance between attack success rate and computational efficiency. Interestingly, the performance plateau and potential decrease with very high iteration counts is less pronounced for CTSR-3, suggesting that overfitting to specific tasks is less problematic when success on only three out of four tasks is required.

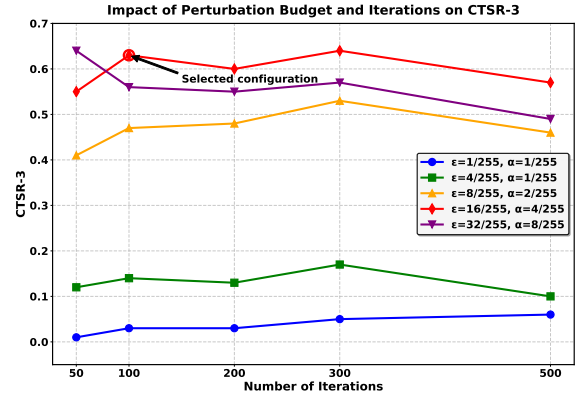


Figure 6. Impact of perturbation budget (ϵ) and iteration count on CTSR-3 performance with Florence-2 model. Each line represents a different ϵ value with corresponding step size α .

9.3. Additional Visualization Examples

Figure 7 illustrates additional successful examples, while Figure 8 complements Section 5.4 by demonstrating that target replacement often becomes unfeasible when modifications occur across categories. However, it can be observed that in most instances, these cross-category alterations also disrupt the original semantic content of the image, enabling untargeted attacks.

9.4. Evaluation on commercial VLMs

While our primary focus is on white-box models where access to model parameters is necessary, we also conducted







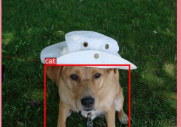



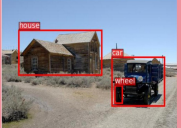







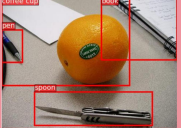

change-pair	x	x'	Image Captioning	Object Detection	Region categorization	Object Localization
zebra ↓ giraffe			A giraffe standing in the snow next to a military vehicle.		giraffe<loc_191> <loc_198><loc_795><loc_904>	
dog ↓ cat			A cat with a toy airplane on its head.		cat<loc_244><loc_505><loc_722><loc_989>	
truck ↓ car			A car parked in front of a wooden building.		car<loc_626><loc_467><loc_921><loc_818>	
toilet ↓ chair			A white office chair sitting on top of a tiled floor.		chair<loc_320><loc_305><loc_701><loc_871>	
knife ↓ spoon			A spoon sitting on top of a table next to a cup.		spoon<loc_192><loc_829><loc_859><loc_982>	

Figure 7. Qualitative examples of CRAFT attack on Florence-2 model across four vision tasks: object detection, object localization, image captioning, and region categorization, with $\varepsilon = 16/255$.



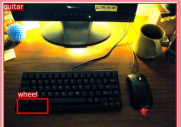



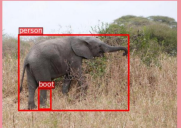







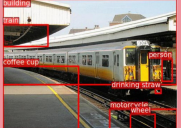





change-pair	x	x'	Image Captioning	Object Detection	Region categorization	Object Localization
keyboard ↓ sheep			A guitar with a picture of a dog on it.		vehicle<loc_75><loc_520><loc_688><loc_838>	
elephant ↓ banana			A woman crouches down in a field of tall grass.		person<loc_89><loc_267><loc_721><loc_829>	
cup ↓ elephant			A person holding a cup of coffee next to a plate of food.		grapefruit<loc_532><loc_208><loc_935><loc_725>	
train ↓ cup			A train station with a bunch of cups on the platform.		building<loc_0><loc_316><loc_921><loc_762>	
horse ↓ keyboard			a vulture with a dead bird in its beak in the grass		bird<loc_471><loc_361><loc_846><loc_867>	

Figure 8. Failed examples of CRAFT attack on Florence-2 model across four vision tasks: object detection, object localization, image captioning, and region categorization, when using cross-category substitutions, with $\varepsilon = 16/255$.

a preliminary evaluation on the black-box transferability of our adversarial attacks to commercial Vision Language Models (VLMs). To this end, we investigate whether adversarial examples generated for an open-source model can successfully deceive a closed-source, commercial model.

Specifically, we generated adversarial examples using Florence-2, a publicly available model, with the objective of causing a cat-to-dog misclassification. These generated images were then presented to GPT-4V, a prominent commercial VLM, to assess the transferability of the attack. As illustrated in Figure 9, the adversarial examples crafted on Florence-2 were effective in misleading GPT-4V, which consequently misidentified the cats in the images as dogs. This successful transfer demonstrates the practical relevance of our attack methodology in a black-box setting, even though it was conducted on a limited scale.

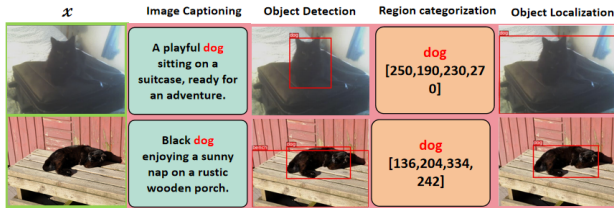


Figure 9. Successful cat-to-dog attacks transferring to GPT-4V.

9.5. Comparison with Object Detection Attack Methods

To further contextualize the performance of our method, we conducted a comparative analysis against established attack methodologies originally designed for object detection.

The results of this comparison on the Florence-2 model are presented in Table 7. The findings indicate that while the adapted baseline methods achieve strong performance on the object detection (OD) task itself, their effectiveness diminishes significantly when transferred to other vision-language tasks such as Image Captioning (IC), Region Categorization (RC), and Object Localization (OL). In contrast, our method, CRAFT, demonstrates superior cross-task effectiveness. Although it shows slightly lower performance on the OD task, it significantly outperforms both baselines across all other evaluated tasks and achieves the highest Cross-Task Success Rates (CTSR-4 and CTSR-3).

Method	IC↑	OD↑	RC↑	OL↑	CTSR-4↑	CTSR-3↑
[2]	0.48	0.75	0.68	0.52	0.38	0.52
[27]	0.45	0.67	0.64	0.50	0.32	0.48
CRAFT (ours)	0.77	0.57	0.85	0.65	0.47	0.61

Table 7. Comparison with object detection attack baselines on Florence-2. Our method demonstrates superior cross-task transferability.

9.6. Effect of Bounding Box Source

In our primary experiments, we assume access to ground-truth bounding boxes to define the target regions for our

attacks. To assess the practical applicability of our method in scenarios where such ground-truth data is unavailable, we conducted an ablation study to evaluate the impact of using bounding boxes generated by a state-of-the-art object detector.

For this analysis, we replaced the ground-truth bounding boxes with boxes detected by YOLOv10. We then performed the same attack procedure on the Florence-2 model. The comparative results are detailed in Table 8. The data shows that the performance difference between using YOLOv10-detected boxes and ground-truth boxes is minimal across all evaluated tasks. This experiment demonstrates that our attack’s effectiveness is not contingent on having perfect bounding box information and that comparable results can be achieved using high-quality, readily available object detectors.

Box Source	IC↑	OD↑	RC↑	OL↑	CTSR-4↑	CTSR-3↑
YOLOv10	0.75	0.53	0.81	0.64	0.45	0.60
Ground-truth	0.77	0.57	0.85	0.65	0.47	0.61

Table 8. Performance comparison using bounding boxes from a SOTA detector (YOLOv10) versus ground-truth boxes. The minimal difference validates our experimental assumption.