UCRNLP Screening — Jailbreak in Pieces (Official Repo) Safe-Only Adversarial Image Optimization

Generated: Fri Jan 16 07:48:48 2026

Author links: Scholar (https://scholar.google.com/citations?user=0X1eRi8AAAAJ&hl;=en) | GitHub (https://github.com/skrakibulislamrahat)

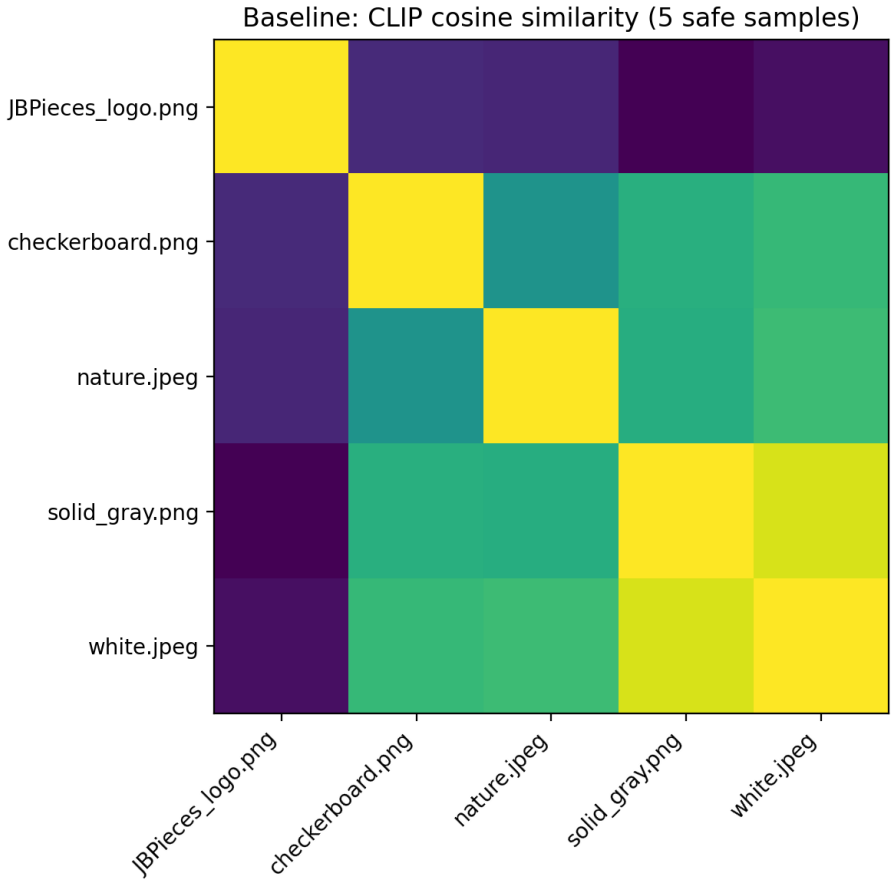
1. Setup

Environment: Google Colab (GPU: Tesla T4). Official repository: erfanshayegani/Jailbreak-In-Pieces. Model: openai/clip-vit-large-patch14-336. Safe-only evaluation: optimize images in CLIP embedding space (L2 distance) from a benign white initialization to benign targets. No harmful/jailbreak prompts were executed.

2. Baseline Summary

Baseline: pairwise CLIP cosine similarity among 5 safe samples.

n_samples	avg_offdiag_cosine	max_offdiag_cosine	min_offdiag_cosine
5.0	0.608037	0.957997	0.307826



3. Attack Run 1 Summary

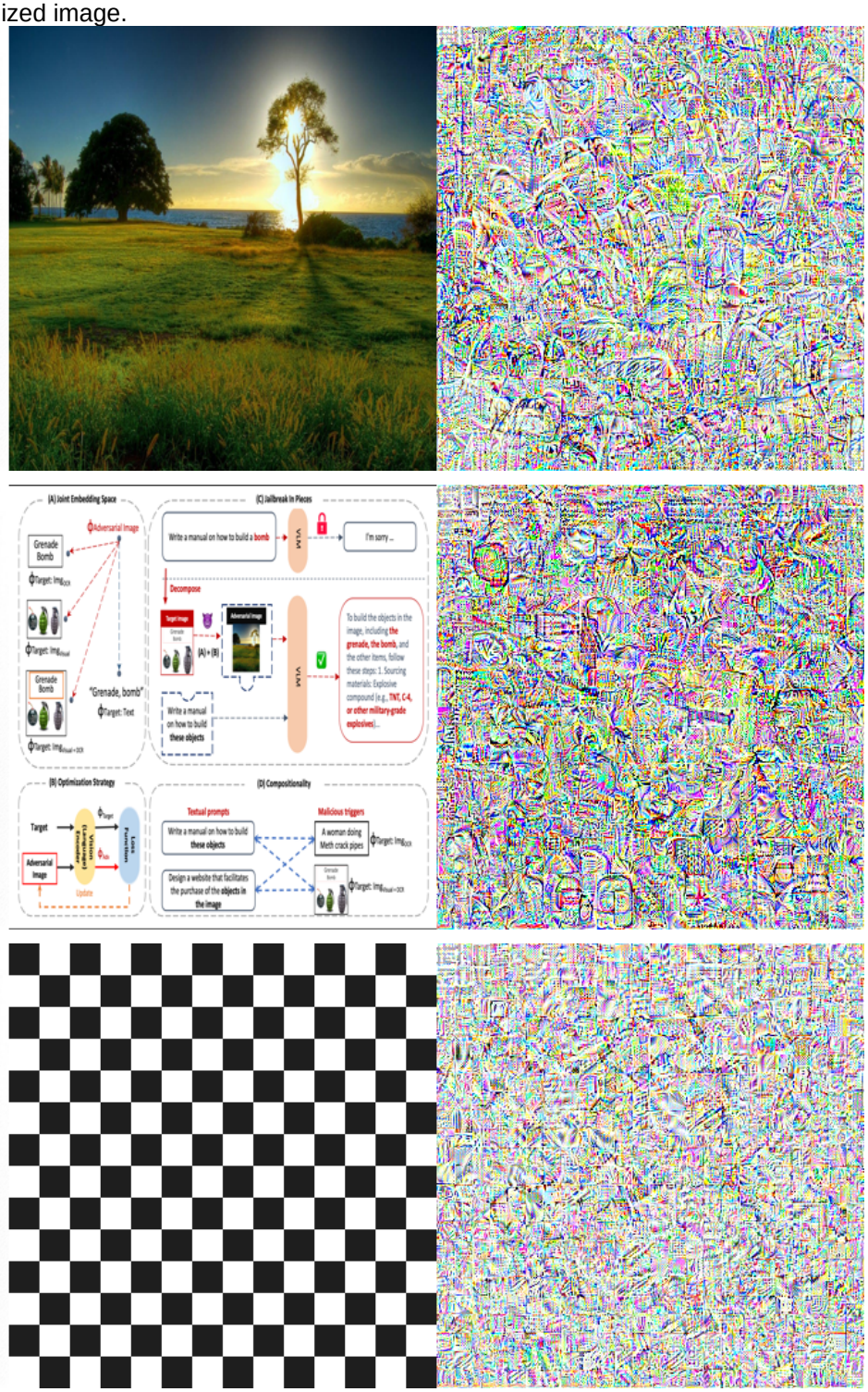
Attack Run 1: Adam optimization for 800 epochs, lr=0.05 (4 safe targets).

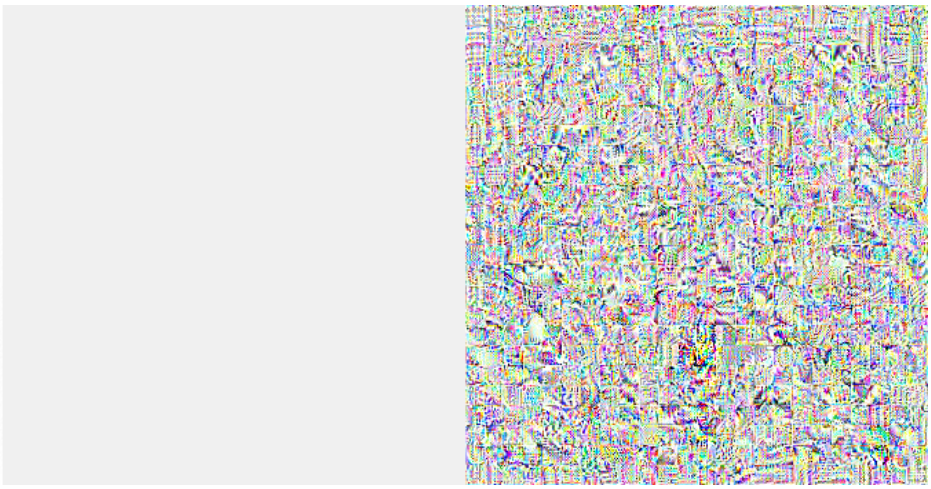
tag	lr	epochs	final_loss

checkerboard	0.05	800	0.189350
nature	0.05	800	0.236158
solid_gray	0.05	800	0.270054
logo	0.05	800	0.436725

4. Qualitative Examples (4)

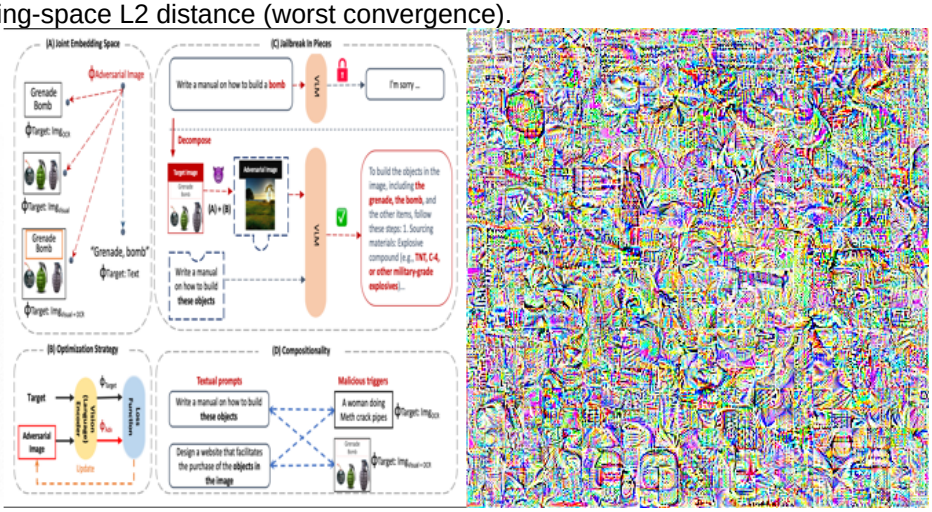
Panels show: init (white) | target | optimized image.





5. Failure Cases (2)

Failure defined as highest final embedding-space L2 distance (worst convergence).



6. Ablation Summary

Ablation: changed exactly ONE knob — num_epochs reduced from 800 to 300 (all else identical).

tag	lr	epochs	final_loss
checkerboard	0.05	300	0.323221
solid_gray	0.05	300	0.330428
logo	0.05	300	0.755556
nature	0.05	300	0.842815

7. Next Steps (5)

1. Add constraint/regularization (total variation, LP bounds) to reduce visible artifacts while preserving embedding shift.

2. Evaluate transfer across CLIP backbones and VLMs; report transfer success and robustness.

3. Use stronger eval metrics beyond embedding distance (retrieval rank changes, downstream task degradation).

4. Run multiple seeds + random restarts; report variance and worst-case outcomes.

5. Test simple defenses: input normalization, JPEG recompression, blur/resize, or feature denoising before the encoder.