



Adversarial Attacks are Reversible with Natural Supervision

Kuan-Yao Huang kh3120@columbia.edu
Sujith Reddy Mamidi sm5116@columbia.edu



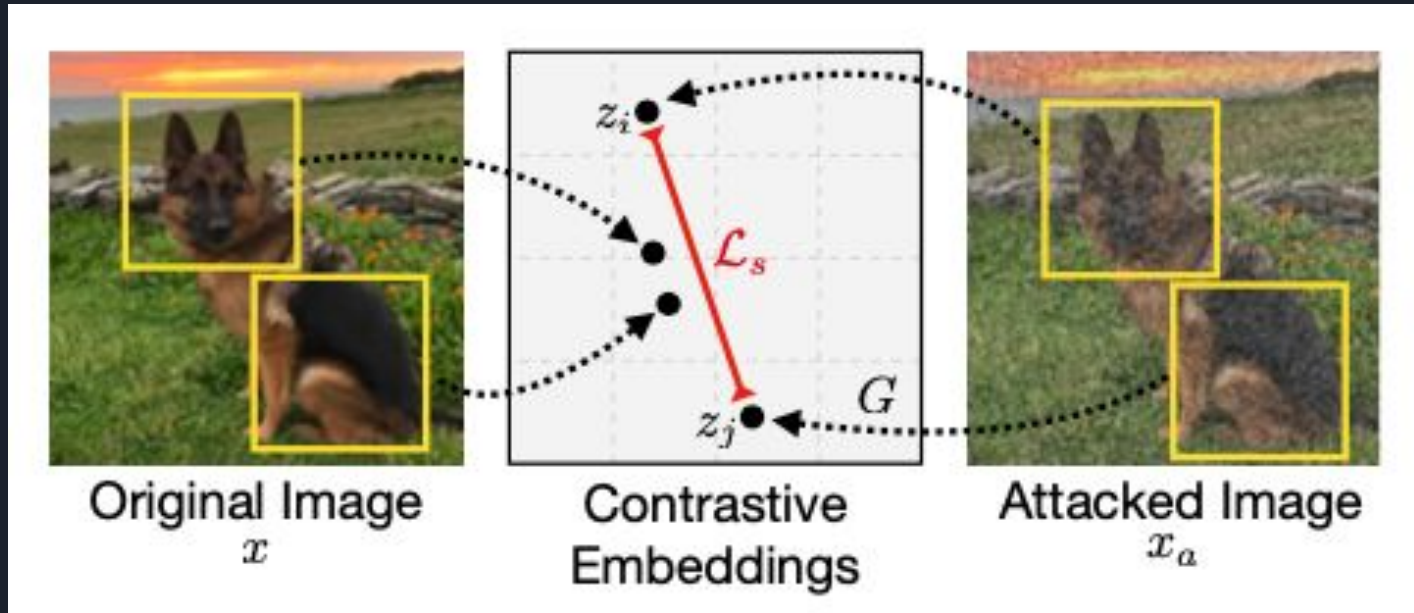
Execution Summary

The goal of this project is to incorporate self-supervised learning and contrastive learning to improve model's robustness on adversarial dataset at inference time.

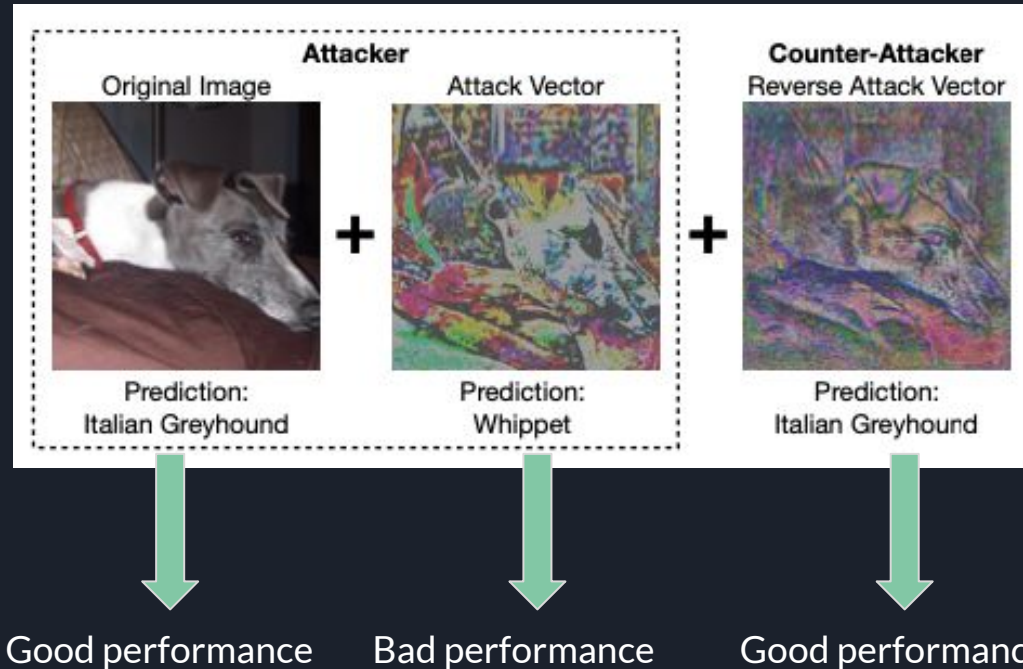
Value and benefit of the design:


1. It works as long as the corruption violates natural manifolds
2. No need to retrain the backbone model
3. It works with almost all CNN based deep learning classifier

Problem motivation: Adversarial attacks can reduce the **mutual information** of similar images



Question: Can restoring the mutual information improve the performance of the target task?





Background work: State-of-the-art counter-attack models

- Semi-SL
- TRADES
- Robust Overfit (RO)
- Bags of tricks (BOT)
- MART
- Fast is Better than Free (FBF)

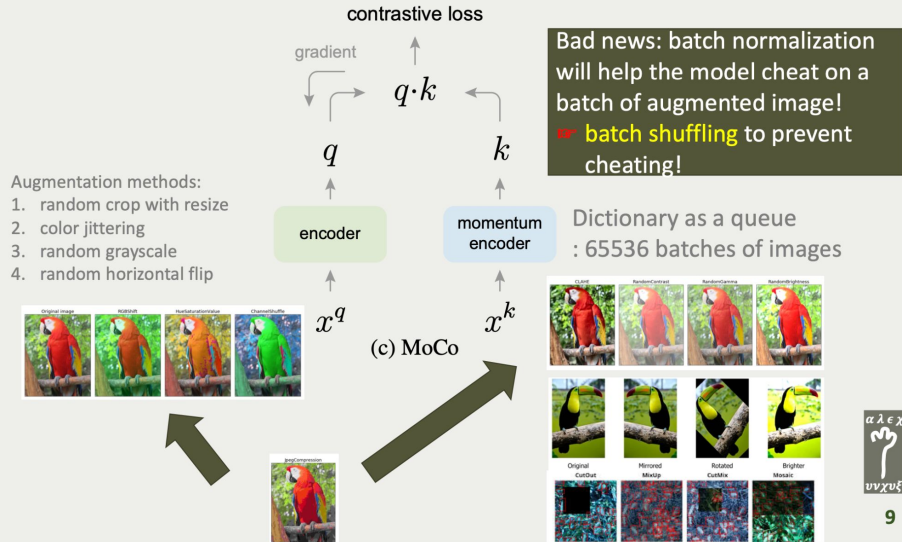


Challenges

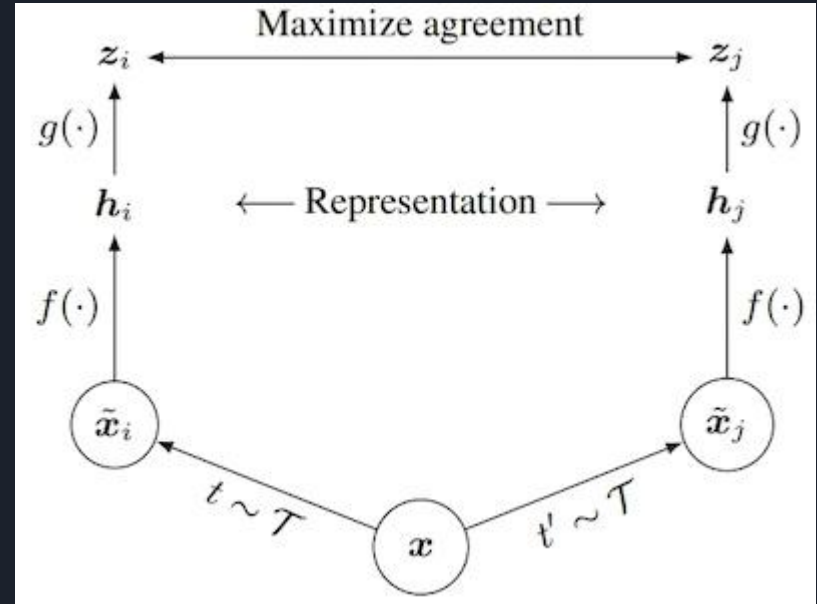
1. Building a robust representation through contrastive learning usually requires a large amount of negative examples and requires a large memory: → Using simCLR, provide positive example and negative from 2+ batches of images
2. How to repair the image at inference time: optimized an added trainable noise on input image to minimize the contrastive loss

Challenges

MOCO requires a memory bank, which is too memory intensive.



SimCLR can utilize images **in the same batch** as positive and negative example





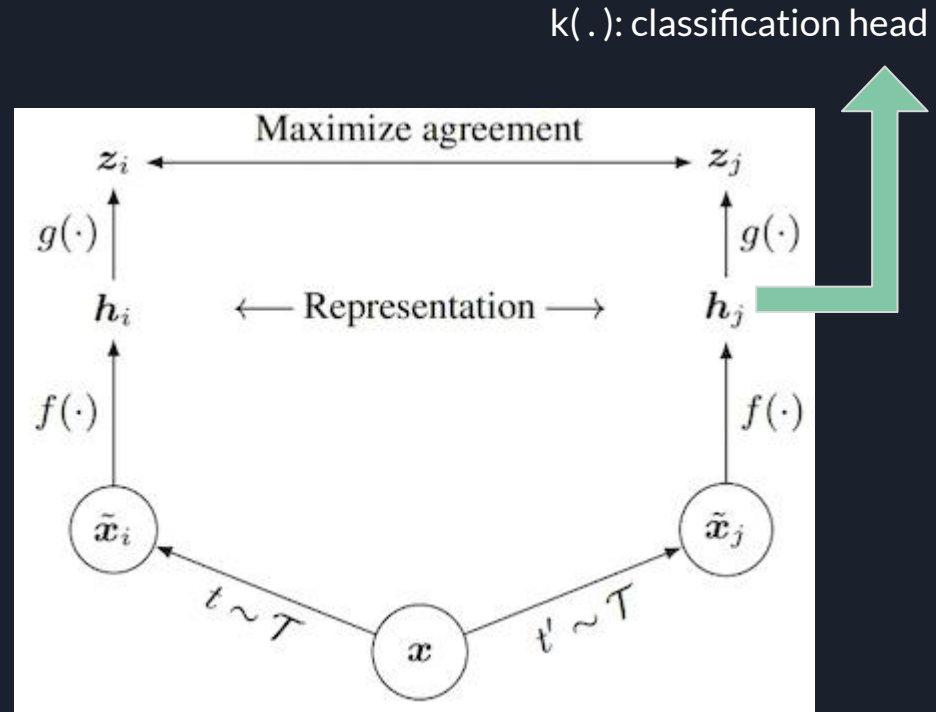
Step 0: Attack by projected gradient descent

Project Gradient Descent(PGD):

- Attacker has a copy of the model
- Uses inference from model to increase loss
- Constrained as Convex Optimization Problem with a loss function
- Move away from gradient
- 10 iterations of Attack
- Higher the epsilon value, higher the perturbations and lower the accuracy

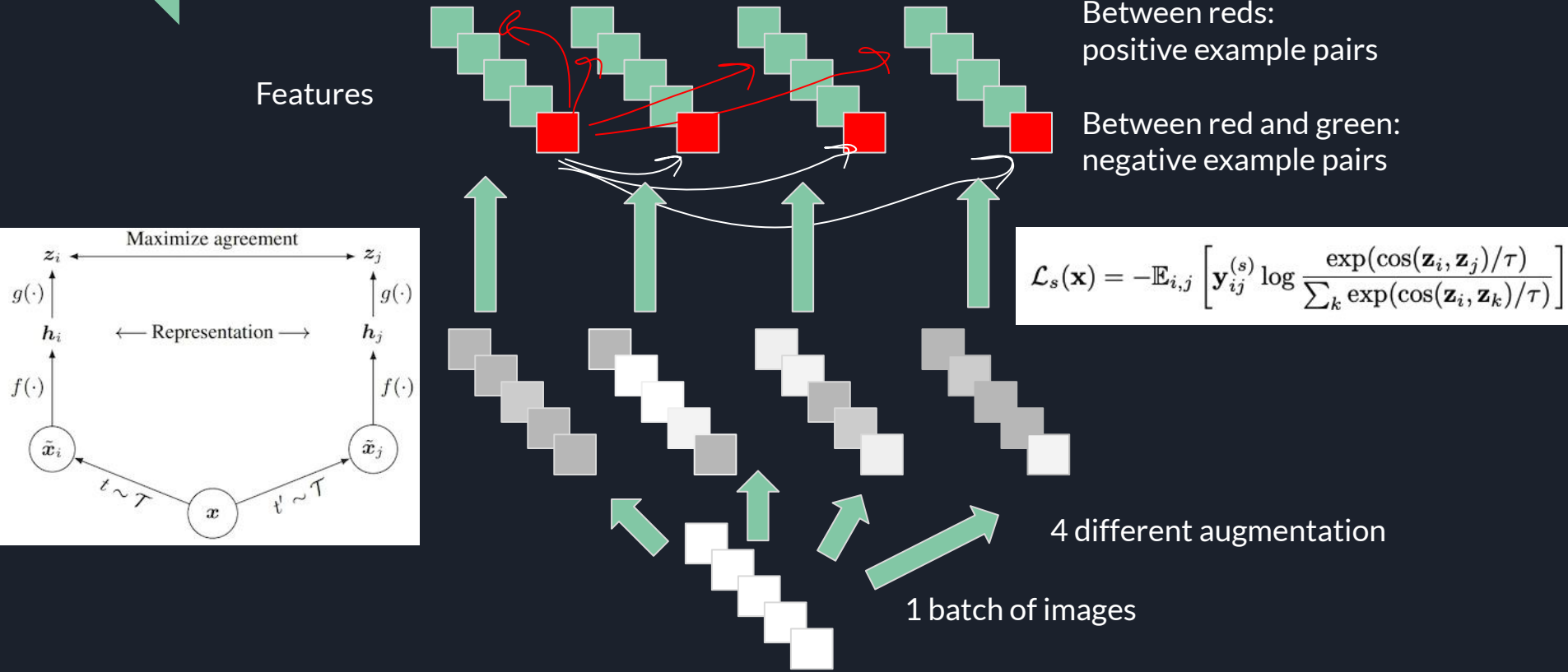
Model Architecture

- $f(\cdot)$: **WideResNet(depth=28, width=10)** all layers before fully connected layers. Model weight from semi-SL paper
- $g(\cdot)$: **2 linear layers**, performing dimension reduction to 16 features
- $k(\cdot)$: classification head: 2 linear layers 10 classes
- Contrastive Loss $L(z_i, z_j)$: computed from 4 batches of 1024 images(4096 images in total)
 - 3 positive example for each image
 - 4092 - 3 negative example for each image

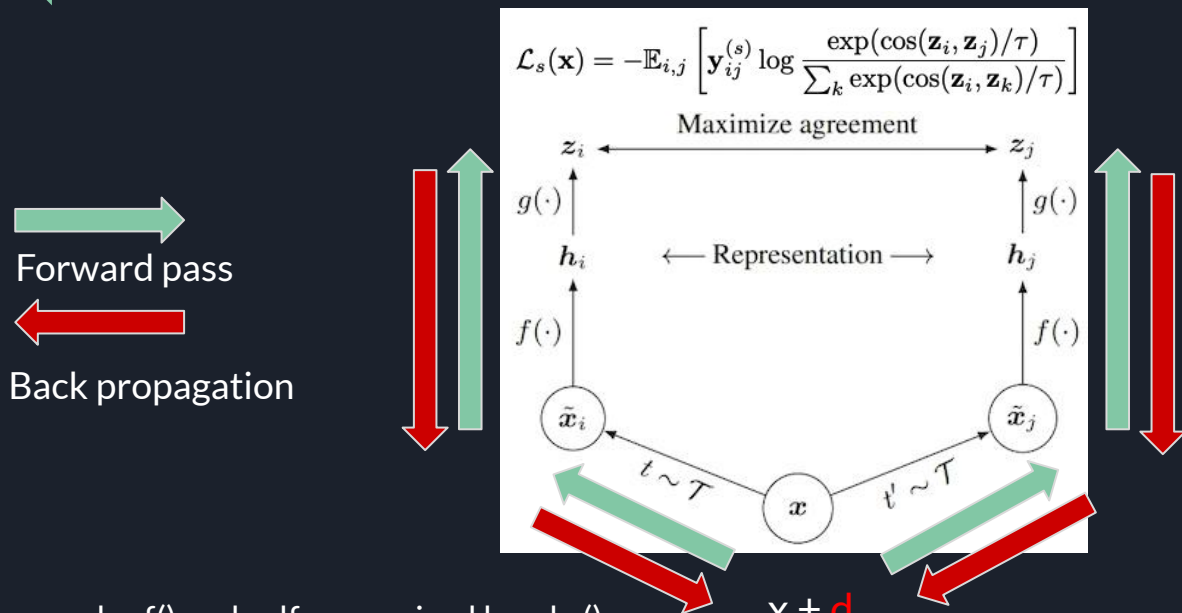


SimCLR by Google

Approach(1): Train the self-supervised head



Approach(2): counter-attack by minimizing the contrastive loss



Remark 1: encoder $f()$ and self-supervised head $g()$ are FREEZED during the optimization process.

Remark 2: we clamp $(x + d)$ to be within $[0, 1]$

$x + d$

x : fixed, a batch of test image (256)
 d : trainable, initialized with noise



Implementation Details

Hardware: using 4v CPU + 1 T4 GPU

Platform: Google Cloud Console

Software(Framework): PyTorch 1.11

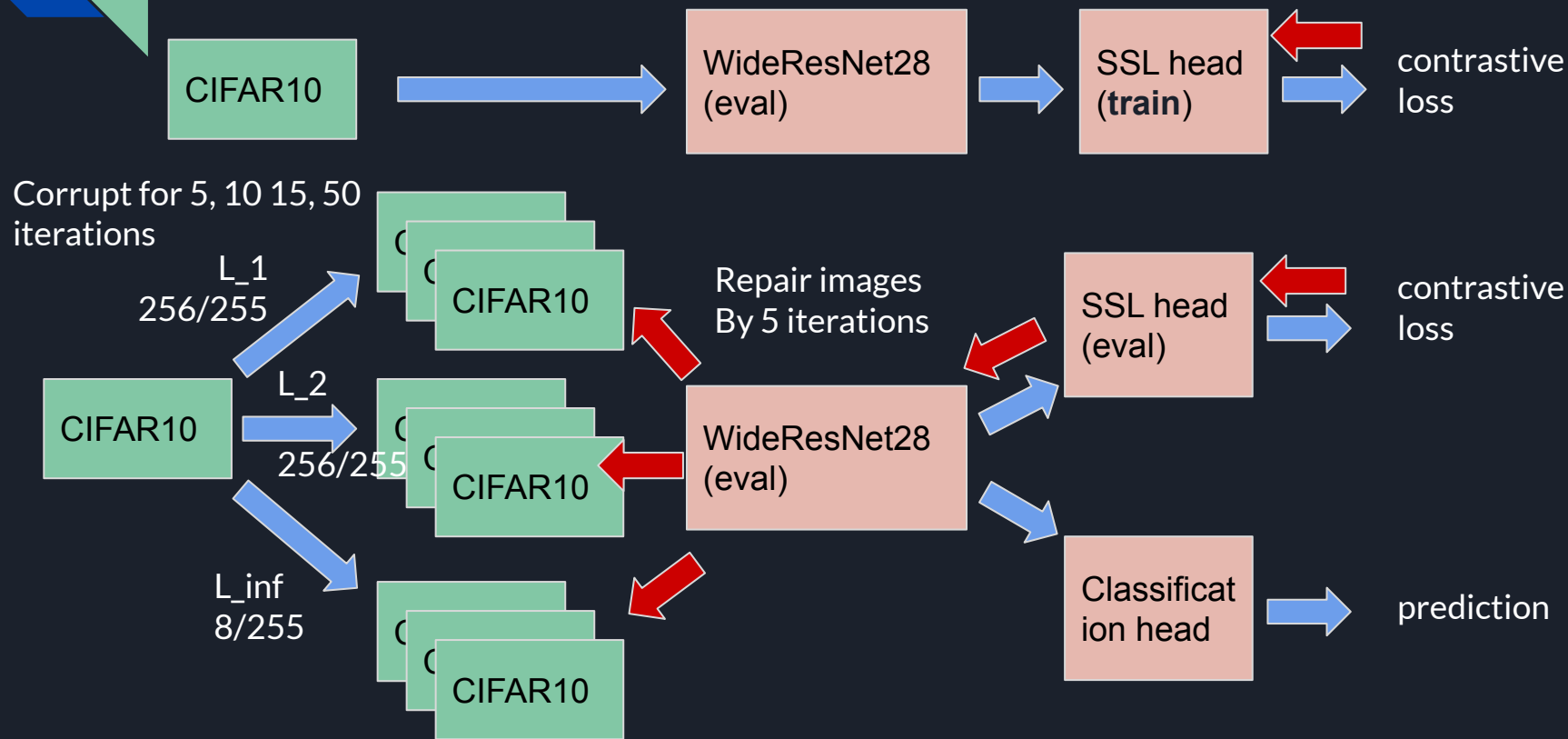
Dataset: CIFAR10

Functionalities:

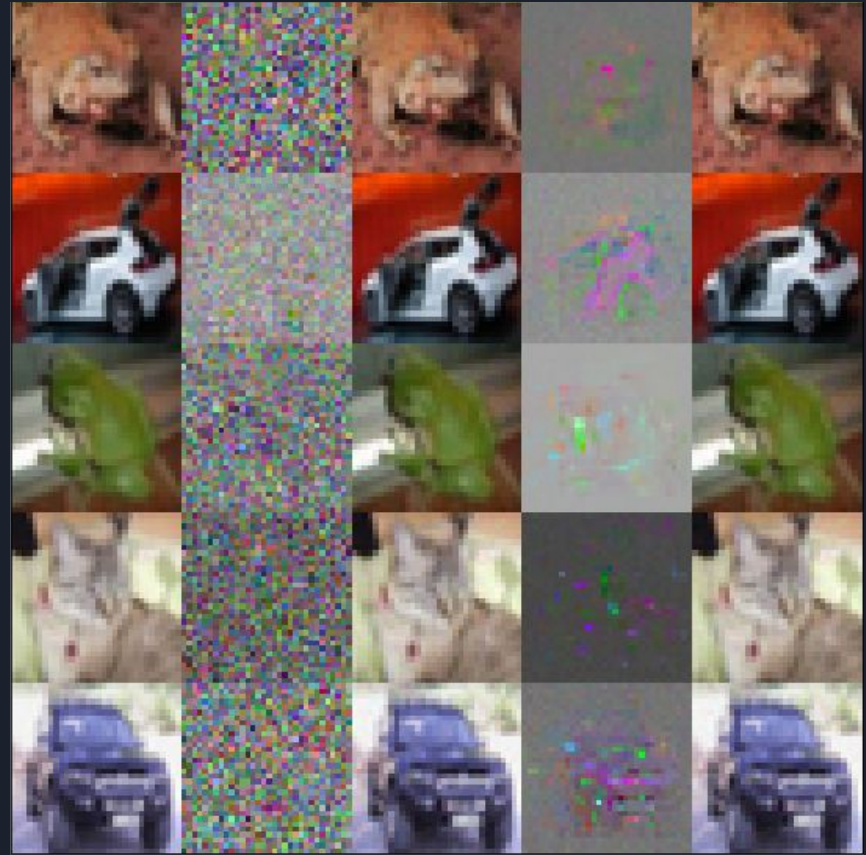
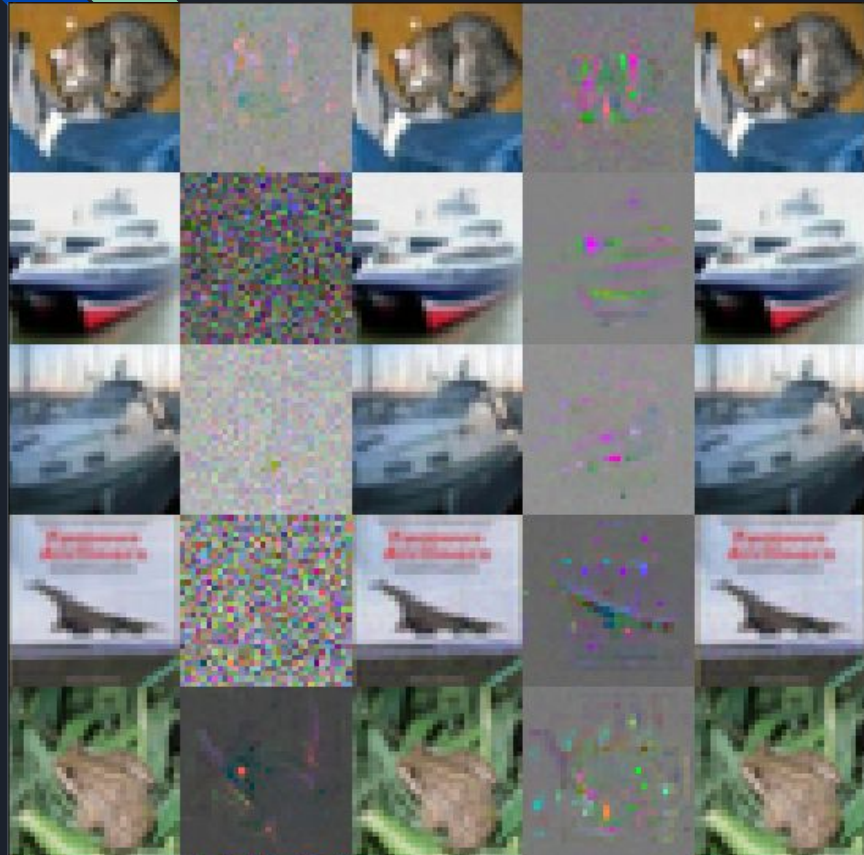
1. Perform projected gradient descent attack on CIFAR10 by 5, 10, 15, 50 iterations using l_1 , l_2 , l_{∞} norm
2. Train self-supervised head with a learning rate of $1e-4$, Adam optimizer, multistep scheduler at $\frac{1}{2}$ epochs scaled by 0.1, batch size 1024, 4 views per images.
3. Inference on the attacked data with 5 iterations of self-supervised repair.

Limitations: may not be able to inference with batch size > 256

Experiment Design Flow



Evaluation(1): Visualize the attack/reverse attack vector

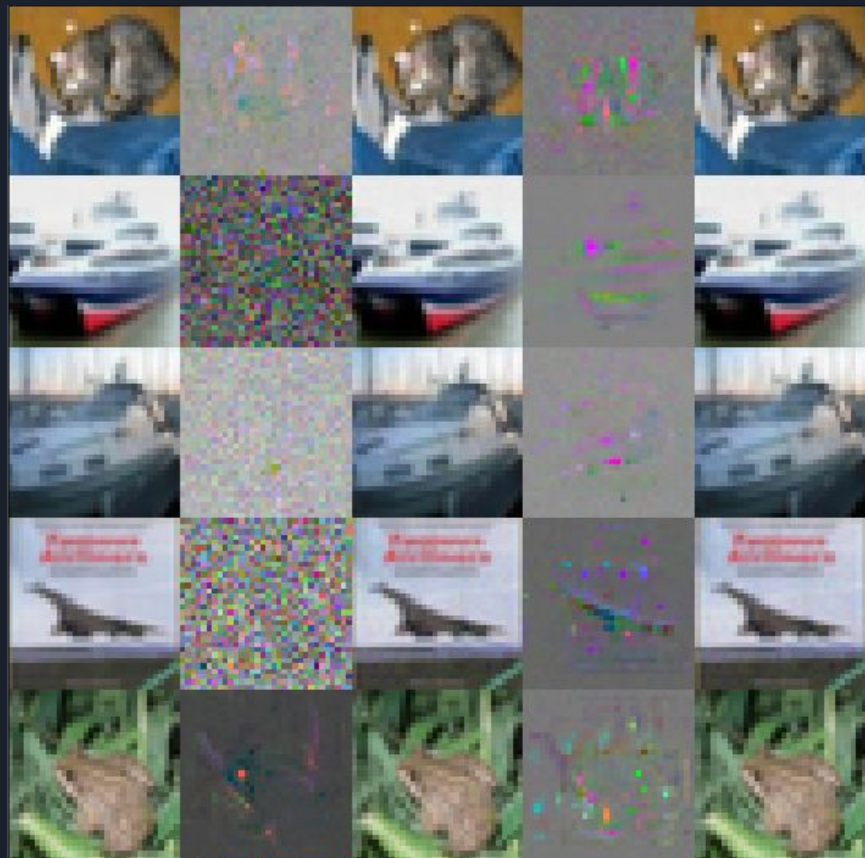


Evaluation(2): Visualize the attack/reverse attack vector

We observe the attack vector may have some saliency over crucial parts such as **the eyes of the animal**.

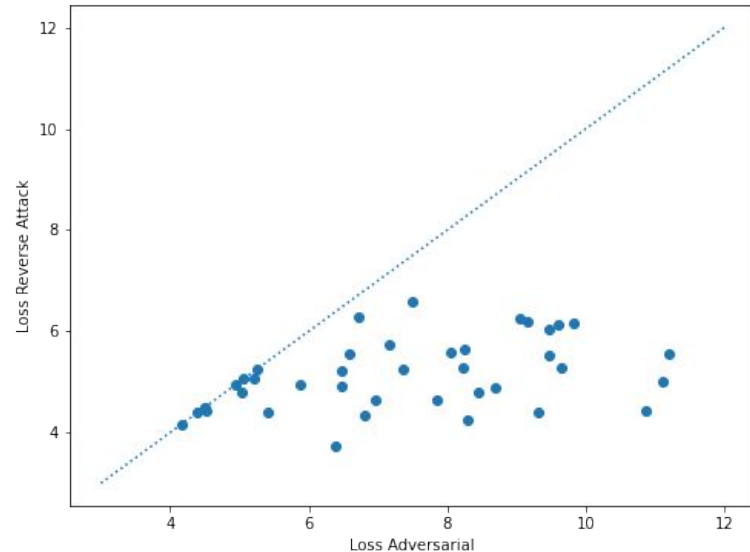
On the other hand, the counter-attack also **shows saliency over these location**.


Does that imply the counter-attack may try to repair the image by making the image more homogeneous?



Evaluation(3): Comparing the contrastive loss distribution before and after the counter-attack

- We perform 10 iterations of counter-attack to “repair” the images at test time.
- Contrastive loss is improved after the counter-attack





Evaluation(4) ~1% performance gain on robust accuracy when the perturbation is moderate

Attack 5 epochs+ counter attack 5 epochs

Perturbation	Baseline Accuracy (%)	Baseline Test Loss	Robust Accuracy(%)	Robust Test Loss
L1	89.64	0.5272	89.64	0.5272
L2	88.71	0.5441	88.70	0.5441
Linf	74.08	0.8110	74.94	0.8048

Attack 10 epochs + counter attack 5 epochs

Perturbation	Baseline Accuracy (%)	Baseline Test Loss	Robust Accuracy(%)	Robust Test Loss
L1	89.58	0.5280	89.59	0.5280
L2	87.74	0.5606	87.73	0.5605
Linf	70.23	0.8850	71.17	0.8725

Attack 15 epochs + counter attack 5 epochs

Perturbation	Baseline Accuracy (%)	Baseline Test Loss	Robust Accuracy(%)	Robust Test Loss
L1	89.54	0.5288	89.54	0.5288
L2	86.89	0.5781	86.89	0.5780
Linf	70.20	0.8888	70.96	0.8821



Conclusion

1. We can observe from the visualization that the **reverse attack vector is trying to repair the images from adversarial attack**.
2. Slight improvement on robust accuracy and loss was observed.
3. We have observed a **significant improvement in contrastive loss** on attacked images after correcting them with natural supervision.
4. Increased perturbation budget(epsilon, iteration) can limit the gain from the self-supervised correction.
5. Given the limitation of GPUs, we inferenced with a batch size of 256, but a higher batch size might have resulted in a better self-supervised correction.