# Supplemental Material: A Hierarchical Assessment of Adversarial Severity

## iNaturalist-H Statistics

We present the statistics of iNaturalist-H on table 1. This dataset contains 189404 images for training, 42140 for validation, and 42756 for testing. The dataset includes a broad span of classes, ranging from animals to plants. The imbalance is extremely high: the least number of images per category is 13, and the maximum is 352. Furthermore, the standard deviation of the number of leaf nodes per father node is enormous, showing imbalance even on supernodes.

| Level | Mean | std | Nodes |
|---|---|---|---|
| Kingdom | 336.67 | 273.92 | 3 |
| Phylum | 252.50 | 255.32 | 4 |
| Class | 112.22 | 169.45 | 9 |
| Order | 29.71 | 17.49 | 34 |
| Family | 17.72 | 9.73 | 57 |
| Genus | 14.02 | 4.95 | 72 |
| Species | 1 | 0 | 1010 |

Table 1: **Node statistics of iNaturalist19.** The iNaturalist presents a high imbalance on both number of instances per class and number of leaf classes descendant from the nodes. Also we present the number of nodes on each level.

## Hierarchical AutoAttack

Our proposed attacks optimize the adversaries based on the probabilities of some chosen classes. So we adapted the AutoAttack benchmark to create the adversarial examples based on these chosen categories. We name this approach Hierarchical AutoAttack. For this experiment, we enhanced the AutoAttack with the NHA@3 attack. Furthermore, to boost the induced mistake, we harm all instances where the input image is correctly classified on the supernode at height $h$. We decided to choose the NHA attack as it provides the best Average Mistake increase. Also, we set $h = 3$ since we consider that this setting provides the best trade-off between Accuracy and Adversarial Mistake. We chose to avoid experimentation over all the values of $h$ because AutoAttack is computational slow. We report the results of the Node-based Hierarchical AutoAttack (NHAA) on table 2. The results show that CHAT is a defense mechanism that enjoys better protection against NHAA@3.

| $\epsilon$ | C | Acc | AM |
|---|---|---|---|
| 4 | | 16.65 | 4.32 |
| 4 | ✓ | 17.77 | 4.21 |
| 6 | | 9.19 | 4.63 |
| 6 | ✓ | 10.76 | 4.56 |
| 8 | | 4.97 | 4.92 |
| 8 | ✓ | 7.08 | 4.76 |

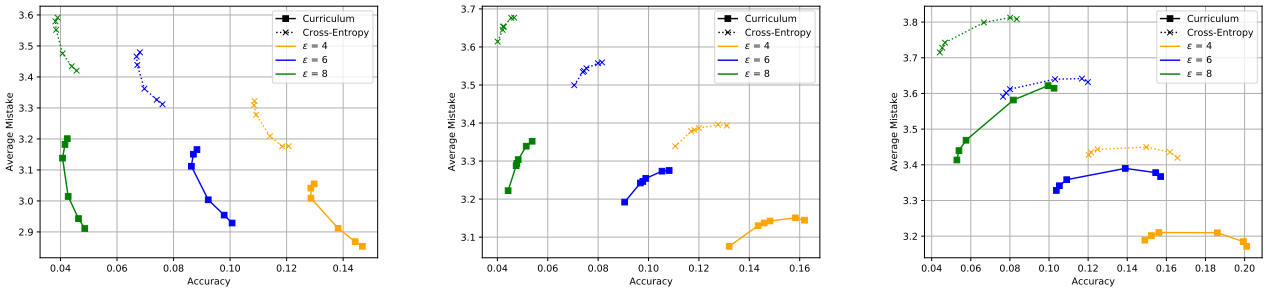
Table 2: **Hierarchical AutoAttack.** We tested the AutoAttack enhanced with NHA@3. Similarly to all results on the main manuscript, our implementation enhances the robustness and reduce the severity of adversarial attacks.

## CHAT-enhanced TRADES

For completeness, we train a model with TRADES and CHAT-enhanced TRADES (CHATeT). To achieve this CHATeT model, we replaced the second step of our curriculum with the traditional TRADES. Thus, it is necessary to minimizes the objective on Equation 1, where $L_{KL}$ is the Kullback-Leibler divergence, $L_{CE}$ the traditional cross entropy loss, and $\beta$ is a constant:

$$min \left\{ L_{CE}(f(x), y) + \max_{x' \in B_{\ell_\infty}(x,\epsilon)} \beta L_{KL}(f(x), f(x')) \right\} \tag{1}$$

For $\epsilon \in \{6/255, 8/255\}$, we trained the standard and CHAT-enhanced models with a step size $\alpha$ of $3/255$ and $4/255$ for 3 or 4 iterations $I$. Then we choose the best performing model, which are $(\alpha, I) = (3/255, 3)$ for TRADES and $(4/255, 4)$ for CHATeT. For $\epsilon = 4/255$, we used $\alpha = 1$ with $I = 4$ for the vanilla and enhanced model. Further, we set the constant $\beta = 6$ for all models. We pick the value of $\beta$ through a hyperparameter search with a perturbation budget of $\epsilon = 4/255$. On table 3 we display the results of the experiments with CHATeT with PGD50 and on Figure 1 the results against the hierarchical-aware attacks, both on the validation set. The results show that CHAT does not




(a) LHA Assessment (b) GHA Assessment (c) NHA Assessment

Figure 1: **TRADES Hierarchical Attack Evaluation.** We evaluate the performance of TRADES on our hierarchical attacks with 50 iterations at each target height on the validation set. The inclusion of our curriculum boost all metrics for all levels compared to the baseline. The results show similar behaviour than the results on Figure 3 of the manuscript.

| $\epsilon$ | C | Clean | | PGD | |
|---|---|---|---|---|---|
| | | Acc | AM | Acc | AM |
| 4 | | 23.90 | 3.33 | 11.06 | 3.34 |
| 4 | ✓ | 29.23 | 3.03 | 13.20 | 3.08 |
| 6 | | 21.26 | 3.49 | 7.04 | 3.50 |
| 6 | ✓ | 27.91 | 3.14 | 9.05 | 3.19 |
| 8 | | 19.94 | 3.59 | 4.01 | 3.61 |
| 8 | ✓ | 29.84 | 3.11 | 4.47 | 3.22 |

Table 3: **Effect of CHAT with TRADES.** Enhancing TRADES with our proposed mechanism boost the accuracy performance and decreases the average mistake on both clean and adversarial settings.

solely boost the performance for FAT but also TRADES. To our surprise, the CHATeT's clean accuracy was higher among all models. We suspect that our coarse hyperparameter search did not yield the best performance. Nonetheless, we achieve a performance gain on both metrics.

## Adversarial Images

We present in table 4 the results on the validation set of the hierarchical attacks against FAT and CHAT. Furthermore, we display some adversarial images created by all our attacks. We show the original image, the adversary, and the noise. We only display the adversaries of the best model for $\epsilon = 8$ to visualize the perturbation on the image. Figures 2 to 9 display adversaries generated by NHA. The adversaries for GHA are between Figure 10 and 16. Lastly, Figures 17 to 23 show some LHA adversaries.

| | Level | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\epsilon$ | C | Acc | AM | Acc | AM | Acc | AM | Acc | AM | Acc | AM | Acc | AM |
| LHA | 4 | | 14.64 | 2.98 | 14.39 | 2.99 | 13.80 | 3.02 | 12.39 | 3.12 | 12.23 | 3.15 | 12.25 | 3.17 |
| | 4 | ✓ | 16.32 | 2.81 | 16.01 | 2.82 | 15.36 | 2.86 | 13.60 | 2.97 | 13.36 | 3.01 | 13.33 | 3.02 |
| | 6 | | 8.99 | 3.23 | 8.78 | 3.23 | 8.33 | 3.27 | 7.29 | 3.36 | 7.09 | 3.39 | 7.08 | 3.41 |
| | 6 | ✓ | 10.71 | 3.06 | 10.49 | 3.07 | 9.92 | 3.10 | 8.53 | 3.20 | 8.36 | 3.24 | 8.32 | 3.26 |
| | 8 | | 5.54 | 3.53 | 5.45 | 3.54 | 5.13 | 3.57 | 4.31 | 3.64 | 4.24 | 3.67 | 4.27 | 3.69 |
| | 8 | ✓ | 7.65 | 3.27 | 7.52 | 3.28 | 7.05 | 3.32 | 6.11 | 3.41 | 5.92 | 3.45 | 5.98 | 3.46 |
| GHA | 4 | | 12.33 | 3.20 | 14.71 | 3.32 | 15.07 | 3.33 | 15.56 | 3.34 | 17.52 | 3.34 | 18.42 | 3.33 |
| | 4 | ✓ | 13.36 | 3.05 | 15.97 | 3.18 | 16.47 | 3.19 | 17.01 | 3.20 | 19.33 | 3.21 | 20.39 | 3.19 |
| | 6 | | 7.13 | 3.44 | 8.55 | 3.54 | 8.82 | 3.55 | 9.14 | 3.56 | 10.69 | 3.58 | 11.60 | 3.58 |
| | 6 | ✓ | 8.34 | 3.29 | 10.20 | 3.40 | 10.55 | 3.41 | 10.89 | 3.43 | 12.75 | 3.44 | 13.79 | 3.44 |
| | 8 | | 4.31 | 3.71 | 5.21 | 3.79 | 5.35 | 3.79 | 5.53 | 3.80 | 6.60 | 3.82 | 7.32 | 3.82 |
| | 8 | ✓ | 6.05 | 3.49 | 7.28 | 3.58 | 7.50 | 3.58 | 7.81 | 3.60 | 9.26 | 3.62 | 10.10 | 3.62 |
| NHA | 4 | | 15.12 | 3.38 | 15.59 | 3.39 | 16.39 | 3.41 | 20.56 | 3.39 | 22.54 | 3.39 | 23.12 | 3.36 |
| | 4 | ✓ | 16.28 | 3.23 | 16.87 | 3.25 | 17.58 | 3.27 | 22.03 | 3.27 | 23.92 | 3.24 | 24.46 | 3.22 |
| | 6 | | 8.63 | 3.59 | 8.88 | 3.60 | 9.29 | 3.62 | 12.85 | 3.66 | 14.71 | 3.65 | 15.24 | 3.63 |
| | 6 | ✓ | 10.29 | 3.45 | 10.54 | 3.47 | 11.09 | 3.49 | 15.20 | 3.52 | 16.96 | 3.51 | 17.49 | 3.49 |
| | 8 | | 5.27 | 3.84 | 5.39 | 3.85 | 5.65 | 3.87 | 8.06 | 3.92 | 9.93 | 3.91 | 10.27 | 3.90 |
| | 8 | ✓ | 7.21 | 3.63 | 7.49 | 3.64 | 7.84 | 3.66 | 11.00 | 3.71 | 12.92 | 3.70 | 13.46 | 3.69 |

Table 4: **Hierarchical Attack Evaluation.** We evaluate the performance of FAT on our hierarchical attacks with 50 iterations on each level. The inclusion of our curriculum boost all metrics for all levels compared to the baseline. Acc stands for Accuracy and AM Average Mistake. LHA is the strongest accuracy-related attack among the proposed ones. It even surpasses the PGD when setting the level $l$ to 5 or 6. The GHA enjoys a balanced between severity and accuracy. The NHA is the most severe but less successful attack among the set of hierarchical attacks.

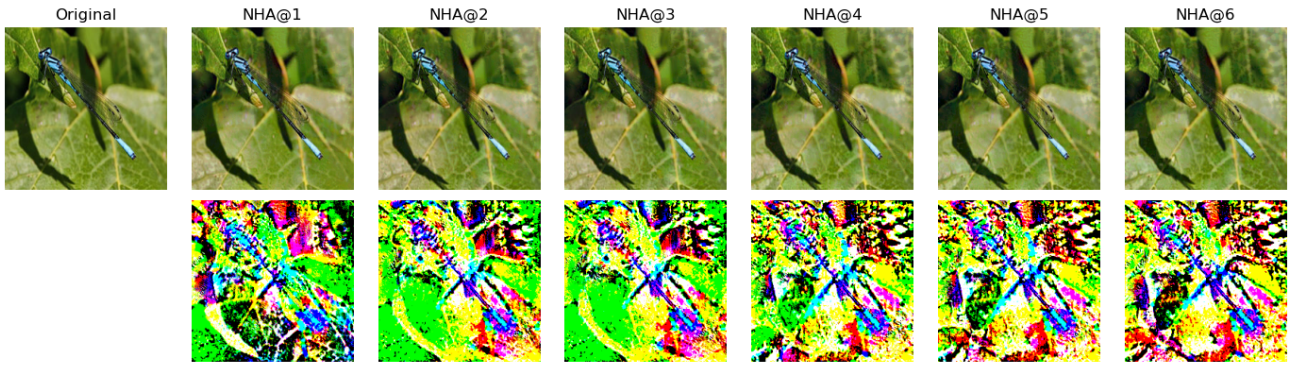



Figure 2: NHA Adversarial examples

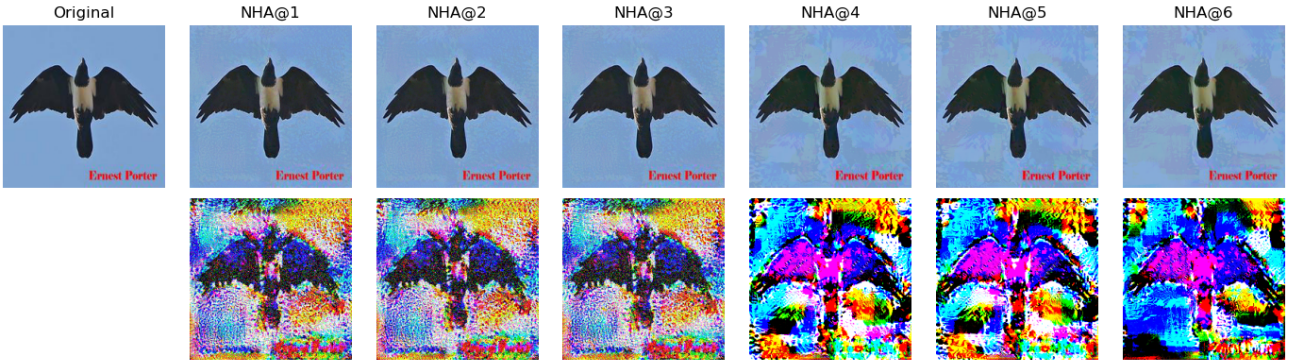



Figure 3: NHA Adversarial examples

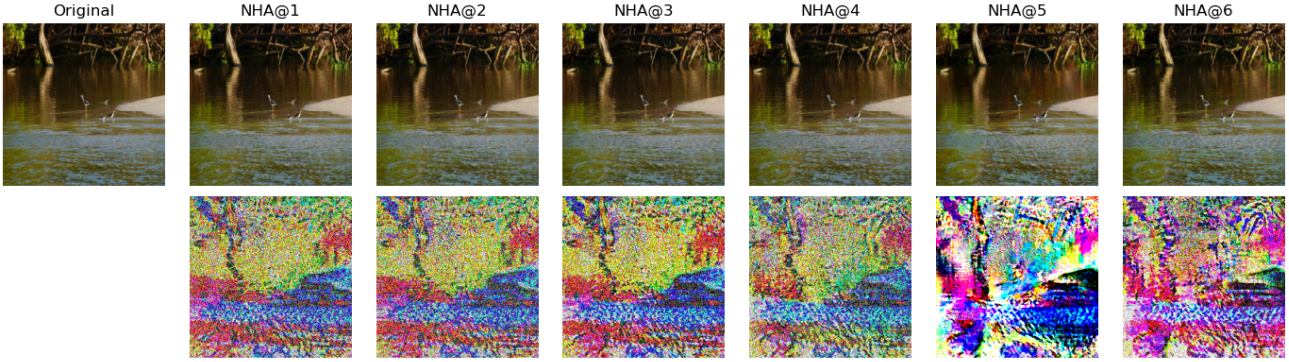



Figure 4: NHA Adversarial examples

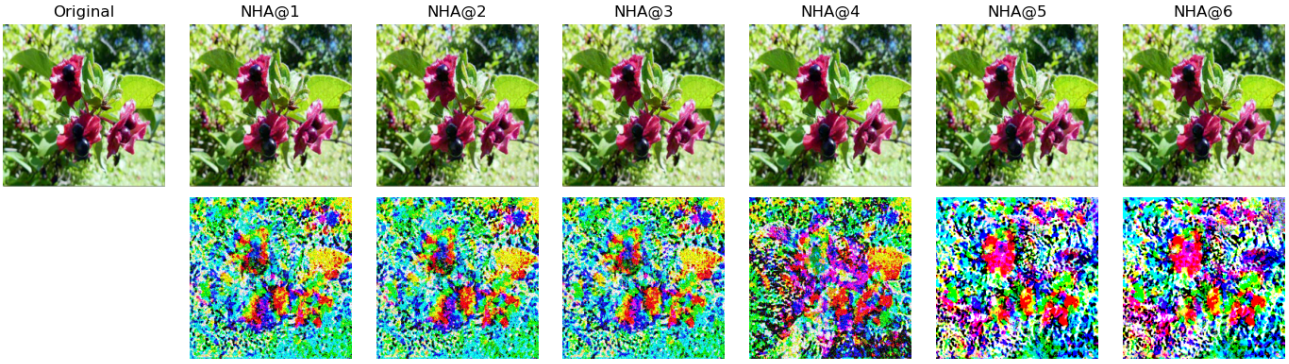



Figure 5: NHA Adversarial examples

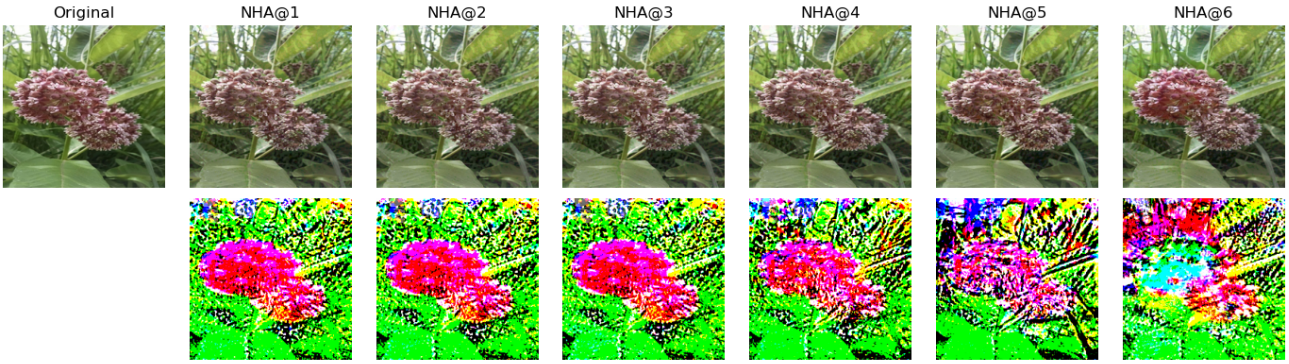



Figure 6: NHA Adversarial examples

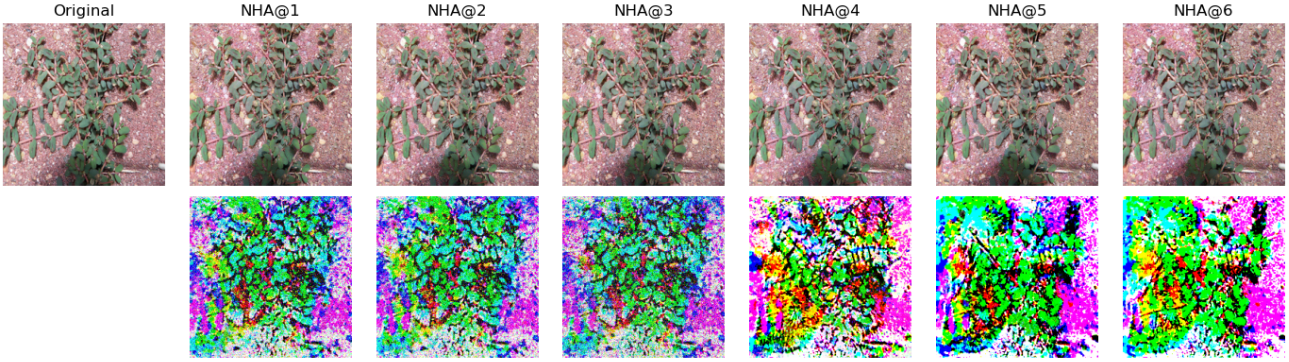



Figure 7: NHA Adversarial examples

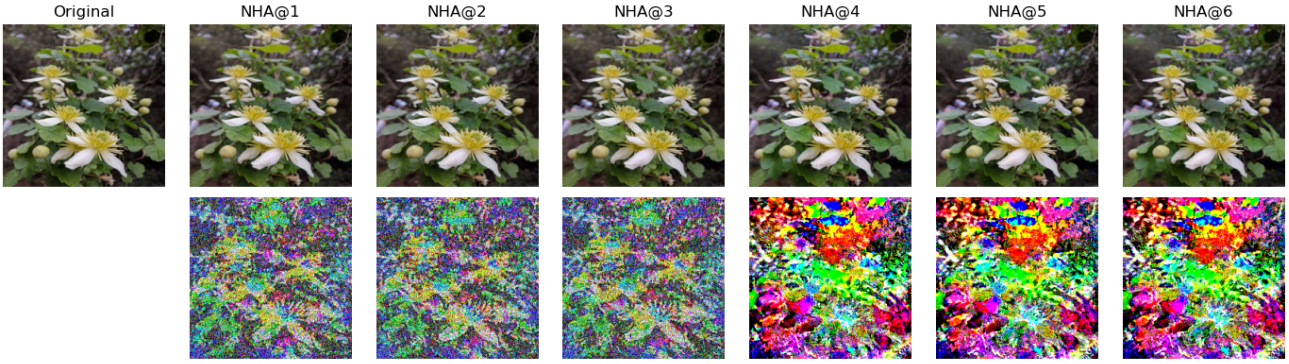



Figure 8: NHA Adversarial examples

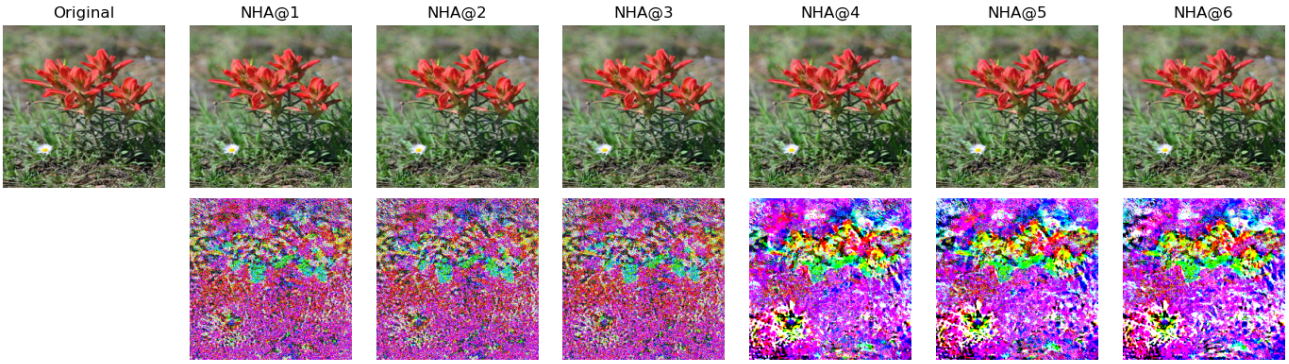



Figure 9: NHA Adversarial examples

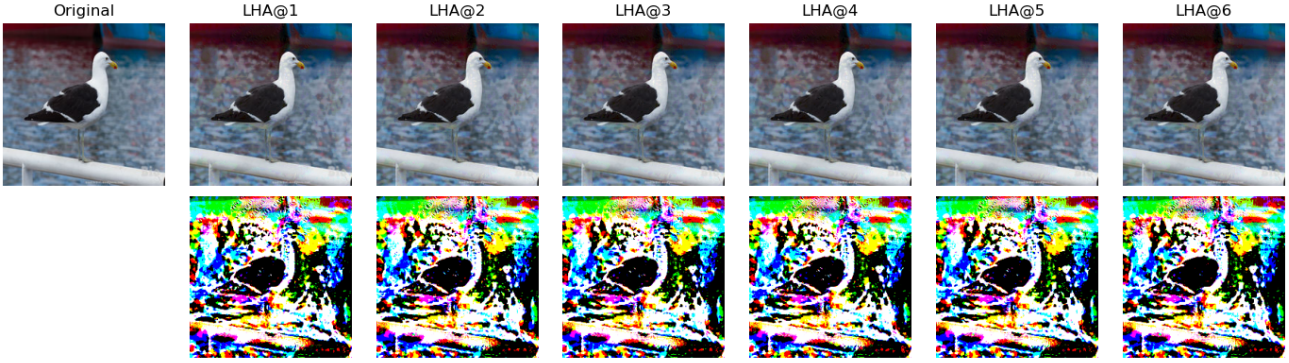



Figure 10: GHA Adversarial examples

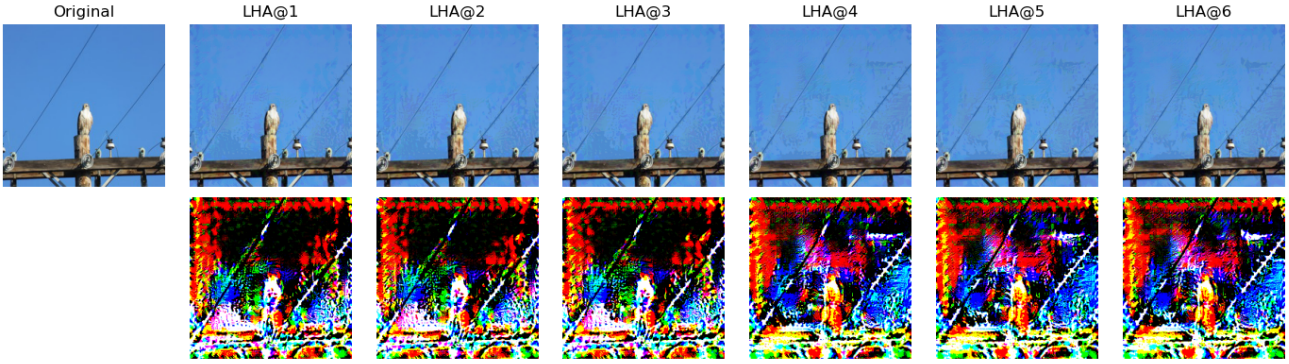



Figure 11: GHA Adversarial examples

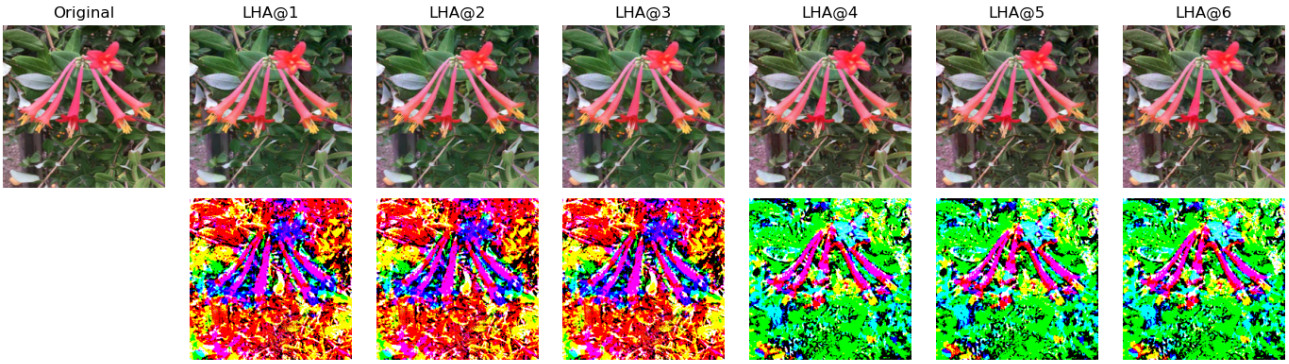



Figure 12: GHA Adversarial examples

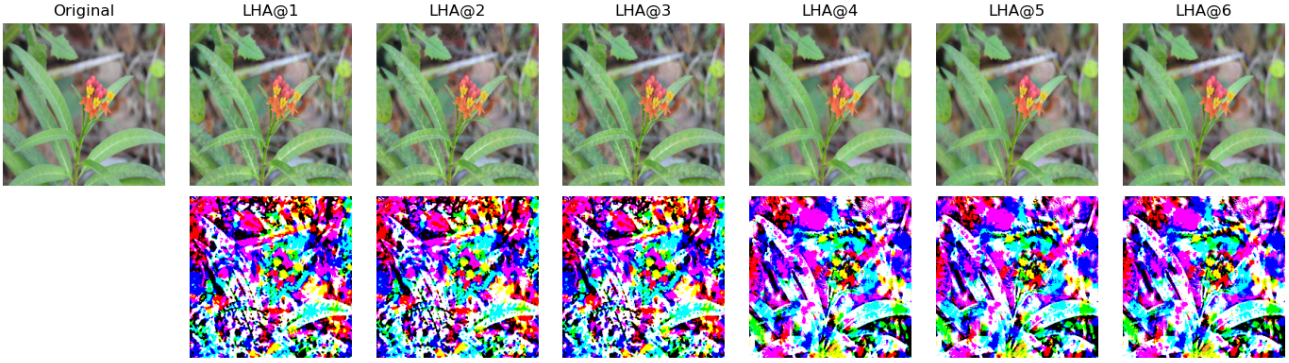



Figure 13: GHA Adversarial examples

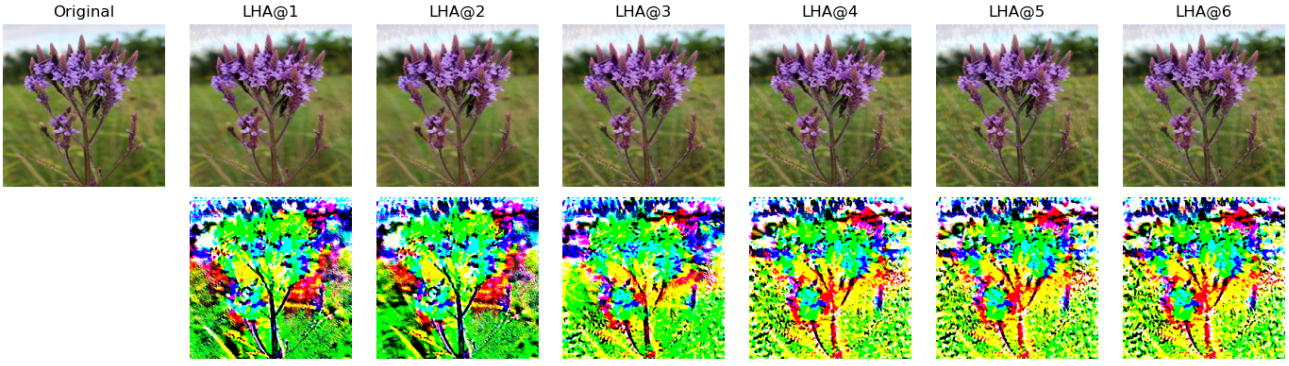



Figure 14: GHA Adversarial examples

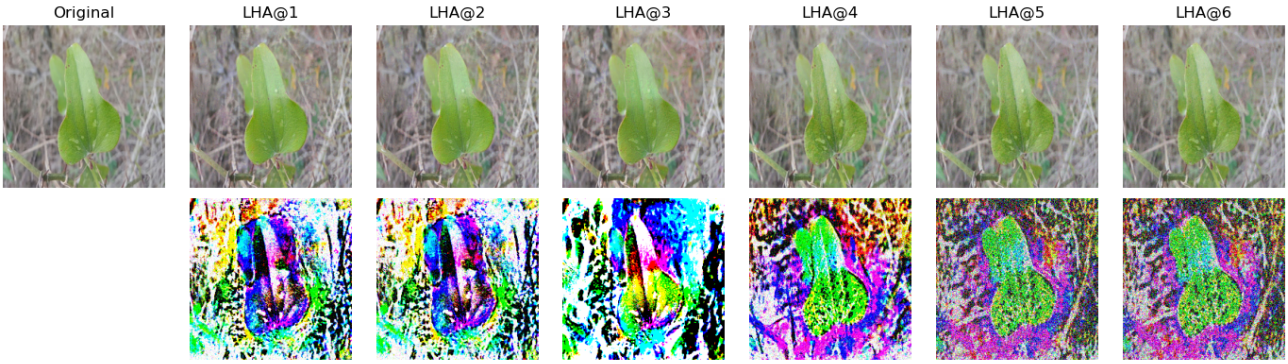



Figure 15: GHA Adversarial examples

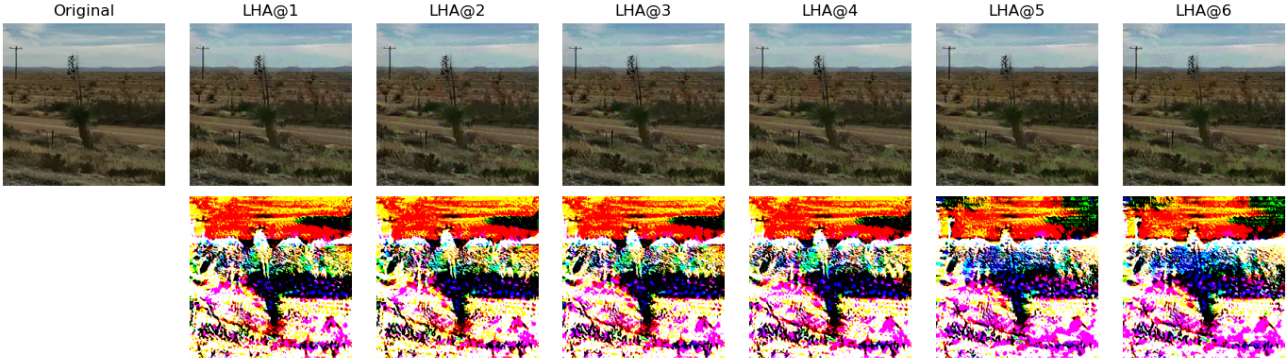



Figure 16: GHA Adversarial examples

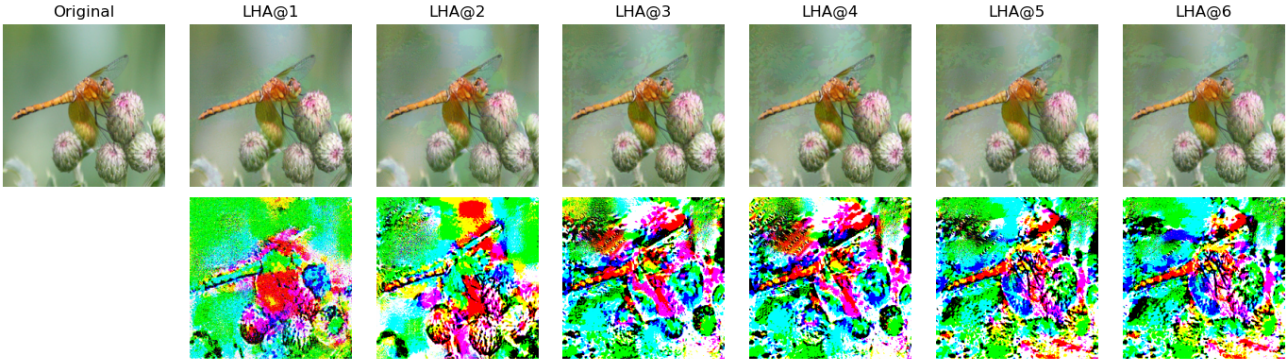



Figure 17: LHA Adversarial examples

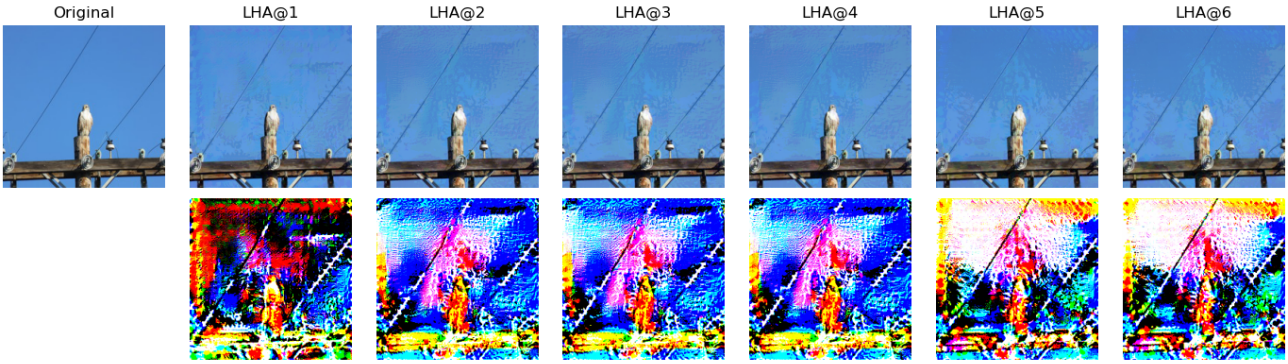



Figure 18: LHA Adversarial examples

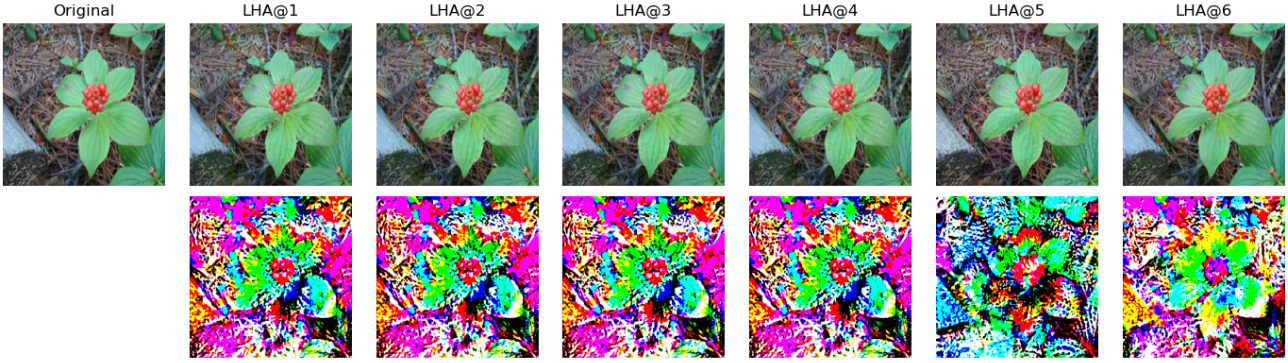



Figure 19: LHA Adversarial examples

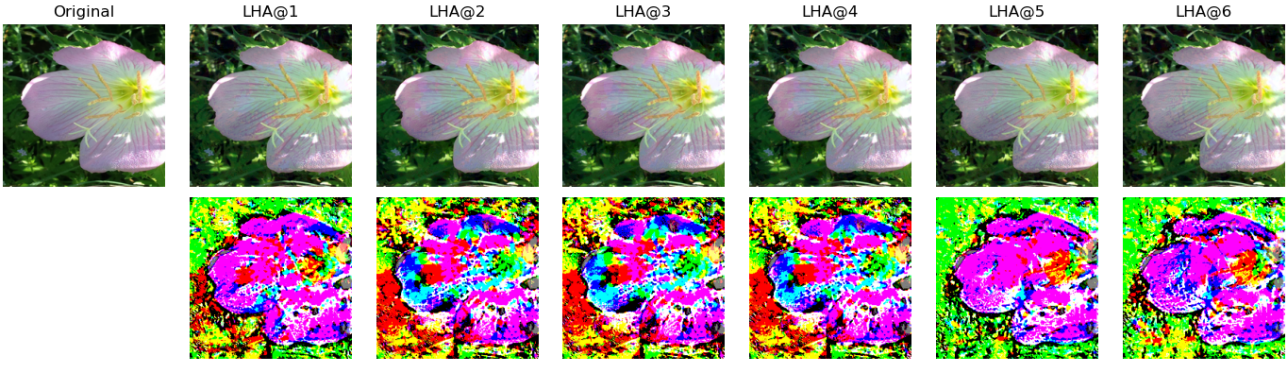



Figure 20: LHA Adversarial examples

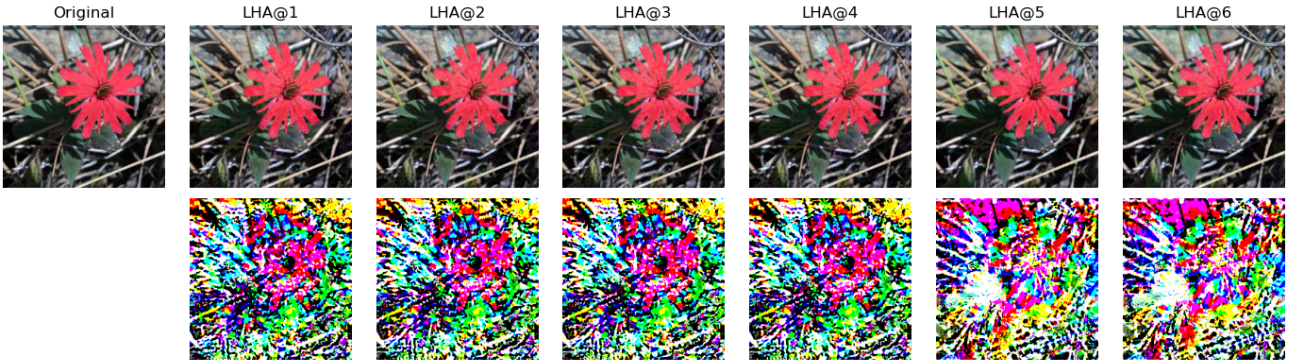



Figure 21: LHA Adversarial examples

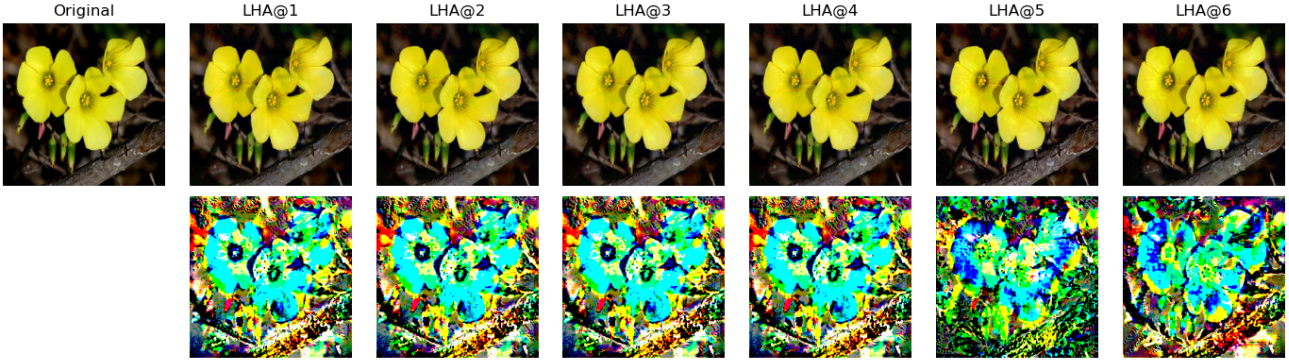



Figure 22: LHA Adversarial examples


Figure 23: LHA Adversarial examples