# A Hierarchical Assessment of Adversarial Severity

Guillaume Jeanneret[1], Juan C. Pérez[2], and Pablo Arbelaez[1]

[1]Center for Research and Formation in Artificial Intelligence, Universidad de los Andes
[2]King Abdullah University of Science and Technology (KAUST)
[1]{g.jeanneret10, pa.arbelaez}@uniandes.edu.co, [2]juan.perezsantamaria@kaust.edu.sa

## Abstract

*Adversarial Robustness is a growing field that evidences the brittleness of neural networks. Although the literature on adversarial robustness is vast, a dimension is missing in these studies: assessing how severe the mistakes are. We call this notion "Adversarial Severity" since it quantifies the downstream impact of adversarial corruptions by computing the semantic error between the misclassification and the proper label. We propose to study the effects of adversarial noise by measuring the Robustness and Severity into a large-scale dataset: iNaturalist-H. Our contributions are: (i) we introduce novel* ***Hierarchical Attacks*** *that harness the rich structured space of labels to create adversarial examples. (ii) These attacks allow us to benchmark the* ***Adversarial Robustness and Severity*** *of classification models. (iii) We enhance the traditional adversarial training with a simple yet effective* ***Hierarchical Curriculum Training*** *to learn these nodes gradually within the hierarchical tree. We perform extensive experiments showing that hierarchical defenses allow deep models to boost the adversarial Robustness by* $1.85\%$ *and reduce the severity of all attacks by* $0.17$*, on average.*

## 1. Introduction

Widely-known adversarial attacks such as FSGM [16], CW [8], Projected Gradient Descent (PGD) [24] or AutoAttack [10], share a common objective: they aim at decreasing accuracy. While these attacks are effective and practical, they ignore the rich semantic structure of the label space. We hypothesize that disregarding these semantic relations amounts to discarding valuable knowledge that states the degree of relation between classes, and we aim at exploring the notion of *severity* of adversarial attacks. The severity quantifies the semantic error of a misclassification induced by an adversary. The hierarchical distance models this error for a label space formed from a hierarchical tree. The adversarial severity extends the traditional top-1 accuracy over a new dimension. Accordingly, to deploy machine learning algorithms in real-life scenarios, we require them to be protected against adversaries that may exploit the structure of the semantic label space. In particular, models need defenses against attacks capable of radically changing their output. Many cases require to cover a complete range of outputs, where some of them have dissimilar connotations. Take as an example action classification for video surveillance. Changing a prediction "playing" to "running" is not severe because of its close semantics. Nonetheless, altering "robbing" to "running" may lead to severe consequences.

This phenomenon is no stranger to DL methods. Since the introduction of DL models, utilizing the hierarchical labels as an additional source of information has been ignored to some extent. Similar to adversarial attacks, these architectures focus exclusively on improving accuracy. Recently, Bertinetto *et al*. [6] studied how modern progress in DL has not translated into improvements in terms of hierarchical error, despite providing significant gains in accuracy.

In this paper, we quantify the semantic errors in which DL models incur when under attack. This study allows us to identify how current attacks and defenses are insufficient to assess the adversarial severity. Thus, we propose: *(i)* a new set of hierarchy-aware attacks. *(ii)* a new benchmark to assess the adversarial robustness and severity. *(iii)* a defense to diminish the severity and the precision of attacks.

We note that the semantic structure of the label space can guide the design of adversarial attacks. Thus, we study hierarchically-labeled datasets and propose a new set of hierarchy-aware attacks. In contrast to traditional adversarial attacks, the target of these hierarchical attacks is input misclassification and large semantic errors. The suite is composed of three novel attacks: *Lower Hierarchical Attack* (LHA), *Greater Hierarchical Attack* (GHA) and *Node-based Hierarchical attack* (NHA). Figure 1 illustrates how traditional attacks ignore the rich structure of the label space

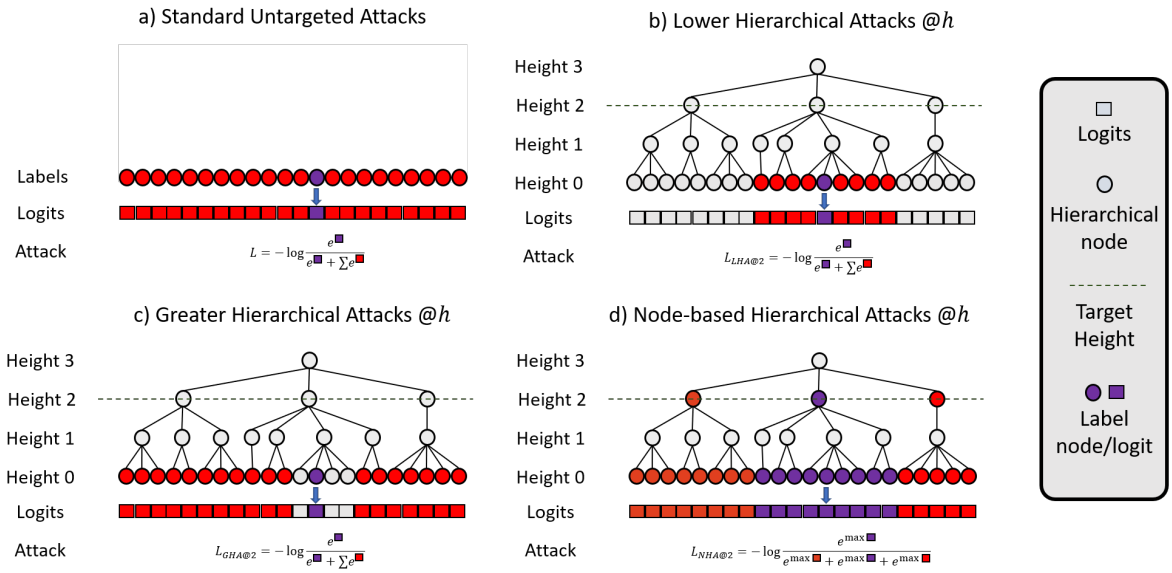



Figure 1: **Hierarchical Attacks.** In contrast to standard adversarial attacks in (a), we generate adversarial perturbations by considering the hierarchical distance between labels: (b) LHA generates the examples by taking into account only the classes with a hierarchical distance lower or equal than $h$. (c) GHA uses those that are higher or equal than $h$. (d) NHA considers the father nodes with a height level of $l$ as the classification nodes to generate the adversaries. We utilize $h = 2$ for the overview of all attacks.

(Figure 1*a*) and how our attacks exploit this information to induce semantic errors (Figures 1*b-d*). Our attacks employ the hierarchical distance to craft adversaries. Firstly, LHA creates distortions with low hierarchical distances. Secondly, GHA attacks the target image to fool the target network with highly unalike semantics. Finally, NHA crafts adversarial images aiming at changing the parent node.

In order to assess adversarial severity, we develop a comprehensive benchmark built upon the work of [6]. We use our proposed attacks to diagnose the brittleness of models and the stress under adversarial attacks. The environment given by iNaturalist-H is an appropriate testbed for analyzing both effects, as its rich label space originates from genetics-based phylogenetic trees. So, we measure the Accuracy and the Average Mistake under the influence of our attacks to quantify the adversarial robustness and severity.

To mitigate adversarial severity, we hypothesize that the inclusion of the hierarchy into adversarial training reduces the severity of adversarial attacks. Thus, our approach takes inspiration from curriculum-based human learning [5, 23]. Intuitively, a student's learning starts from basic and coarse concepts and goes towards finer and specialized ones. This mechanism allows the student to easily learn finer concepts as the coarse notions may be a strong prior. Therefore, we exploit the fact that coarse/fine concepts are naturally defined when an underlying hierarchical structure encompasses the classes. Consequently, we propose a Curriculum for Hierarchical Adversarial Training (CHAT) to gradually learn all nodes at every level in the class hierarchy. Our experiments show that CHAT improves robustness and diminishes the severity against adversaries.

We summarize our contributions as follows:

- We introduce a new set of *Hierarchy-aware Attacks* that optimize adversarial examples by aiming at both diminishing the accuracy and increasing hierarchical errors.
- We provide the *first assessment on adversarial severity* on a highly challenging, large-scale, long-tailed, and hierarchically-structured benchmark: iNaturalist-H.
- We show how employing *CHAT* to gradually learn classes from a tree results in enhanced defenses against hierarchical adversarial attacks.

Our code and models are available at `https://github.com/BCV-Uniandes/AdvSeverity`.

## 2. Related Work

**Robustness Assessments.** As new attacks are released, new defenses are created and vice-versa, generating an arms race. However, several works have demonstrated that reliably assessing adversarial robustness is an elusive task [3]. Thus, recent works have tried to provide either formal certifications of robustness [28, 9] or reliable benchmarks for empirical assessment of adversarial robustness [13, 10].

Despite the sophistication of these benchmarks, we observe that they are only concerned with evaluating whether attacks can induce error in network predictions. Thus, these benchmarks disregard how *severe* such errors are. We fill this gap in the literature by introducing novel hierarchy-aware adversarial attacks. Then, we conduct a large-scale evaluation of the severity of errors induced by such attacks in the challenging *iNaturalist-H* [6, 21] dataset.

**Curriculum in Deep Learning.** A plethora of studies show the benefits of using a learning curriculum during training [33, 5, 22]. Hacohen and Weinshall [18] show that progressive learning from easy to hard instances enhances the performance when compared with the standard training. Graves *et al*. [17] create an automatic curriculum learning to improve convergence time. Weinshall and Amir [37] provide a theoretical perspective on curriculum learning. Duan *et al*. [14] explore a curriculum for 3D-shape representation learning. Wug Oh *et al*. [26] use a curriculum on the task of video object segmentation to learn object appearance over long time frames. We propose an adversarial training curriculum that learns concepts from coarse to fine on the hierarchical tree.

**Hierarchical Classification.** Despite the availability of datasets with hierarchical information [21, 25, 12, 31], the computer vision community has not yet standardized the usage of a rich structured space of labels for either training or evaluation. Recently, Bertinetto *et al.* [6] conducted the first large-scale evaluation of the Accuracy and the Average Mistakes of modern methods by using the taxonomic trees of iNaturalist [21] and tieredImageNet [31]. In [6], Hierarchical-based approaches are split into three groups: Label-embedding methods [4, 15, 32, 20, 38], hierarchical losses [6, 7, 11, 35] and hierarchical architectures [30, 1, 39]. We refer the interested reader to [6] for a thorough review on hierarchical classification. We draw inspiration from hierarchical architectures to introduce a hierarchical curriculum during training to enhance adversarial robustness and reduce the induced semantic error.

# 3. Methods

## 3.1. Notation

A hierarchical tree with height $H$ is a set of groups of nodes $\{Y_0, Y_1, ..., Y_{H-1}\}$ and their corresponding transition functions $C_{h,h'}$. Each set $Y_h = \{1, 2, ..., n_h\}$ contains all nodes at height $h$, *e.g*. $Y_0$ is the set of leaf nodes and $Y_{H-1}$ is the root. Additionally, the transition function $C_{h,h'}$ maps any label in $Y_h$ into a subset of nodes in $Y_{h'}$, representing the label's offspring when $h > h'$. Formally, $C_{h,h'} : Y_h \rightarrow \mathbb{P}(Y_{h'})$, where $\mathbb{P}(\cdot)$ is the powerset of its input. These transition operations have the following properties:

1. $\bigcup_{i \in Y_h} C_{h,h'}(i) = Y_{h'}$
2. $C_{h,h'}(i) \neq \varnothing, \forall i \in Y_h$
3. $C_{h,h'}(i) \cap C_{h,h'}(j) = \varnothing, \forall i, j \in Y_h, i \neq j$ if $h \geq h'$.

Let be a dataset with images and their corresponding hierarchical labels from the tree. Let a neural network be a composition between a backbone $g$ that extract features representation of images and a linear classifier $f_{W,b}$, where $W \in \mathbb{R}^{n_0 \times m}$ and $b \in \mathbb{R}^{n_0}$ are the weights and biases of $f$, respectively. $m$ is the feature dimension of the representation vector, and $n_0$ is the number of leaf classes. Define $W_i \in \mathbb{R}^{1 \times m}$ to be the $i$-th row of $W$ and $b_i \in \mathbb{R}$ to be the $i$-th component of the bias vector.

To measure distances within the topology of a tree, we use the hierarchical distance $d_H$. Hence, the height of the least common ancestor between nodes defines this metric. So, we use $d_H$ to measure a semantic difference between an instance's prediction and the corresponding ground-truth. Throughout the rest of the manuscript, we refer as *mistake* as the hierarchical distance between two nodes.

## 3.2. Hierarchical Attacks

In order to assess the severity of adversarial attacks, we propose novel hierarchy-aware adversaries that perturb images aiming at reducing standard accuracy and increasing the severity of hierarchical mistakes. Figure 1 provides a graphical illustration of our new attacks: *Lower Hierarchical Attack*, *Greater Hierarchical Attack* and *Node-based Hierarchical Attack*. As we intend to increase hierarchical mistakes, all our attacks aim to extract and harm the target image by exploiting the rich structure of the label's space hierarchy.

**Lower and Greater Hierarchical Attacks at height $h$.** These attacks aim at perturbing the target image by creating distortions with a criterion based on the hierarchical distance $h$ between the target leaf label and the other leaf nodes. Both attacks operate similarly: to choose the adversary classes depending on the severity of the attack, given by $h$. Please refer to Figure 1b and 1c for an illustration of the intuition behind our proposed attacks.

On the one hand, the most semantically closed classes to a target label are those whose similarity within the tree is at most $h$ in the hierarchical distance. Accordingly, LHA creates the adversarial examples targeting classes whose distance to the original class is *less* than or equal to $h$. Thus, for an image $x$ and its leaf label $y$, LHA optimizes the loss (Equation (1)):

$$L_{LHA@h}(x, y) = - \log \frac{e^{W_y g(x) + b_y}}{\sum_{j \in Y_0 \text{ s.t. } d_H(y,j) \leq h} e^{W_j g(x) + b_j}}. \quad (1)$$

On the other hand, GHA@$h$ aims at creating adversaries that induce large mistakes (*i.e.* cause remarkable "confuse"

of the network). Conversely to LHA, GHA uses information from the classes with a hierarchical distance *greater* or equal to $h$. Thus, GHA maximizes the loss (Equation (2)):

$$L_{GHA@h}(x, y) = -\log \frac{e^{W_y g(x)+b_y}}{\sum_{j\in\{k\in Y_0 | d_H(y,k)\geq h \text{ or } k=y\}} e^{W_j g(x)+b_j}}. \quad (2)$$

To ease the understanding of these losses, we point out that the objective to minimize is similar for both attacks. On Figure 1b and 1c, in essence, both attacks harms the purple node -the label- by using as negative classes the red leaves.

**Node-based Hierarchical Attacks.** The NHA harms the nodes directly with a height of $h$. Figure 1d exemplifies our proposed attack. To compute the probability of classifying a target image $x$ as a node $y$ with height $h$, we follow the conditional probability theory. Thus, this probability equals the sum of the probabilities of choosing any leaf class derived from $y$. Therefore,

$$P(c(x) = y \in Y_h) = \sum_{i\in C_{h,0}(y)} P(c(x) = i \in Y_L) \quad (3)$$

where $c(x) = y$ is the action of classifying $x$ as $y$ and $P(c(x) = y \in Y_0)$ is the probability of classifying $x$ as $y \in Y_0$. As is standard, all state-of-the-art models use a Softmax layer to compute the probability distribution of an input image by using the corresponding logits. Therefore, the probability of the neural network to classify $x$ as class $i$ is:

$$P(c(x) = i \in Y_0 | W, b) = \frac{e^{W_i g(x)+b_i}}{\sum_{j=1}^{n_0} e^{W_j g(x)+b_j}}. \quad (4)$$

Hence, following Equation (3), the probability of classifying $x$ as $y \in Y_h$ is:

$$P(c(x) = y \in Y_h | W, b) = \frac{\sum_{i\in C_{h,0}(y)} e^{W_i g(x)+b_i}}{\sum_{j=1}^{n_0} e^{W_j g(x)+b_j}}. \quad (5)$$

To extract adversaries based on Equation (5), we compute the cross-entropy loss. To speed up this operation, we estimate the adversaries by attacking the cross-entropy on Equation (6). This function computes the loss over the maximum logits within nodes of interest:

$$\tilde{L}_{NHA@h}(x, y) = -\log \frac{e^{\tilde{L}_y}}{\sum_{j=1}^{n_h} e^{\tilde{L}_j}} \quad (6)$$

where, $\tilde{L}_y = \max_{i\in C_{h,0}(y)} W_i g(x) + b_i$. We base our design on the mathematical principle that, if $L_y$ is the logit corresponding to node $y \in Y_h$, with straightforward algebraic manipulation we can obtain that

$$L_y = \log \left[ \sum_{i\in C_{h,0}(y)} e^{W_i g(x)+b_i} \right]. \quad (7)$$

It is widely known that the $\max$ function has a variety of differentiable approximations. We recall one in particular: $max(x_1, ..., x_m) \approx \log(e^{x_1} + ... + e^{x_m})$. Thus, we find that $L_y \approx \tilde{L}_y$. As a result, we attack an approximation of the true cross-entropy function–Equation (6). In Figure 1d, we exemplify the idea behind NHA. This attack extracts the maximum logit over the leaf offspring of each node at the target height. So, NHA harness these logits to create the adversary.

### 3.3. Benchmarking Adversarial Severity

In order to conduct an assessment of both adversarial robustness and severity, we compute two metrics that reflect these concepts. First, we compute the standard *Robust Accuracy*. This measurement quantifies the worst-case endurance of a defense against adversarial examples. However, this metric does not provide information about the semantic error the model incurred under the attack. Hence, we also compute the *Average Mistake* [6]. So, we average the hierarchical distance $d_H$ of all misclassified instances. This metric quantifies the semantic dissimilarity between the prediction and the ground-truth label, thus, addressing the issue presented by only measuring accuracy.

Fundamentally, our attacks propose novel optimization objectives. In practice, for optimizing such objectives, we implement a PGD-based strategy. The standard PGD [24] iteratively performs FSGM steps [16] to maximize the Cross-Entropy Loss $L$ between the prediction for an instance $x$ and its corresponding ground-truth label $y$ to find an adversarial example:

$$x_{t+1} = \prod_{B_{\ell_\infty}(x,\epsilon)} x_t + \alpha \operatorname{sign}\left(\nabla_{x_t}(L(m(x_t), y))\right), \quad (8)$$

where $m(\cdot) = Softmax(f_{W,b}(g(\cdot)))$, and the initial point $x_0$ is perturbed with noise, namely $x_0 = x + u$, with $u$ being sampled from the uniform random distribution $\mathcal{U}[-\epsilon, \epsilon]$, $\alpha$ is the step per iteration and $\prod_{B_{\ell_\infty}(x,\epsilon)}$ is the projection function over the set $B_{\ell_\infty}(x, \epsilon)$, namely, the intersection set between the $\epsilon$-ball with the $\ell_\infty$ norm around $x$ and the set $[0, 1]^{size(x)}$. We refer to an $n$-step PGD attack as PGD$n$. Using PGD-optimization for our attacks amounts to replacing the Loss $L$ in Equation (8) with each of the losses in Equations (1), (2) and (6). Analogously, we refer to the $n$-step versions of these attacks as LHA$n$@$h$, GHA$n$@$h$ and NHA$n$@$h$. For all our evaluation experiments, we set $\alpha = 1/255$ and report the worst-case accuracy. Furthermore, we compute the adversaries for correctly-classified instances.

**Dataset:** We construct our methodology on top of the *iNaturalist-H*, created by [6]. This dataset is a partition of the challenging iNaturalist 19 [21]. It is known by how its class distribution follows a long-tail, resembling the imbalance of the real world. This dataset covers a total of

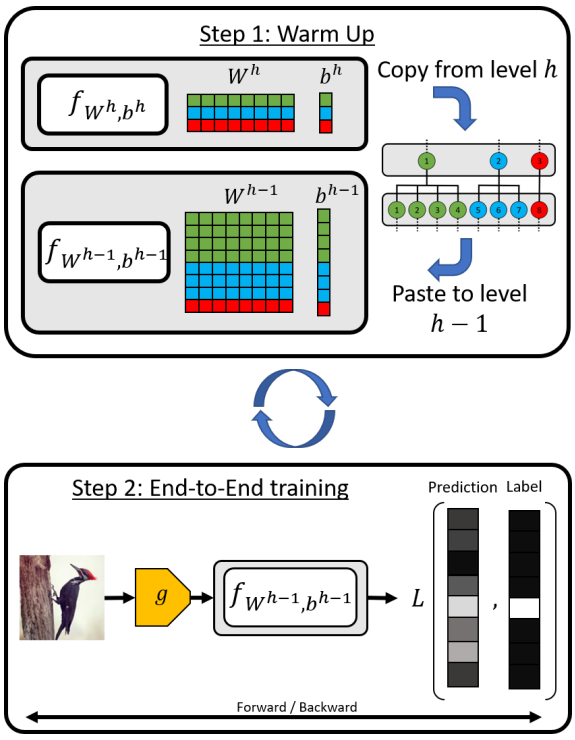



Figure 2: **Overview of CHAT.** Our hierarchical curriculum has two steps: the warm up (upper Figure) and the end-to-end training (bottom Figure). The warm up consist on transferring the weights and biases from the parent node to its offspring. The second step consist on training the model's parameters, $W^{h-1}$ and $b^{h-1}$ in a end-to-end manner.

1010 leaf classes dominated by an 8-level phylogenetic tree. We perform all our ablation experiments on the validation set and report the main results in Figure 3 on the test set. We provide a detailed description of this dataset and useful statistics on the **Supplemental Material**.

### 3.4. Curriculum for Hierarchical Adversarial Training

In order to defend our model against accurate and severe attacks, we propose CHAT: Curriculum for Hierarchical Adversarial Training. Inspired by human learning, we create a curriculum-based training to learn all the nodes by progressively increasing the difficulty of the task, *i.e.* iteratively deepening the current height $h$ of tree to $h-1$ at each stage, until reaching $h=0$. Our curriculum comprises two iterative steps: the warm up and the end-to-end training. Figure 2 illustrates a stage on our curriculum training. For the first step, consider $W^h$ and $b^h$ to be the weights and biases of the current classifier at the $h$-th stage of the curriculum. When updating the step from $h$ to $h-1$, the size of the weights of $f$ must change to account for the increase in the number of classes. Thus, we initialize a new classifier $f_{W^{h-1},b^{h-1}}$, with $W^{h-1} \in \mathbb{R}^{n_{h-1} \times m}$ and $b^{h-1} \in \mathbb{R}^{n_{h-1}}$.

| $\epsilon$ | $m$ | $\alpha$ | C | Clean Acc | Clean AM | PGD50 Acc | PGD50 AM |
|---|---|---|---|---|---|---|---|
| 4 | 6 | 6 | | 31.40 | 3.21 | 12.33 | 3.20 |
| 4 | 8 | 6 | ✓ | 32.84 | 3.04 | 13.36 | 3.06 |
| 6 | 6 | 4 | | 24.87 | 3.44 | 7.13 | 3.44 |
| 6 | 8 | 4 | ✓ | 27.19 | 3.28 | 8.32 | 3.29 |
| 8 | 6 | 6 | | 19.65 | 3.76 | 4.31 | 3.71 |
| 8 | 8 | 6 | ✓ | 23.29 | 3.49 | 6.07 | 3.49 |

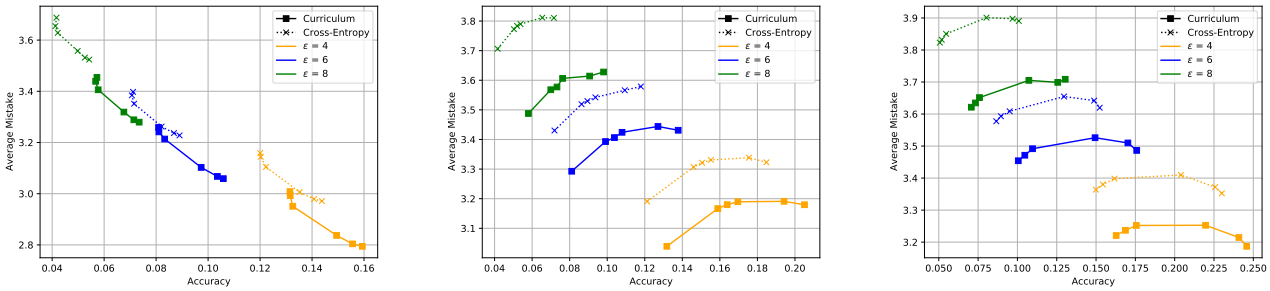
Table 1: **Clean and PGD50 performaces for the best models.** We compare the best models we extracted through the grid search on the space of hyperparameters. C stands for curriculum, $m$ are the number of iterations per images and $\alpha$ the step size. The results show that our proposed curriculum greatly enhances the performance on all metrics by using FAT.

Then, we provide a warm up for the weights $W^{h-1}$ and bias $b^{h-1}$ by transferring the parameters from each node $j$ to all its children. Finally, we discard the weights $W^h$ and biases $b^h$. Formally, for all $i \in C_{h,h-1}(j)$ we apply:

$$\begin{aligned} W_i^{h-1} &= W_j^h \\ b_i^{h-1} &= b_j^h \end{aligned} \tag{9}$$

Due to properties 1, 2 and 3 of the transition operations in section 3.1, we ensure *(i)* that there is always at least an offspring at each step, *(ii)* that there are no child nodes that descend from two parents, and *(iii)* we cover all nodes at height $h-1$. The second step consists of training all parameters of both the feature extractor and the linear classifier, $W^{h-1}$ and $b^{h-1}$, in an end-to-end manner. The optimization process uses the labels from the current stage, namely $Y_{h-1}$, to minimize the cross-entropy loss. Furthermore, we robustify the model on this step by replacing the training with any adversarial learning method [36, 34, 24]. We iterate these stages until convergence at the leaf nodes. We enforce the number of epochs at each stage is closely related to the number of nodes at each level.

To train our models to be robust, we adopt an Adversarial Training [24] inspired strategy. In particular, we follow the computationally-efficient *Free Adversarial Training* (FAT) technique [34], which exploits each forward-backward pass to optimize both model parameters and the adversarial examples used for training. FAT has three hyper-parameters of interest: the number of times each adversarial example is replayed, $m$, the bound for the adversarial examples, $\epsilon$, and the adversary optimization's steps size, $\alpha$.




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

Figure 3: **Hierarchical Attack Evaluation.** We evaluate the performance of FAT on our hierarchical attacks with 50 iterations at each target height on the test set. The inclusion of our curriculum boost all metrics for all levels compared to the baseline. (a) LHA is the strongest accuracy-related attack among the proposed ones. It even surpasses the PGD when setting the height $h$ to 5 or 6. (b) The GHA enjoys a balanced between severity and accuracy. (c) The NHA is the most severe but less successful attack among the set of hierarchical attacks.

# 4. Experiments

## 4.1. Curriculum-Enhanced Models against Hierarchical Attacks

**Implementation Details.** For all our experiments we follow the experimental setup of [6]. We initialize a ResNet-18 [19] from ImageNet-pretrained [12] weights and then train the network for $200,000$ steps. We use the Adam optimizer [29] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, learning rate of $10^{-5}$, no weight decay, and a mini batch-size of 256. We fix the updates in curriculum stages to occur at $2\%$, $4\%$, $6\%$, $15\%$, $25\%$ and $35\%$ of the training iterations.

**Model Selection.** We assess the effects of our new attacks against our CHAT-enhanced models. We selected the models that provided the best robustness against PGD50 on the validation set for a fair comparison between the vanilla and CHAT-enhanced models. We explore the space of parameters $m \in \{2, 4, 6, 8\}$ and $\alpha \in \{1/255, 2/255, 4/255, 6/255\}$ for the standard and hierarchy-enhanced models to search for the best-performing models for all $\epsilon \in \{4/255, 6/255, 8/255\}$. We report each model's hyperparameters and performance on the validation set in Table 1. The results show that the proposed CHAT enhances *all* metrics on the clean and PGD50 settings by a large margin. We underscore that all curriculum-enhanced models were trained with a higher number of iterations $m$. We suspect that a higher number of iterations per height will increase even further the gap. Nonetheless, a complete study on the curriculum aspect of the training is out of the scope of this paper. Also, the step sizes of the models are large in comparison to their perturbation budget. Wang and Zhang [36] evidence that some models enjoy similar robustness for larger step sizes and fewer iterations when trained adversarially on the CIFAR dataset. Although we are tackling a higher-dimensional dataset, we observe this phenomenon too. FAT approximates a 1-step PGD training routine. Thus, large step sizes may further enhance adversarial robustness.

| Metric | PGD5 | PGD10 | PGD50 | PGD100 |
|---|---|---|---|---|
| Accuracy | 13.85 | 11.44 | 11.07 | 11.06 |
| AM | 3.25 | 3.25 | 3.25 | 3.25 |

Table 2: **PGD iterations.** We assess the number of iterations under the same PGD attack for a vanilla ResNet-18 trained with Cross-Entropy. This experiment shows that PGD50 explores close results as the PGD100.

Once we choose all the best models, we quantify the effectiveness of our attacks on the testing set. Figure 3 report the results for LHA50@$h$, GHA50@$h$ and NHA50@$h$ in the test set. These results demonstrate that introducing CHAT improves both the adversarial accuracy and severity metrics for *all* models. The results of LHA present an average increase of $1.45\%$ for the robust accuracy and an average mistake decrease of $0.18$. The GHA and NHA present an average boost on the robust accuracy of $1.72\%$ and $1.91\%$, respectively. Furthermore, The average mistake metric decreases by $0.16$ for GHA and NHA. We attribute these gains to the curriculum's semantic partition, which demonstrates that semantically-coherent representation spaces enhance model robustness.

We further observe that the average mistake saturates at heights greater than or equal to 4. In contrast, the accuracy does not reach such a plateau. This event seems contradictory to the result of Figure 4. Nonetheless, the average mistake metric on this table considers both instances that initially were erroneous and not perturbed and those whose attack was successful.

| Method | Clean | | PGD50 | |
|---|---|---|---|---|
| | Acc | AM | Acc | AM |
| Standard | 29.64 | 3.25 | 11.07 | 3.25 |
| Scratch | *34.02* | *3.04* | *11.90* | *3.05* |
| CHAT (Ours) | **34.11** | **2.98** | **12.15** | **3.01** |

Table 3: **Effects of CHAT on Adversarial Training.** We report the results of Clean and PGD50 attack accuracy (Acc) and Average Mistake (AM). Results in **bold** and *italic* show the best and second best performances, respectively.

| Curriculum | Clean | | PGD50 | |
|---|---|---|---|---|
| | Acc | AM | Acc | AM |
| Linear | 26.39 | 2.97 | 10.52 | 3.04 |
| Ours | 34.11 | 2.98 | 12.15 | 3.01 |
| Change | **+7.72** | *+0.01* | **+1.63** | **-0.03** |

Table 4: **Effect of the Curriculum Steps.** We test two different curriculums and report both accuracy (Acc) and Average Mistake (AM) for clean and PGD50 adversaries. We test an exponential-like curriculum, labelled as ours, and a linear one. Given the exponential growth of the number of nodes at each level of the hierarchy tree, an exponential-like curriculum fits perfectly. **Bold** changes show gains and in *italic* losses.

### 4.2. Ablation Studies

This section studies *(i)* the proposed defense mechanism's components and *(ii)* the effects of the hierarchical-aware attacks. We perform all ablation experiments in the validation set. From the side of the attacks, we first explore the effect of the number of iterations for PGD and report our results in Table 2. From these results, we conclude that having $50$ iterations is reasonable for a clear evaluation. Thus, for all evaluations of adversarial robustness, we report numbers with respect to PGD50. We study two factors of interest from the training side: the curriculum and the number of iterations between curriculum stages. We train the ResNet-18 with a perturbation budget of $\epsilon = 4$, a step size of $\alpha = 2$ and $m = 8$ iterations to evaluate all our ablations.

**Weight Transfer.** At each stage of the curriculum, the size of the linear classifier $f$ increases, implying that new weights are required. Here we assess the impact of employing the warm-up stage in contrast to *(i)* employing no curriculum (Standard) and *(ii)* a naive initialization from scratch at each curriculum stage (Scratch). We report the results of this experiment in Table 3. Our results show that the sole inclusion of hierarchical information into FAT ("Scratch" *vs.* "Standard" in Table 3) greatly improves accuracy and diminishes the error severity on both clean and PGD50 settings. Moreover, employing our warm-up strategy when changing between hierarchical levels boosts performance metrics ("CHAT (Ours)" *vs.* "Scratch" in Table 3). Since the Cross-Entropy ignores the relationship between labels, semantically dissimilar classes may be adjacent in the representation space. Although previous works [7] show that Cross-Entropy is capable of learning class hierarchies, our experiments suggest that employing a curriculum encourages semantically-similar classes to lie closer in representation space. We argue that this may be because they have a disjoint hyperspace for coarse classes as a prior, learned previously in the curriculum stages.

**Step Spacing.** The number of iterations used during parameter optimization can have large consequences on final performance. Since a curriculum on a hierarchy implies a changing number of classes at each stage, the number of optimization iterations run at each stage becomes an influential factor in performance. Thus, we assess the effect of the amount of iterations to optimize each curriculum stage. In particular, we test two spacing strategies between curriculum stages: a naive linear spacing and our proposal, an exponential-like spacing. We report our results on Table 4. These results suggest that providing an exponentially-increasing number of iterations for optimizing the classes at each height of the tree leads to higher-performing *and* more robust models. Combining these results with the evidence from Table 3, we argue that inducing hierarchy-aware priors is key to training more accurate and robust models. Using a linear spacing, we enforce this phenomenon: the small difference on Average Mistake values shows a semantically disjoint space for the parent nodes. Nonetheless, the amount iterations available on the last stage does not enable the convergence into a local minimum for the leaf nodes.

On the **Supplemental Material** we present further experimentation with TRADES [40] and TRADES enhanced with CHAT.

**Hierarchical Attacks.** Recall that our proposed hierarchy-aware attacks have different objectives in mind, all related to inducing mistakes of varying severities. In order to visualize each effect, we plot on Figure 4 the trade-off between *(i)* the Accuracy Drop ("Clean Accuracy" minus "Robust Accuracy") against *(ii)* the average mistake of those instances whose the attack was successful ("Flipped Average Mistake"). The former dimension looks at how powerful an attack is, and the latter reviews the severity of the successful attack. The results show the intended effects of each attack: The LHA and GHA produce inverse effects, the former being semantically similar and stronger attacks, and the latter more severe but less effective attacks. Note that GHA@1 is equal to standard PGD. We expected the standard PGD to outperform all other attacks in the accuracy metric because it has a broader range of information. To our surprise, both LHA@6 and LHA@5 achieve slightly

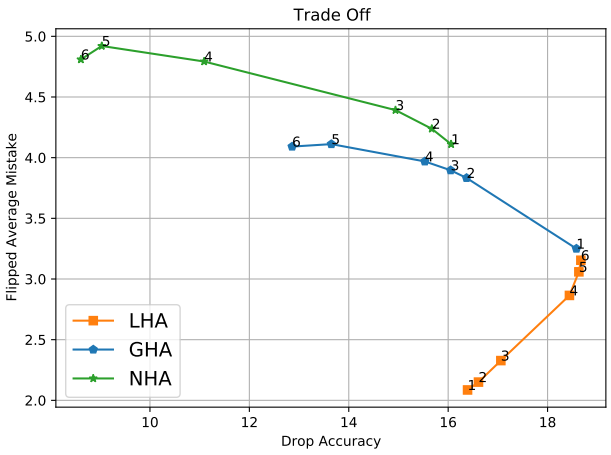



Figure 4: **Hierarchical Attack effects.** We plot the Difference between Clean and Robust Accuracy ("Accuracy Drop") against the flipped Average Mistake. The LHA generates semantically similar variances to each instance, while GHA creates dissimilar noise. The NHA creates weak but severe attacks as the curve is below the other ones. The number on top of each point of the curve represents the height of the hierarchical attack.

larger accuracy drops of $0.09$ and $0.05$ points, respectively. We attribute this result to the existence of useless gradients that hinder the effectiveness of gradient-based attacks, as observed by previous studies [2, 27]. Finally, the most severe attack, in terms of semantic mistakes, is our newly proposed NHA. On average, we find average mistakes of at least 0.86 and up to 1.67 points compared to the standard PGD50. Note that the proposed attacks are not as effective at degrading accuracy compared to the standard PGD. Since some images may contain semantic regions similar to some classes, the PGD attack covers this spectrum of ranges. In contrast, our proposed attacks have a reduced sight of these classes. On the **Supplemental Material** we present further experimentation mixing AutoAttack [10] with NHA.

### 4.3. Qualitative Results

We visualize the effects of the proposed attacks and PGD adversaries on Figure 5. We used our CHAT model to compute the adversaries. We set the height $h$ to 3, and $\epsilon = 8/255$. In addition, we visualize their corresponding perturbation for all adversarial examples. In the first set of attacks, we notice that PGD and LHA behave similarly, while the GHA and NHA noises followed different directions, as noted by the color of the noise. These results exemplify that the PGD attack and the LHA use the same information to create the adversaries for this instance, while the NHA and GHA used different gradients to reach their local minimum. Contrastively, the PGD and LHA noises behave differently in the second set of adversarial examples; the color of both

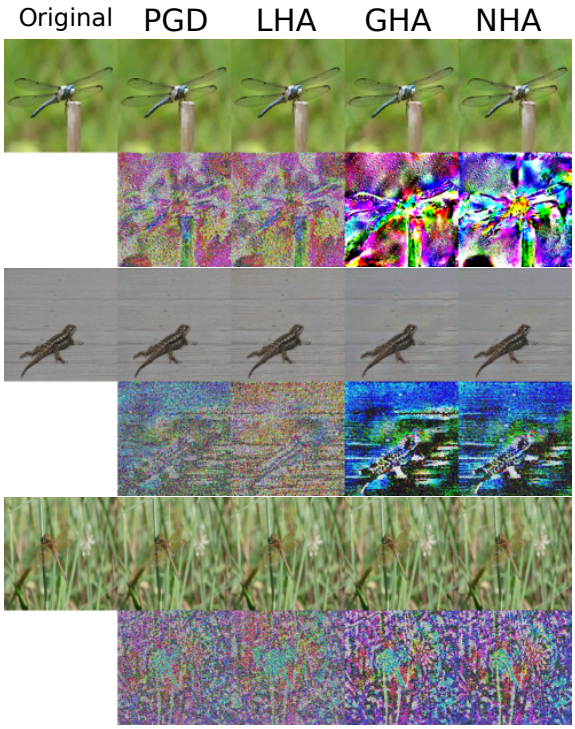



Figure 5: **Adversarial Examples.** We visualize some adversarial examples with $\epsilon = 8/255$. From left to right, we display the original image, PGD, LHA@3, GHA@3 and NHA@3. Each image bellow the adversarial example is the corresponding adversarial noise. All proposed attacks explore the semantics within the image in different manners.

noises differs. Nonetheless, the GHA and NHA create similar perturbations. Finally, the last instance shows that all attacks may similarly create adversaries. We set further visualizations of multiple adversarial examples under different depths on the **Supplemental Material**.

## 5. Conclusions

In this paper, we unravel the rich semantic structure of the label space to devise a new set of hierarchical attacks. To assess their effects, we extend the classical evaluation metric of the adversarial accuracy and explore a new dimension of adversarial attacks: their severity. Consequently, we exploit iNaturalist-H, a large-scale dataset with a label space generated from a taxonomic tree, and create a benchmark with the aforementioned metrics. Furthermore, we propose CHAT, a curriculum-enhanced training to improve the robustness against adversarial examples and the severity of the damage by using all the hierarchical nodes of the taxonomic tree. We hope that studying adversarial severity opens new research directions in robustness.

## References


[1] Karim Ahmed, Mohammad Haris Baig, and Lorenzo Torresani. Network of experts for large-scale image categorization. 2016.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

[4] B. Barz and J. Denzler. Hierarchy-based image embeddings for semantic image retrieval. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 638–647, 2019.

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In ICML 2009.

[6] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A. Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] A. Bilal, A. Jourabloo, M. Ye, X. Liu, and L. Ren. Do convolutional neural networks learn class hierarchy? *IEEE Transactions on Visualization and Computer Graphics*, 24(1):152–162, 2018.

[8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.

[9] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[10] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, 2020.

[11] Jia Deng, Alexander C. Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *Computer Vision – ECCV 2010*, pages 71–84, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[13] Y. Dong, Q. A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu. Benchmarking adversarial robustness on image classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 318–328, 2020.

[14] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J. Guibas. Curriculum deepsdf. In *European Conference on Computer Vision (ECCV)*, pages 51–67, Cham, 2020. Springer International Publishing.

[15] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26. Curran Associates, Inc., 2013.

[16] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[17] Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 1311–1320, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[18] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR, 09–15 Jun 2019.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[20] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.

[21] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[22] Faisal Khan, Bilge Mutlu, and Jerry Zhu. How do humans teach: On curriculum learning and teaching dimension. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[23] Yong Jae Lee and Kristen Grauman. Learning the easy things first: Self-paced visual category discovery. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representation (ICLR)*, 2017.

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[26] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *International Conference on Computer Vision (ICCV)*, 2019.

[27] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, and Ananthram Swami Z. Berkay Celik. Practical black-box attacks against machine learning. In *Asia Conference on Computer and Communications Security (ASIA CSS)*, 2017.

[28] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

[29] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018.

[30] Jopseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] Mengye Ren, Sachin Ravi, Eleni Triantafillou, Jake Snell, Kevin Swersky, Josh B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018.

[32] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2152–2161. PMLR, 07–09 Jul 2015.

[33] T. D. Sanger. Neural network learning control of robot manipulators using gradually increasing task difficulty. *IEEE Transactions on Robotics and Automation*, 10(3):323–333, 1994.

[34] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32. Curran Associates, Inc., 2019.

[35] N. Verma, D. Mahajan, S. Sellamanickam, and V. Nair. Learning hierarchical similarity metrics. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2280–2287, 2012.

[36] Jianyu Wang and Haichao Zhang. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[37] Daphna Weinshall and Dan Amir. Theory of curriculum learning, with convex loss functions. *Journal of Machine Learning Research (JMLR)*, 21(222):1–19, 2020.

[38] Yongqin Xian, Gaurav Sharma Zeynep Akata, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[39] Zhicheng Yan, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Robinson Piramuthu. HD-CNN: hierarchical deep convolutional neural network for image classification. 2015.

[40] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.