NO FEATURE IS AN ISLAND: ADAPTIVE COLLABORATIONS BETWEEN FEATURES IMPROVE ADVERSARIAL ROBUSTNESS

Anonymous authors
Paper under double-blind review

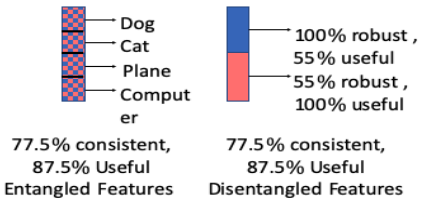
ABSTRACT

To classify images, neural networks extract features from raw inputs and then sum them up with fixed weights via the fully connected layer. However, the weights are fixed despite the input types. Such fixed prior limits networks' flexibility in adjusting feature reliance, which in turn enables attackers to flip networks' predictions by corrupting the most brittle features whose value would change drastically by minor perturbations. Inspired by the analysis, we replace the original fixed fully connected layer by dynamically calculating the posterior weight for each feature according to the input and connections between them. Also, a counterfactual baseline is integrated to precisely characterize the credit of each feature's contribution to the robustness and generality of the model. We empirically demonstrate that the proposed algorithm improves both standard and robust error against several strong attacks across various major benchmarks. Finally, we theoretically prove the minimal structure requirement for our framework to improve adversarial robustness in a fairly simple and natural setting.

1 INTRODUCTION

The vulnerability of deep neural networks to adversarial examples has recently drawn significant attention. Previous works has explored the phenomenon from a wide range of aspects, such as empirically or theoretically effective defense methods against adversarial examples Zhang et al. (2019), potential causes of adversarial examples Bubeck et al. (2019); Ilyas et al. (2019), trade-off between robustness and generality for robust models Raghunathan et al. (2020); Carmon et al. (2019); Zhang et al. (2019). Despite the significant advances made in many aspects of robust machine learning, the representations achieved by robust models are not interpretable. We cannot predict the outcome of manipulations on them. Without a clear understanding and control of the representations' inner structure, it wouldn't be possible to reliably avoid unpredictable model behaviors. For example, in a scenario that sometimes requires high robustness but also sometimes requires high generality, even the best existing robust models would fail. Current robust models can ensure overall generality and robustness, but we wouldn't know how the models balance the two for each sample, let alone control and optimize model behavior according to the input and our desires. Since these neural representations can be decomposed into a set of features, a natural starting point of controlling model behavior is investigating these atomic features. How are features, or certain dimensions of neural representations, influence model properties like robustness or generality? How can we control them since a single feature is more interpretable and controllable compared to the whole representation?

As a starting point, we examine the representations of TRADES Zhang et al. (2019), which achieves the best standard accuracy on robust accuracy on CIFAR10, by analyzing representations of samples misclassified by it. The loss function of TRADES contains two parts where one part maximizes standard accuracy and the other maximizes adversarial accuracy. Ideally, representations achieved from this formulation are expected to have both useful and robust features that take care of standard and adversarial accuracy. However, there's a trade-off between the two targets. Namely, features cannot have 100% robustness and 100% standard accuracy. Instead, the solution would sacrifice some performance of one target for the other. The situation can be briefly viewed as the sum of standard accuracy and robust accuracy are fixed, which makes the improvement of one part decreases the other. So there have to be features that are consistent after perturbation, as well as features that



	dog	plane
top 5	1	4
top 10	5	7
top 20	9	14
top 30	11	23
top 40	14	31
top 50	19	37
top 50	22	46

Figure 1: Left: Possible representation structures under a loss optimizes both standard and adversarial accuracy. The representations achieved from previous works usually look like the purple vector, whose features are entangled together, neither very robust nor very useful.
Right: Each entry is the number of top features shared between the representations achieved after the dog class received target attacks at the plane class, and those of representations achieved from standard input of the column class.

are not robust but useful for generalization. As shown in Figure 1(Left), there are two cases for the learned feature. (1)Useful features and consistent features are tangled together, both of moderate performance, (2) features are of high robust scores but low useful score, (3) features are of low robust scores but high useful score.

As a case study, we compare the feature scores of dogs before and after a targeted attack at the plane class. Note that we only look into the misclassified samples. The whole representation is a 640 dimension vector. For a normal dog image, the dimension receives the highest score are 547, 361, 543, 388, 446. For the plane class, 566, 74, 331, 94, 137 are the top 5. However, after the targeted attack, 74, 566, 94, 137, 446 dimensions of the dog encoding get the highest score. The top features have much more overlapping with the plane compared to that with the dog. As shown in Figure 1 (Right), this phenomenon still happens in the top 60 features. This phenomenon indicates that the features are not consistent enough since few of the original cat features get a score high enough, giving no cues for the original ground-truth class. The observation can be empirically be attributed to the fact that the loss functions of TRADES don't put any constraint on the structure of robust features. The form of solution probably looks similar to the left of Figure 1, where robust and useful figures are entangled together. But they're neither very robust nor useful. For example, the solution avoids fur texture, which is highly useful but not very robust, and shapes which are robust but not useful.

In contrast, humans don't compromise feature usage to achieve both robustness and generality. Humans use both very robust features and very useful features instead of something between, but they adaptively adjust the reliance on the two kinds of feature to ensure performance Agrawal; Walsh & Gluck (2015). As an example, we illustrate this process in Figure 2. Before humans see a husky, fur texture is enough for us to distinguish a husky from many other objects like a keyboard. Humans can rely on it, along with some other features like the shape of eyes, in most cases with great accuracy. However, given a wolf, where similar texture is also present, the texture feature would lead us to the wrong prediction. We would learn to lower the weights of fur texture but rely on features like shapes. But we only do this when we find a cue conflict, where features pointing to a different class. We still rely on fur for husky when we find it's safe to do so.

However, there are two major challenges in establishing such human-like adaptability. First, the preliminary of adaptability for machines is a well-disentangled representation where each feature has a clear influence on either robustness or usefulness. It's neither effective nor interpretable to work on entangled representations, which we will cover in more detail in Section 2.2. However, controlling representation structures is by nature hard for neural models that are optimized end-to-end. Second, the model has to understand the difference between examples, in terms of its risk against robustness and adjusts the prediction strategy accordingly. Namely, the algorithm needs to change its feature portfolio according to its judgment on the input type. which is nontrivial to automate.

We resolve the aforementioned challenges by a natural combination of two ideas: a counterfactual baseline that disentangles the representation, and a triple-head layer that dynamically calculates the

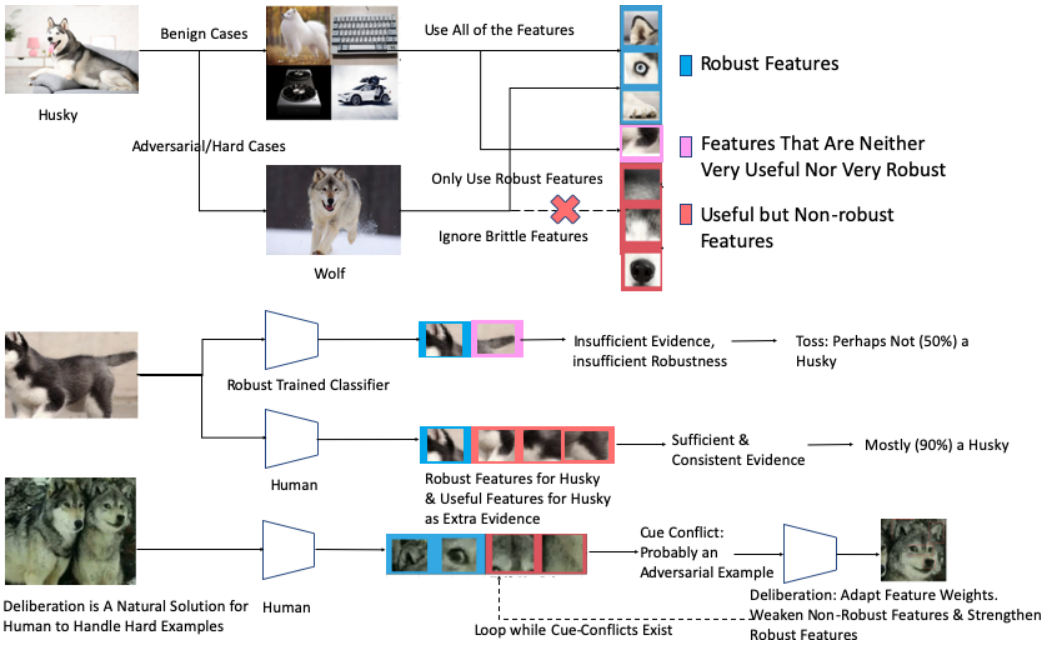


Figure 2: Figure 2 (Top): Standard-trained models adopt both robust and non-robust features that can distinguish Husky from other classes. Robust trained classifier would abandon non-robust features like texture. Figure 2 (Middle) The face of the dog is missing. Though humans naturally seek extra evidence as long as they do not contradict with the robust features present, e.g., the shape of ears. Adversarially trained models abandon brittle features. The remaining features are not sufficient to make solid predictions. Figure 2 (Bottom) Humans rely on features of various robustness and usefulness with great adaptability according to the input type. If some cues indicate contradicting classes, we would be more careful and rethink our prediction strategy.

posterior weights for each type of feature. The counterfactual baseline starts with the raw representations achieved from a robust classifier. It then masks part of the features as counterfactual and checks if the changes in loss match the counterfactual hypothesis. By doing so, it iteratively builds disentangled representations. The triple-head layer consists of three parallel output layers trained with different objectives that respectively take care of benign examples, adversarial examples, and those of uncertain types. The reweighting function is activated depending on the consistency among the predictions of the three heads. Through the cooperation of the three heads, our model can interpretably and controllably ensure both standard and adversarial accuracy.

In summary, the contribution of this paper is fourfold: (1)An interpretable robust representation method achieved by counterfactual manipulation; (2) A dynamic weighting mechanism, through which our proposed model achieves the state-of-the-art performance across several major benchmarks; (3) A theoretic analysis on how the number of cooperative features would affect the robustness of neural networks; (4) A framework that can improve downstream image generation tasks.

1.1 Related Work

1.1.1 Trade-off between Robustness and Generality

Several works have attempted to study the existence of the tradeoff between generality and robustness, as well as how to mitigate such tradeoff Raghunathan et al. (2020); Carmon et al. (2019); Zhang et al. (2019). Zhang et al. (2019) proposed a regularized loss which optimizes the trade-off between robustness and accuracy. Carmon et al. (2019) proposed a semi-supervised method called robust self-training(RST) to improve adversarial robustness, which enforces consistency on additional data labeled by standard-trained models during adversarial training. Raghunathan et al. (2020) attributed the tradeoff to network overparameterization and proposed a RST approach to eliminate the tradeoff in linear regression setting.

1.1.2 FEATURE ROBUSTNESS

Another line of work study the robustness of feature and its connection to model behavior. Geirhos et al. (2019) found shape features are more robust compared to texture features while networks tend to be biased toward textures instead shapes, which humans rely their judgement on. Zhang & Zhu (2019) similarly studied the bias of CNN and its influence on model behavior, where bias towards global structures like shapes or edges make model robust to perturbation. Ilyas et al. (2019) suggested that the robustness and generality of a model are decided by the robustness of features it adopted, where robust and non-robust features inherently exist in data.

1.1.3 OVERFITTING IN DEEP LEARNING

Li et al. (2020) empirically studied the generalization drop of adversarially trained networks and found early-stopping useful to alleviate such overfitting issue in adversarial training. Schmidt et al. (2018) theoretically showed the sample complexity of robust models is much larger than that of standard model and therefore require more data to train. Chen et al. (2020); Min et al. (2020) showed that the generality of adversarially trained models can actually be hurt by more training data, which challenged previous idea that more data can diminish the tradeoff between robustness and generality.

2 APPROACH

Setup. We now describe the overall architecture and training paradigm for the proposed model.

2.1 ROBUST TRAINING: A REVISIT

We begin this section with an introduction to the robust training schema. Adversarially robust classifiers are training using the robust optimization objective, where instead of minimizing the expected loss $\mathcal{L}$ over the data:

$$\mathbb{E}_{(x,y)\in\mathbb{D}}[\mathcal{L}(x,y)], \tag{1}$$

we minimize the worst case loss over a specific perturbation set Δ:

$$\mathbb{E}_{(x,y)\in\mathbb{D}}[max_{\sigma\in\Delta}\mathcal{L}(x+\sigma,y)]. \tag{2}$$

Typically, the set Δ captures imperceptible changes, such as small l_{inf} perturbations, the problem in equation (2) can be solved using adversarial training. TRADES points out that the objective function in equation (2) serves as an upper bound of the robust error $R_rob(f)$. In complex problem domains, however, this objective function might not be tight as an upper bound of the robust error, and might not capture the trade-off between natural and robust errors. Trades captures the trade-off between natural and robust errors and suggest minimizing:

$$min_f\, \mathbb{E}\{\underbrace{\phi(f(X)Y)}_{\text{for accuracy}} + \underbrace{max_{X'\in\mathbb{B}(X,\epsilon)}\phi(f(X)f(X')/\lambda}_{\text{regularization for robustness}}\}\,. \tag{3}$$

The first term encourages the natural error to be optimized by minimizing the difference between f(X) and Y, while the second term encourages the robust error prediction to be optimized by keeping prediction for x_{adv} consistent with x. Our methods mainly optimize the features learned by TRADES, making them more interpretable and robust.

2.2 THE COUNTER-FACTUAL BASELINE

Assuming we have already got two parts of features disentangled by robustness and accuracy. Then the counter fact is: we only have the relative non-robust features, model robustness should be lower than the fact. Formally we have the below formulation:

$$Rob[f(x)] < Rob[f_{rob}(x)] \tag{4}$$

The above analysis sheds light on the algorithmic design of feature disentanglement. Given a training image x and its adversarial version x_{adv}, we first derive the whole representation $f(x_{adv})$ of

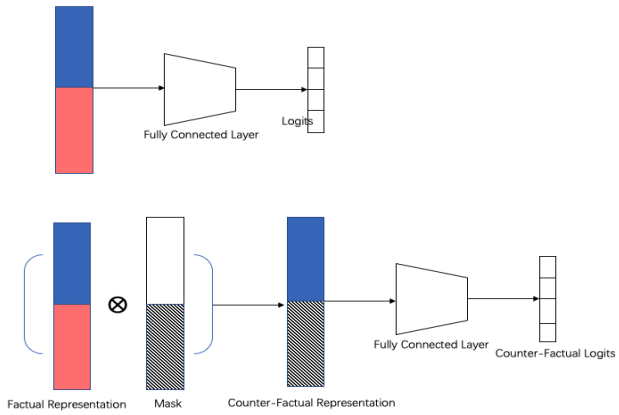


Figure 3: Top: Factual representation Bottom: Mask the first K dimension of factual representation with a binary mask to get the counter-factual representation.

x_{adv} through forwarding propagation. To characterize the counter fact, we mask the first K features of $f(x_{adv})$ using a binary mask, and the masked representation is treated as the relatively more robust part of the feature(with relatively low accuracy). The counter-factual features should have lower robust loss than the factual features, which can be formalized below:

$$l_{counter} = \frac{KL[f(x_{adv}), f(x)]}{KL[f_{rob}(x_{adv}), f(x)]} \tag{5}$$

2.3 REWEIGHTING

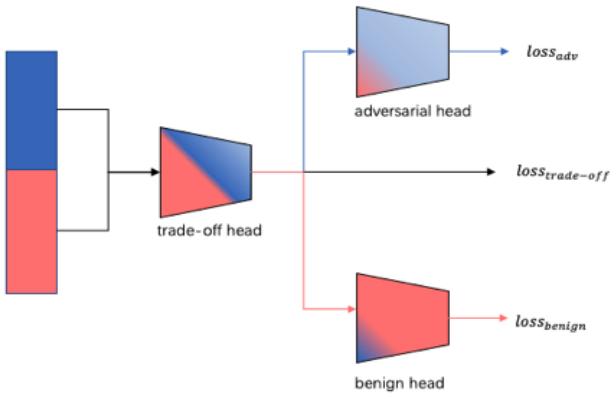


Figure 4: The triple-head model: all examples first go through trade-off head to compute robust prediction, non-robust prediction and trade-off prediction. If robust prediction meets trade-off prediction, then we use adversarial head to get final prediction. If non-robust prediction meets trade-off prediction, then we use adversarial head to get final prediction. Otherwise, we use the prediction of the trade-off head.

Since the objective function of our model is to find the equilibrium between robustness and accuracy, the weights for combining the disentangled features in the fully connective layer should not biased towards either part. The trade-off is a waste of high accuracy of the non-robust part with high accuracy when it comes to benign example, and high robustness of the robust part when it comes to adversarial example. A natural idea to solve this is to dynamically calculate the posterior weight for each type of feature according to the example type: the model puts more weight on the robust part if

the input is probably adversarial, and more weight on the non-robust part if the input is benign. The above process can be implemented by replacing the original fully connective layer with a triple-head layer. Head one, denoted as $head_{tradeoff}$, represents the head biased towards neither feature, which is the head learned using benign data and adversarial data. The loss of the head can be formalized as follows:

$$f_1(x) = head_{tradeoff}(f(x)), \tag{6}$$

$$loss_{tradeoff} = min_f \mathbb{E}[f_1(x), Y] + maxKL(f_1(x), f_1(x')), x' \in B(x, \epsilon). \tag{7}$$

Head two, denoted as $head_{benign}$, represents the head biased towards non-robust features in order to gain high accuracy on the benign images. The head is trained using only benign data. The loss of the head can be formalized as follows:

$$f_2(x) = head_{benign}(f(x)), \tag{8}$$

$$loss_{benign} = min_f \mathbb{E}(f_2(x), Y). \tag{9}$$

Head three, denoted as $head_{adv}$, represents the head biased towards robust features in order to gain high robustness on the perturbed images. The head is trained using only adversarial data. The loss of the head can be formalized as follows:

$$f_3(x) = head_{adv}(f(x)), \tag{10}$$

$$loss_{adv} = min_f max_{x \in B(x,\epsilon)} KL(f_3(x), f_3(x')). \tag{11}$$

We employ a multi-task learning approach for the joint learning of embeddings for the three heads. At the inference stage, the $head_{tradeoff}$ is used as a pointer to decide whether to use the robust head or the non-robust head. For a given input x, we first get its feature representation $f(x)$. Then we calculate the feature score of the robust part, $rob(f(x))$, and non-robust part, $nr(f(x))$ and draw the prediction from

$$y_{pred} = argmax(rob(f(x)) + nr(f(x))) \tag{12}$$

$$y_{pred_{rob}} = argmax(rob(f(x)) \tag{13}$$

$$y_{pred_{nr}} = argmax(nr(f(x)) \tag{14}$$

If the three predictions point to the same class, it means we encounter a benign example, and we use the $head_{benign}$ for prediction. If the y_{pred} and $y_{pred_{rob}}$ point to the same class, it means we encounter an adversarial example, and we use the $head_{adv}$ for prediction. We use the $head_{tradeoff}$ to process the other cases since we cannot judge the sample type.

2.4 Implementation

We use $d = 128$ as the dimension of robust features. We train the model by minimizing the losses in a multi-task learning way.

$$loss = loss_{tradeoff} + loss_{benign} + loss_{adv} + loss_{counter} \tag{15}$$

3 Experiments

CIFAR-10 Setup We use wide residual network WRN-34-10, the same as TRADES. We set perturbation $\epsilon = 0.031$, perturbation step size $\eta = 0.007$, number of iterations $K = 10$, and run 200 epochs on the training set.
MNIST Setup We use the CNN architecture with four convolutional layers, followed by three fully-connected layers. We set perturbation $\epsilon = 0.3$, perturbation step size $\eta_1 = 0.01$, number of iterations $K = 40$, learning rate $\eta_2 = 0.01$, and run 100 epochs on the training set.

3.1 INTERPRETABILITY

We first show our model's accuracy and robustness using each part alone to confirm the claimed disentangle structure on CIFAR-10. We also demonstrate the feature activation of the ground truth class of each part when the attack succeeds.

Shown in Table1, the robust part surpass the non-robust part and the baseline in robustness, while it's inferior to the other parts in accuracy by a remarkable margin. Also, when evaluated on the false-classified set after adversarial attacks, the robust part shows higher similarity to the embedding of the ground truth than the target class, compared with the completely dissimilar situation of the baseline model and the non-robust part.

This validates the nice disentanglement and Interpretability of our learned feature representation. We also include the results under SVHN. The gap between our model and the baseline methods gets further enlarged across all metrics as shown in Table 1.

	Robustness	Accuracy
Robust part	**61.5**	80.7
Non-robust part	49.14	**88.64**
Baseline	55.4	84.9

Table 1: Accuracy and robustness of each part of our model alone compared to TRADES on Cifar-10

3.2 PERFORMANCE AGAINST ADVERSARIAL ATTACKS

We test our model with a reweighting module under the white-box attack, which is the most difficult threat model and no effective defense exists yet. The model of Hendrycks et al. (2019) is based on ImageNet adversarial pretraining and is less directly comparable to ours due to the differences in external data and training methods. Finally, we perform standard self-training using the unlabeled data, which offers a moderate 0.4% improvement in standard accuracy over the intermediate model but is not adversarially robust.

	PGD	C & W	No Attack
Ours	**63.1**	**65.4**	**91.4**
TRADES	55.4	65.0	84.9
Adv Pre-training	57.7	-	87.1
Madry	45.8	47.8	87.3

Table 2: Accuracy and robustness of each part of our model alone compared to TRADES on Cifar-10

3.3 EXPLAINABLE IMAGE GENERATIVE MODEL

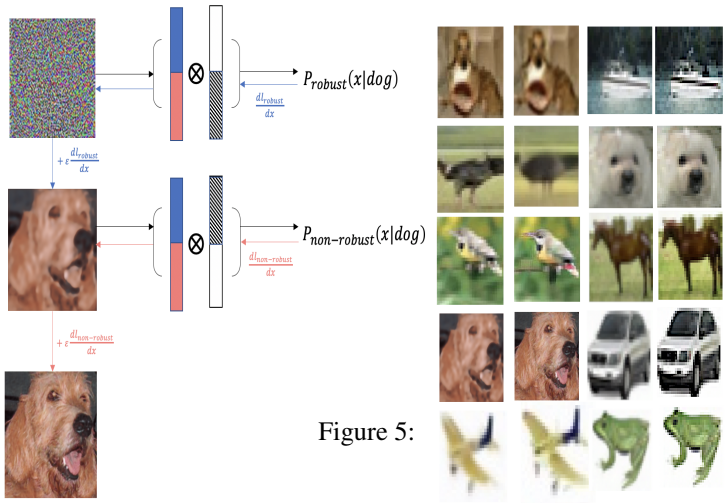


Figure 5:

Many of the modern image generative models, such as WGAN, BIGGAN etc. , can produce high-quality novel images from datasets at one hand, while on the other hand, can be tricky to train and often require intense computation.

As a side application, we show how the learned disentangled classifier can directly synthesizing realistic natural images, in an explainable and data-efficient way, without any special training or auxiliary networks. Fig 7 shows the general idea. Our model undergoes two-stage optimization during the generation process: 1) an early stage of robust feature generation. 2) a later stage of non-robust feature generation.

Formally, for a trained classifier to generate a sample of class y, we sample a seed from G_y and minimize the loss L of label y by adding small adjustment to x through gradient descent. The progress is like the reverse procedure of adversarial attack.

$$x = argmin_{\|x'-x_0\|_2 \leq \epsilon} \mathcal{L}(x', y), x_0 \sim \mathcal{G}_y, \tag{16}$$

At stage one, we only use the robust part of the classifier. The purpose of this is to generate robust features of the target class first, such as shapes of ears or faces of a dog in Figure 5. We can observe from the second row of Figure 5, a well-recognizable sketchy dog with fine-grained contour has already been generated at the end of stage one. At state two, we only use the non-robust part of the classifier aiming to complement the missing non-robust details. We can find out that delicate fur texture and pupils are added at this stage. The progress draws an analogy between human visual imagination and generative model: when it comes to the word 'dog', a silhouette of a dog first emerges to most people's mind, and then with other cues such as 'white dog', 'husky', more details are added to the silhouette, making the image more precise. Figure 5 Right demonstrates samples produced using our methods.

4 A Theoretic Analysis of Framework Design

How Many Heads Do We Need? During our experiments, we empirically assume three three heads can cover all the cases. In this section, we discuss the theoretic lower bound of the number of heads.
Setup. Suppose $\mathcal{S}$ is a test set consists of N samples. The attacker would draw a sample from $\mathcal{S}$ to attack. $\mathcal{P} = \{p_i\}_n$ is the set of robust accuracy of each head. n is the number of heads. Our goal is to find the smallest n so that for any sample attacked in $\mathcal{S}$, there are more heads correctly predict the sample than those do wrongly. Given an input λ, let x_i represents whether the i_th head make the right prediction.

$$x_i = \begin{cases} 1, & \text{wp. p prediction is correct} \\ 0, & \text{wp. 1- p prediction is false} \end{cases} \tag{17}$$

$$Pr(x \leq \frac{1}{2p}\mathbb{E}[x]) \leq e^{-\frac{(1-\frac{1}{2p})^2 * np}{2}}, \tag{18}$$

$$Pr(\text{more heads predict wrong}) \leq |S| * -\frac{(1-\frac{1}{2p})^2 * np}{2}. \tag{19}$$

If we would like constrain the attacker's success rate lower ϵ,

$$|S| * -\frac{(1-\frac{1}{2p})^2 * np}{2} \leq \epsilon, \tag{20}$$

$$n \geq \frac{2log\frac{|S|}{2}}{p(1-\frac{1}{2p})^2} \tag{21}$$

So the lower bound of n is $\frac{2log\frac{|S|}{2}}{p(1-\frac{1}{2p})^2}$.

5 Conclusion

We present a multi-task learning approach that can dynamically adjust feature dependence according to input types to interpretably ensure robustness and generality. A counter-factual approach is proposed for constructing well disentangled feature representations to make such dynamic reweighting available and interpretable. A triple-head structure is proposed to dynamically decide the feature weights by comparing the predictions of all the heads. The proposed framework shows superiority over state-of-the-art models across several benchmarks and proven to be useful for downstream generation tasks.

REFERENCES

Anurag A Agrawal. Transgenerational induction of defences in animals and plants. *Nature.*, 401 (6748). ISSN 0028-0836.

Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya P. Razenshteyn. Adversarial examples from computational constraints. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 831–840. PMLR, 2019. URL http://proceedings.mlr.press/v97/bubeck19a.html.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C. Duchi, and Percy Liang. Unlabeled data improves adversarial robustness. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 11190–11201, 2019. URL http://papers.nips.cc/paper/9298-unlabeled-data-improves-adversarial-robustness.

Lin Chen, Yifei Min, Mingrui Zhang, and Amin Karbasi. More data can expand the generalization gap between adversarially robust and standard models. *CoRR*, abs/2002.04725, 2020. URL https://arxiv.org/abs/2002.04725.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2712–2721. PMLR, 2019. URL http://proceedings.mlr.press/v97/hendrycks19a.html.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 125–136, 2019. URL http://papers.nips.cc/paper/8307-adversarial-examples-are-not-bugs-they-are-features.

Jieling Li, Hao Zhang, and Zhiqiang Wei. The weighted word2vec paragraph vectors for anomaly detection over HTTP traffic. *IEEE Access*, 8:141787–141798, 2020. doi: 10.1109/ACCESS.2020.3013849. URL https://doi.org/10.1109/ACCESS.2020.3013849.

Yifei Min, Lin Chen, and Amin Karbasi. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization. *CoRR*, abs/2002.11080, 2020. URL https://arxiv.org/abs/2002.11080.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *CoRR*, abs/2002.10716, 2020. URL https://arxiv.org/abs/2002.10716.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pp. 5019–5031, 2018. URL http://papers.nips.cc/paper/7749-adversarially-robust-generalization-requires-more-data.

Matthew M. Walsh and Kevin A. Gluck. Mechanisms for robust cognition. *Cogn. Sci.*, 39(6):1131–1171, 2015. doi: 10.1111/cogs.12192. URL https://doi.org/10.1111/cogs.12192.

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 Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482. PMLR, 2019. URL http://proceedings.mlr.press/v97/zhang19p.html.

Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7502–7511. PMLR, 2019. URL http://proceedings.mlr.press/v97/zhang19s.html.

A Appendix

You may include other additional sections here.