Part-Based Models Improve Adversarial Robustness

Chawin Sitawarin[1]* **Kornrapat Pongmala**[1] **Yizheng Chen**[1] **Nicholas Carlini**[2]

David Wagner[1]

[1] EECS Department, University of California, Berkeley
[2] Google

Abstract

We show that combining human prior knowledge with end-to-end learning can improve the robustness of deep neural networks by introducing a *part-based* model for object classification. We believe that the richer form of annotation helps guide neural networks to learn more robust features without requiring more samples or larger models. Our model combines a part segmentation model with a tiny classifier and is trained end-to-end to simultaneously segment objects into parts and then classify the segmented object. Empirically, our part-based models achieve both higher accuracy and higher adversarial robustness than a ResNet-50 baseline on all three datasets. For instance, the clean accuracy of our part models is up to 15 percentage points higher than the baseline's, given the same level of robustness. Our experiments indicate that these models also reduce texture bias and yield better robustness against common corruptions and spurious correlations. The code is publicly available at `https://github.com/chawins/adv-part-model`.

1 Introduction

Adversarial training [Madry et al., 2018] is the state-of-the-art method for improving the adversarial robustness of deep neural networks. Recent work has made substantial progress by scaling adversarial training to very large datasets [Carmon et al., 2019, Gowal et al., 2021a, Rebuffi et al., 2021] These works fall in line with a recent trend of deep learning on "scaling up," i.e., training large models on massive datasets [Kaplan et al., 2020]. Unfortunately, progress has begun to stagnate here as we have reached a point of diminishing returns: for example, Gowal et al. [2021a] show that an exponential increase in model size and training samples will only yield a linear increase in robustness.

Our work presents a novel alternative to improve adversarial training: we propose to utilize *additional supervision* that allows for *a richer learning signal*. We hypothesize that an auxiliary human-aligned learning signal will guide the model to learn more robust and more generalized features.

To demonstrate this idea, we propose to classify images with a *part-based model* that makes predictions by recognizing the parts of the object in a bottom-up manner. We make use of images that are annotated with part segmentation masks. We propose a simple two-stage model that combines *a segmentation model* with *a classifier*. Fig. 1 illustrates our model. The idea is that this approach may guide the model to attend more to global shape than to local fine-grained texture, hopefully yielding better robustness. We then combine this part-based architecture with adversarial training.

We show that our model achieves strong levels of robustness on three realistic datasets where our part-based models outperform the ResNet-50 baselines on both clean and adversarial accuracy simultaneously. **For any given value of clean accuracy, our part models achieve more than 10**

*Contact: `chawins@berkeley.edu`, `kornrapatp@berkeley.edu`

ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

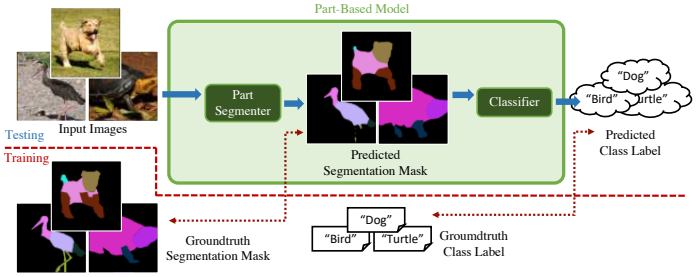


Figure 1: Our part-based model consists of (1) the part segmenter and (2) a tiny classifier. We train it for the object classification task end-to-end using part-level segmentation labels to improve its robustness.

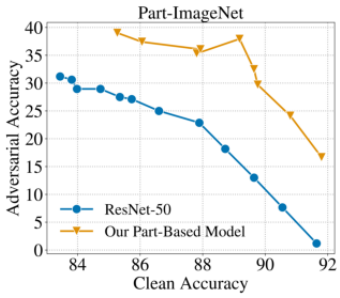


Figure 2: Accuracy-robustness trade-off of our part model and the ResNet-50 baseline on the Part-ImageNet dataset.

percentage points higher adversarial accuracy compared to the baseline on Part-ImageNet (see Fig. 2). Our part-based models also improve non-adversarial robustness, without any specialized training or data augmentation. We believe that our part-based model is the first promising example of how a richer supervised training signal can substantially improve the robustness of neural networks.

2 Related Work

Adversarial robustness. Adversarial training [Madry et al., 2018] has become a standard method for training robust neural networks against adversarial examples. Many improvements on this technique have been proposed [Zhang et al., 2019, Xie et al., 2019, Pang et al., 2019, Qin et al., 2019, Rice et al., 2020]. More recent state-of-the-art methods follow a recent trend of "large-scale learning from weak signals." Carmon et al. [2019] and Gowal et al. [2021a] rely on a large number of unlabeled training data while others utilize large generative models for data augmentation [Rebuffi et al., 2021] or synthetically generating more training samples [Gowal et al., 2021b, Sehwag et al., 2022]. The improvement from scaling up, however, has started to reach its limit [Gowal et al., 2021a].

Part-based models. Part models generally refer to hierarchical models that recognize objects from their parts in a bottom-up manner , e.g., Deformable Part Models [Endres et al., 2013, Felzenszwalb et al., 2010, Chen et al., 2014, Girshick et al., 2015, Cho et al., 2015]. The most relevant work on adversarial robustness and part-based models are Freitas et al. [2020] and Chandrasekaran et al. [2019]. Freitas et al. [2020] detect adversarial examples by using a Mask R-CNN to extract object parts and verify that they align with the predicted label. However, unlike our work, their scheme is not evaluated against an adaptive adversary. Chandrasekaran et al. [2019] propose a robust classifier with a hierarchical structure where each node separates inputs into multiple groups, each sharing a certain feature (e.g., object shape). Unlike our part model, its structure depends heavily on the objects being classified and does not utilize part segmentation.

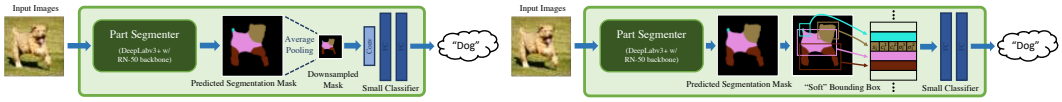
3 Part-Based Models

3.1 Architecture

Our part-based model consists of two stages: the segmenter and a tiny classifier. The segmenter takes the original image x as the input and outputs logits for the $K+1$ masks denoted by $\hat{M} := f_{\text{seg}}(x)$. The second-stage classifier then processes $\hat{M}$ and returns the predicted class probability $f(x) = f_{\text{cls}}(\hat{M}) = f_{\text{cls}}(f_{\text{seg}}(x))$. Fig. 1 visually summarizes our design.

We use DeepLabv3+ [Chen et al., 2018] with ResNet-50 backbone [He et al., 2016] as the segmenter, but our part-based model is agnostic to the choice of segmenter architecture. Additionally, all of the classifiers are designed to be end-to-end differentiable. This facilitates the evaluation process as well making our models compatible with adversarial training.

Classifier design principles. We experimented with various classifier architectures, each of which processes the predicted masks differently. Our design criteria were:



(a) Downsampled Part-based Model

(b) Bounding-Box Part-based Model

Figure 3: Illustration of our two part-based models: (a) downsampled and (b) bounding-box.

1. *Part-based classification*: The classifier should rely on the segmenter's output only and not the original image. If the segmenter is robust, the class label can be obtained from the masks alone.
2. *Disentangle irrelevant features*: The background is not part of the object being classified so the segmenter should separate it from the foreground pixels. We either simply drop the predicted background pixels or leave it to the second-stage classifier to correctly utilize them.
3. *Location-aware*: The second-stage classifier should utilize the *location* and the *size* of the parts, in addition to their existence.

Following these principles, we designed four part-based classifiers but focus on two, *Downsampled* and *Bounding-Box*, in the main paper. Appendix C has details on the others.

Downsampled part-based model. This model first applies softmax on the predicted mask logits $\hat{M}$. After that, the masks are downsampled to size 4×4 by adaptive average pooling before being passed to a tiny neural network with one convolution layer and two fully-connected layers (see Fig. 3a). Downsampling maintains coarse-grained spatial information about each part's rough shape and location while compressing high-dimensional masks to a low-dimensional feature vector. This keeps the classifier small, making the part-based model comparable to the normal ResNet-50 in size.

Bounding-box part-based model. The bounding-box classifier also compresses the predicted mask to a lower-dimensional representation, but instead of downsampling, it uses bounding boxes. Specifically, it processes each of the logit segmentation masks, $\hat{M}_i$, into K "soft" bounding boxes, one for each object part, excluding the background channel (see Fig. 3b). Each bounding box is represented by five features: a logit score ($s_i \in [0, 1]$), a centroid ($c_i^1, c_i^2 \in [-1, 1]$) representing the (normalized) 2D coordinates of the center of the bounding box, and a standard deviation ($\sigma_i^1, \sigma_i^2 \in [0, 1]$) capturing the height and the width of the box. We describe how these features are computed in Appendix C.1. Finally, a tiny fully-connected neural network predicts the class label given v and no other information. Crucially, we ensure that the computation of these features is differentiable to enable effective training as well as reliable evaluation of adversarial robustness.

3.2 Experiment Setup

We demonstrate our part models on three datasets where part-level annotations are available: Part-ImageNet [He et al., 2021], Cityscapes [Meletis et al., 2020], and PASCAL-Part [Chen et al., 2014].

Cityscapes and PASCAL-Part were originally created for segmentation, so we construct a classification task from them. For Cityscapes, we create a human-vs-vehicle classification task. For each human or vehicle instance with part annotations, we crop a square patch around it with some random amount of padding and assign the appropriate class label. For PASCAL-Part, we simply assign a label to each image based on the largest object in that image.

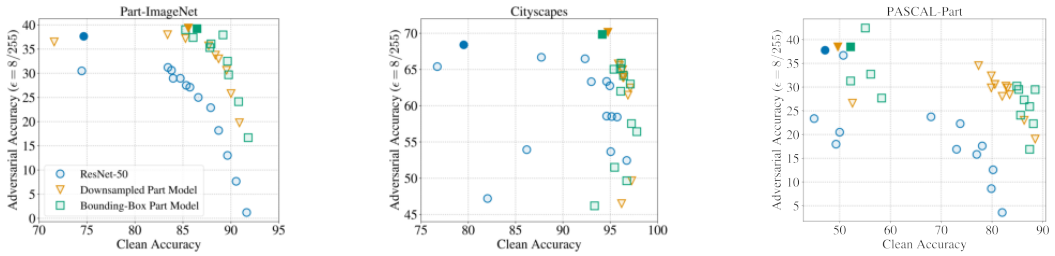
ResNet-50 [He et al., 2016] is our baseline. Our part-based models (which use DeepLabv3+ with ResNet-50 backbone) are of similar size as the baseline (26.7M vs 25.6M parameters). All models are trained with SGD and a batch size of 128, using either adversarial training or TRADES, with 10-step ℓ_∞-PGD with $\epsilon = 8/255$ and step size of $2/255$. All models are pre-trained on ImageNet and are trained end-to-end. Appendix A and Appendix B presents additional detail.

4 Robustness Evaluation

We compare the adversarial robustness and the clean accuracy of the part-based models to the ResNet-50 baseline. We must examine both metrics at the same time since there is a known trade-off between them [Tsipras et al., 2019]. We use AutoAttack [Croce and Hein, 2020], a standard and reliable ensemble of attacks, to compute the adversarial accuracy of all models. We ensure that our evaluation is free from the notorious gradient obfuscation problem (see Appendix E.4).

Table 1: Comparison of normal and part-based models under different training methods. Adversarial accuracy is computed with AutoAttack ($\epsilon = 8/255$). For TRADES, we first train a ResNet-50 model with clean accuracy of at least 90%, 96%, and 80% for Part-ImageNet, Cityscapes, and PASCAL-Part, respectively, then we train part-based models with similar or slightly higher clean accuracy.

Training Method	Models	Part-ImageNet		Cityscapes		PASCAL-Part	
		Clean	Adv.	Clean	Adv.	Clean	Adv.
PGD [Madry et al., 2018]	ResNet-50	74.7	37.7	79.5	68.4	47.1	37.8
	Downsampled Part Model	85.6 (↑ 10.9)	39.4 (↑ 1.7)	94.8 (↑ 15.3)	70.2 (↑ 1.8)	49.6 (↑ 2.5)	38.5 (↑ 0.7)
	Bounding-Box Part Model	86.5 (↑ 11.8)	39.2 (↑ 1.5)	94.2 (↑ 14.7)	69.9 (↑ 1.4)	52.2 (↑ 5.1)	38.5 (↑ 0.7)
TRADES [Zhang et al., 2019]	ResNet-50	90.6	7.7	96.7	52.5	80.2	12.6
	Downsampled Part Model	90.9 (↑ 0.3)	19.8 (↑ 12.1)	97.1 (↑ 0.4)	62.5 (↑ 10.0)	83.1 (↑ 2.9)	29.9 (↑ 17.3)
	Bounding-Box Part Model	90.8 (↑ 0.2)	24.1 (↑ 16.4)	97.1 (↑ 0.4)	63.0 (↑ 10.5)	88.5 (↑ 8.3)	29.5 (↑ 16.9)



(a) Part-ImageNet (b) Cityscapes (c) PASCAL-Part

Figure 4: Accuracy and robustness trade-off plots of normal and part-based models trained on (a) Part-ImageNet, (b) Cityscapes, and (c) PASCAL-Part. The filled dots represent PGD adversarial training while the unfilled ones denote TRADES with different values of its parameter β.

Table 1 compares the models under two training methods: PGD adversarial training [Madry et al., 2018] and TRADES [Zhang et al., 2019]. For PGD-trained models, both of the part-based models achieve about **3–15 percentage points higher clean accuracy than the baseline** with similar adversarial accuracy. The models trained on Cityscapes show the largest improvement, followed by ones on Part-ImageNet and PASCAL-Part. TRADES allows controlling the tradeoff between clean vs adversarial accuracy, so we choose models with similar clean accuracy and compare their robustness. **The part models outperform the baseline by about 16, 11, and 17 percentage points on Part-ImageNet, Cityscapes, and PASCAL-Part, respectively.** These results show that part-based models significantly improve adversarial robustness.

Fig. 4 plots the robustness-accuracy trade-off curves for all three datasets, generated by sweeping the TRADES hyperparameter β (see Section B.1). Our part-based models are closer to the top-right corner of the plot, indicating that they outperform the baseline on both clean and adversarial accuracy.

Fig. 5 shows ten randomly chosen test samples from Part-ImageNet along with their predictions, with and without the attack. Most of the part-based models, including this one, achieve above 80% pixel-wise segmentation accuracy on clean samples and about 70% on adversarial ones. Successful attacks can change most, but not all, foreground pixels to the wrong class, but the shape and foreground-vs-background prediction for each part remains correct. This suggests that part-based models may learn shape features that are more difficult to manipulate, an observation that aligns with our quantitative results on shape-vs-texture bias below. We suspect the robustness of these part shapes might account for the model's improved robustness.

Non-adversarial robustness. We evaluate the models on three scenarios: common corruptions [Hendrycks and Dietterich, 2019], foreground-vs-background spurious correlation [Xiao et al., 2021], and shape-vs-texture bias [Geirhos et al., 2019]. **On all of the benchmarks, the part-based models outperform the baseline by 3–7 percentage points** (see Tables 2, 3, and 4). The improve-

ment over the ResNet-50 baseline is statistically significant (two-sample t-test, p-values below 10^{-6}). We note that these robustness gains do *not* come at a cost of clean accuracy as the clean accuracy of our part models is about 1% higher on average than that of the ResNet-50. See Appendix D for detailed explanations and results.

Acknowledgements

We would like to thank Vikash Sehwag for the feedback on the paper. This research was supported by the Hewlett Foundation through the Center for Long-Term Cybersecurity (CLTC), by the Berkeley Deep Drive project, and by generous gifts from Open Philanthropy and Google Cloud Research Credits program under Award GCP19980904.

References

M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: A query-efficient black-box adversarial attack via random search. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, volume 12368, pages 484–501, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58591-4 978-3-030-58592-1. doi: 10.1007/978-3-030-58592-1_29. 17

N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *arXiv:1902.06705 [cs, stat]*, Feb. 2019. 16

Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang. Unlabeled data improves adversarial robustness. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1, 2

V. Chandrasekaran, B. Tang, N. Papernot, K. Fawaz, S. Jha, and X. Wu. Rearchitecting classification frameworks for increased robustness. *arXiv preprint arXiv:1905.10900*, 2019. 2

J. Chen and Q. Gu. RayS: A ray searching method for hard-label adversarial attack. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1739–1747, Virtual Event CA USA, Aug. 2020. ACM. ISBN 978-1-4503-7998-4. doi: 10.1145/3394486.3403225. 17

L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 833–851, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01234-2. 2

X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2, 3, 8

M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1201–1210, 2015. 2

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR, July 2020. 3, 9, 17

I. Endres, K. J. Shih, J. Jiaa, and D. Hoiem. Learning collections of part models for object recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 939–946, Portland, OR, USA, June 2013. IEEE. ISBN 978-0-7695-4989-7. doi: 10.1109/CVPR.2013.126. 2

M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 8

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 2

S. Freitas, S.-T. Chen, Z. J. Wang, and D. H. Chau. UnMask: Adversarial detection and defense through robust feature alignment. In *IEEE BigData*, pages 1081–1088, Nov. 2020. 2

R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. ImageNet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 4, 12

R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 437–446, 2015. 2

S. Gowal, C. Qin, J. Uesato, T. Mann, and P. Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv:2010.03593 [cs, stat]*, Mar. 2021a. 1, 2

S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. Mann. Improving robustness using generated data. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021b. 2

S. Gupta, P. Dube, and A. Verma. Improving the affordability of robustness training for DNNs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 9

J. He, S. Yang, S. Yang, A. Kortylewski, X. Yuan, J.-N. Chen, S. Liu, C. Yang, and A. Yuille. PartImageNet: A large, high-quality dataset of parts. *arXiv:2112.00933 [cs]*, Dec. 2021. 3, 8

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. doi: 10.1109/CVPR.2016.90. 2, 3

D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 4, 12

J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models, Jan. 2020. 1

I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 9

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 2, 4

P. Meletis, X. Wen, C. Lu, D. de Geus, and G. Dubbelman. Cityscapes-panoptic-parts and PASCAL-panoptic-parts datasets for scene understanding, Apr. 2020. 3, 8

T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu. Improving adversarial robustness via promoting ensemble diversity. *arXiv:1901.08846 [cs, stat]*, May 2019. 2

C. Qin, J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli. Adversarial robustness through local linearization. *Advances in Neural Information Processing Systems*, 32, 2019. 2

S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. Mann. Fixing data augmentation to improve adversarial robustness. *arXiv:2103.01946 [cs]*, Mar. 2021. 1, 2

L. Rice, E. Wong, and Z. Kolter. Overfitting in adversarially robust deep learning. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8093–8104. PMLR, July 2020. 2

L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Mądry. Adversarially robust generalization requires more data. *arXiv:1804.11285 [cs, stat]*, May 2018. 25

V. Sehwag, C. Hazirbas, A. Gordo, F. Ozgenel, and C. C. Ferrer. Generating high fidelity data from low-density regions using diffusion models. *arXiv:2203.17260 [cs]*, Mar. 2022. 2

R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt. Measuring robustness to natural distribution shifts in image classification. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18583–18599. Curran Associates, Inc., 2020. 13

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. 3

K. Y. Xiao, L. Engstrom, A. Ilyas, and A. Madry. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations*, 2021. 4, 12

C. Xie, Y. Wu, L. V. D. Maaten, A. L. Yuille, and K. He. Feature denoising for improving adversarial robustness. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:501–509, 2019. ISSN 9781728132938. doi: 10.1109/CVPR.2019.00059. 2

H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019. 2, 4, 11

A Datasets

Part-ImageNet. Proposed by He et al. [2021], this dataset is a subset of ImageNet-1K where the 158 of the original classes are grouped into 11 coarse-grained classes, e.g., "Quadruped," "Biped," "Reptile," etc. Each object is accompanied by pixel-wise annotation of 2–5 parts. For instance, a quadruped may have up to four segmentation masks for its head, body, feet, and tail. The dataset is originally proposed for part segmentation or part discovery tasks and is publicly available to download.[2] We note that the Part-ImageNet dataset splits the data by their original ImageNet-1K classes, i.e., 109, 19, and 30 classes for training, validation, and test sets, respectively. This allows one to measure generalization across sub-population under the same group. However, our focus is different; we want to evaluate the robustness under a similar setting to CIFAR-10 whose samples are split i.i.d. Hence, for this paper, we ignore the original ImageNet class and re-partition the dataset randomly, independent of its original class. The Part-ImageNet dataset has 24,095 samples in total.

Cityscapes. The Cityscapes dataset is a driving-oriented image dataset whose data were collected from a dashboard camera [Cordts et al., 2016]. We use the part-aware panoptic annotations on Cityscapes from Meletis et al. [2020] to create our classification dataset. The Cityscapes dataset is available under a non-commercial license[3], and the annotation is available under Apache-2.0.[4] Five kinds of objects are part-annotated, and we group them into two classes. Specifically, "person" and "rider" are grouped as "human," and "car," "truck," and "bus" as "vehicle." We use the same part labels as Meletis et al. [2020] where humans are annotated with "torso," "head," "arms," and "legs," and vehicles with "windows," "wheels," "lights," "license plate," and "chassis."

Since the samples in Cityscapes are wide-angle photos containing numerous objects, we crop each annotated object out to create a classification dataset. In particular, we crop each patch into a square and then add a small amount of extra random padding (0–10% of the image size). Additionally, we also filter out small objects that have the total area, determined from the segmentation mask of the entire object, less than 1000 pixels. After filtering, we are left with 29,728 samples in the dataset.

PASCAL-Part. The PASCAL-Part dataset [Chen et al., 2014] provides part-aware segmentation annotation of the PASCAL VOC (2010) dataset [Everingham et al., 2010] which is an object recognition and detection dataset. Both the annotations and the original dataset are available to the public.[5] The original PASCAL-Part dataset comprises 20 classes, but most of them have 500 or fewer samples. To ensure that we have a sufficient number of samples per class and avoid the class imbalance problem, we opt to select only the top-five most common classes: "aeroplane," "bird," "car," "cat," and "dog." In PASCAL-Part, the parts are annotated in a more fine-grained manner, compared to the other two datasets. For example, the legs of a dog are labeled as front or back and left or right. To make the number of parts per object manageable and comparable to the other two datasets, we group multiple parts of the same type together, e.g., all legs are labeled as "legs." Our PASCAL-Part dataset has 3,662 samples in total.

We also emphasize that we do not use a common benchmark dataset such as CIFAR-10 since it is not part-annotated and is too low-resolution to be useful in practice. The datasets we use are more realistic and have much higher resolution. For training and testing the models, we use the same preprocessing and data augmentation as commonly used for the ImageNet dataset. Specifically, the training samples are randomly cropped and resized to 224×224 pixels, using PyTorch's `RandomResizedCrop` function with the default hyperparameters, and applied a random horizontal flip. Test and validation samples are center cropped to 256×256 pixels and then resize to 224×224 pixels.

[2] https://github.com/tacju/partimagenet.

[3] https://www.cityscapes-dataset.com/license/

[4] For the Cityscape dataset, https://www.cityscapes-dataset.com/, and for the annotation, https://github.com/pmeletis/panoptic_parts/tree/master/panoptic_parts/cityscapes_panoptic_parts/dataset_v2.0.

[5] PASCAL VOC and its license can be found at http://host.robots.ox.ac.uk/pascal/VOC/voc2010/, and for PASCAL-Part, see https://roozbehm.info/pascal-parts/pascal-parts.html.

B Detailed Experiment Setup

Here, we provide information regarding the model implementation in addition to Section 3.2. All models are adversarially trained for 50 epochs. To help the training converge faster, we also pre-train every model on clean data for 50 epochs before tuning on adversarial training as suggested by Gupta et al. [2020]. We save the weights with the highest accuracy on the held-out validation data which does not overlap with the training or the test set. We use the cosine annealing schedule to adjust the learning rate as done in Loshchilov and Hutter [2017]. Training is early stopped according to adversarial accuracy computed on the held-out validation set. Our experiments are conducted on Nvidia GeForce RTX 2080 TI and V100 GPUs.

To evaluate all the models, we rely on both the strong ensemble AutoAttack and the popular PGD attack. However, the AutoAttack is always stronger than the PGD attack in all of the cases we experiment with so we only report the adversarial accuracy corresponding to the AutoAttack in the main paper. AutoAttack comprises four different attacks: adaptive PGD with cross-entropy loss (`apgd-ce`), targeted adaptive PGD with DLR loss (`apgd-t`), targeted FAB attack (`fab-t`), and Square attack (`square`) [Croce and Hein, 2020]. However, since the DLR loss requires that there are four or more classes, we have to adapt the AutoAttack on the Cityscapes dataset which has two classes. As a result, we use only three attacks and remove the targeted ones which leave adaptive PGD with cross-entropy loss (`apgd-ce`), FAB attack (`fab`), and Square attack (`square`). We use the default hyperparameters for all of the attacks in AutoAttack. For the PGD attack, we use a step size of 0.001 with 100 iterations and five random restarts.

B.1 Hyperparameters

Since our experiments are conducted on new datasets, we take particular care in tuning the hyper-parameters (e.g, learning rate, weight decay factor, TRADES' β, and c_{seg}) for both the baseline and our part-based models. For all models, we use grid search on the learning rate (0.1, 0.05, 0.02) and the weight decay (1×10^{-4}, 5×10^{-4}) during PGD adversarial training. For the part-based models, after obtaining the best learning rate and weight decay, we then further tune c_{seg} by sweeping values $0.1, 0.2, \ldots, 0.9$ and report on the model with comparable adversarial accuracy to the baseline. Results for other values of c_{seg} are included in Section E.2.

For TRADES, we reuse the best hyperparameters obtained previously and sweep a range of the TRADES parameter β, from 0.05 to 2, to generate the accuracy-robustness trade-off curve. However, we do not tune c_{seg} here and keep it fixed at 0.5 which puts equal weight on the classification and the segmentation losses. The same hyperparameter tuning strategy is used on both the baseline and our part models. We include our code along with the data preparation scripts in the supplementary material. Appendix B contains a detailed description of the experiment.

C Additional Descriptions and Results on the Part-Based Models

In this section, we elaborate on the part models and the loss functions we use to train them.

C.1 Bounding-Box Part Model

First, we compute a mask $\hat{F}$ for all foreground pixels. Then, the confidence score for each part mask, s_i, is the weighted average of the part logit mask $\hat{M}_i$ over all pixels, weighted by $\hat{F}$.

$$s_i = \frac{\sum_{h=1}^{H}\sum_{w=1}^{W} \hat{M}_i^{(h,w)} \cdot \hat{F}^{(h,w)}}{\sum_{h=1}^{H}\sum_{w=1}^{W} \hat{F}^{(h,w)}} \quad \text{where} \quad \hat{F}^{(h,w)} = \text{Sigmoid}\left(\sum_{k=1}^{K} \hat{M}_k^{(h,w)} - \hat{M}_0^{(h,w)}\right), \quad (1)$$

The other four bounding-box features are computed as follows:

$$c_i^1 = \sum_{h=1}^{H} p_i(h) \cdot h, \qquad \sigma_i^1 = \sqrt{\sum_{h=1}^{H} p_i(h) \cdot (h - c_i^1)^2} \quad (2)$$

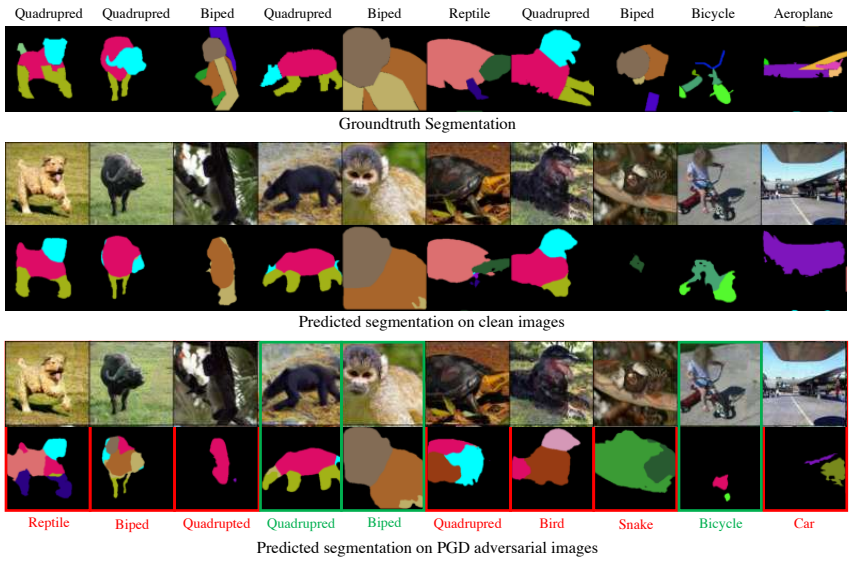


Figure 5: Visualization of the part segmentation predicted by the segmenter of the bounding-box part model adversarially trained on Part-ImageNet. All of the clean samples shown in the second and the third rows are correctly classified. The last two rows show PGD adversarial examples and their predictions. The misclassified (resp. correctly classified) samples are indicated with a red (resp. green) box, and the misclassified class labels are shown below in red (resp. green). The ground-truth labels and segmentation mask can be found on the top row.

$$c_i^2 = \sum_{w=1}^{W} p_i(w) \cdot w, \qquad \sigma_i^2 = \sqrt{\sum_{w=1}^{W} p_i(w) \cdot (w - c_i^2)^2} \tag{3}$$

$$\text{where} \quad p_i(h') = \frac{\sum_{w=1}^{W} \bar{M}_i^{(h',w)}}{\sum_{h=1}^{H}\sum_{w=1}^{W} \bar{M}_i^{(h,w)}}, \quad p_i(w') = \frac{\sum_{h=1}^{H} \bar{M}_i^{(h,w')}}{\sum_{h=1}^{H}\sum_{w=1}^{W} \bar{M}_i^{(h,w)}}, \tag{4}$$

and $\bar{M}^{(h,w)} = \text{Softmax}\left(\hat{M}^{(h,w)}\right)_{[1,\ldots,K]}$ is the softmax mask with the background channel removed. Note that $p_i(h)$ and $p_i(w)$ can be interpreted as the (normalized) density of the i-th object part in row h or column w, and $\hat{M}_i^{(h,w)}$ as its mass. Hence, c_i^1 and c_i^2 are simply the centroid of the i-th part. σ_i^1 and σ_i^2 measure the spread of mass so we use them as a proxy for the height and the width of the part.

C.2 Two-Headed Part Model

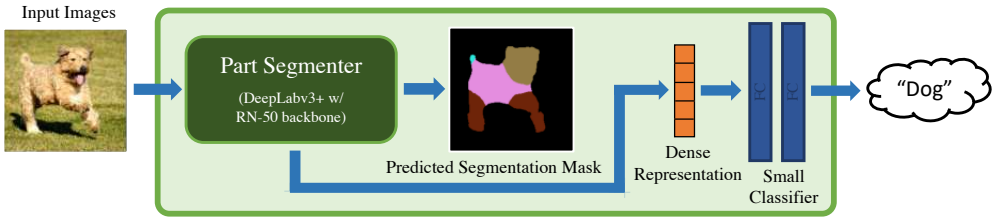


Figure 6: Diagram of the two-headed part model.

The two-headed part model uses a similar architecture to multi-task models with multiple heads. Here, there are two heads, one for segmentation and one for classification, sharing the same dense representation from the bottleneck layer of DeepLabv3+, as illustrated by Fig. 6. It is important to note that the two-headed part model does not explicitly use the predicted segmentation masks in classification. Instead, the classifier only sees the dense representation that will later be turned into the segmentation mask by the remaining layers of the segmenter. From an information-theoretic standpoint, the classifier of the two-headed part model should receive equal or more information

than the classifier in the bounding box or the downsampled part model. The difference is that this information is represented as dense vectors in the two-headed part model. However, in the other two models, the information is more human-interpretable and more compressed.

C.3 Pixel Part Model

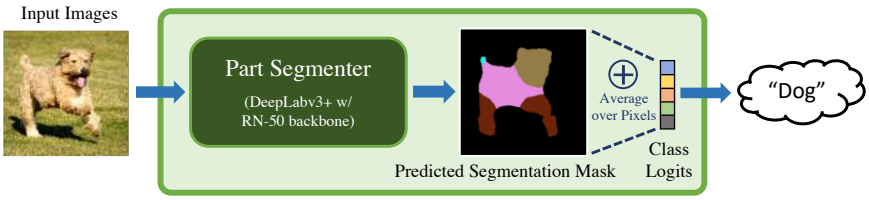


Figure 7: Diagram of the pixel part model.

The pixel part model is arguably the simplest among all of our part-based models. It does not use a small neural network classifier and involves only two simple steps. First, for each pixel, it sums together the part logits belonging to the same object class. In other words, the part segmentation mask is turned into the object segmentation mask, i.e., $\mathbb{R}^{(K+1)\times H\times W} \rightarrow \mathbb{R}^{C\times H\times W}$ where K and C are the numbers of parts and classes, respectively. Then, the object scores are averaged across all pixels in the segmentation mask to obtain the final class logits. This model is summarized in Fig. 7. It is also possible to treat the pixel part model as a specific case of the downsampled one where the convolution layer with a kernel size of 1×1 mimics the first step, and the classifier represents the average function in the second.

Importantly, averaging the logits across pixels means that the spatial information is ignored completely in the classification process. This eventually results in a minor reduction in the accuracy compared to the downsampled or the bounding-box model as shown in Appendix E.10 and Table 18. We do not recommend this model in practice, and it partially serves as an ablation study in our work.

C.4 Training Losses

Normal loss. These part-based models are trained end-to-end with a combined *segmentation-classification* loss, i.e., a weighted sum of two cross-entropy losses, one for the classification task and one for the pixel-wise segmentation task. A hyperparameter, $c_{\text{seg}} \in [0, 1]$, balances these two losses.

$$L_{\text{normal}}(x, y) = (1 - c_{\text{seg}}) \cdot L_{\text{cls}}(x, y) + c_{\text{seg}} \cdot L_{\text{seg}}(x, y) \tag{5}$$

where
$$L_{\text{cls}}(x, y) = L_{\text{CE}}\left(f(x), y\right) \tag{6}$$

and
$$L_{\text{seg}}(x, y) = \frac{1}{(K+1)HW} \sum_{k=0}^{K} \sum_{j=1}^{H\times W} L_{\text{CE}}\left(f_{\text{seg}}(x), M_k^{(j)}\right). \tag{7}$$

Adversarial loss. We construct an adversarial version of this loss, that measures susceptibility to adversarial examples. The adversary's goal is to maximize the classification loss (since it is the main task we evaluate on). The same adversarial example x^* generated from the classification loss is also used to compute the segmentation loss.

$$L_{\text{adv}}(x, y) = (1 - c_{\text{seg}}) \cdot L_{\text{cls}}(x^*, y) + c_{\text{seg}} \cdot L_{\text{seg}}(x^*, y) \tag{8}$$

where
$$x^* = \underset{z:\|z-x\|_p \leq \epsilon}{\arg\max} \; L_{\text{cls}}(z, y) \tag{9}$$

TRADES loss. We combine this with TRADES loss [Zhang et al., 2019] which introduces an extra term, a Kullback–Leibler divergence (D_{KL}) between the clean and the adversarial probability output.

$$L_{\text{TRADES}}(x, y) = (1 - c_{\text{seg}}) \cdot L_{\text{cls}}(x, y) + c_{\text{seg}} \cdot L_{\text{seg}}(x^*, y) + \beta \cdot D_{\text{KL}}\left(f(x), f(x^*)\right) \tag{10}$$

where
$$x^* = \underset{z:\|z-x\|_p \leq \epsilon}{\arg\max} \; D_{\text{KL}}\left(f(x), f(z)\right) \tag{11}$$

Table 2: Accuracy on the common corruption benchmark. We report a 95% confidence interval across different random seeds for training.

Model	Corruption Robustness
ResNet-50	82.3 ± 1.6
Downsampled Part Model	85.5 ± 0.8
Bounding-Box Part Model	**85.8** ± 0.7

Table 3: Accuracy on the background/foreground spurious correlation benchmark, with 95% CI across different random seeds.

Model	Spurious Correlation
ResNet-50	58.6 ± 4.2
Downsampled Part Model	**65.1** ± 0.8
Bounding-Box Part Model	**65.1** ± 2.1

Table 4: Accuracy on the shape-vs-texture bias benchmark. We report a 95% confidence interval across 10 different random seeds for training. Higher accuracy is better, suggesting that the model is less dependent on the texture features and more biased toward robust shape features.

Model	Shape-vs-Texture
ResNet-50	40.6 ± 1.8
Downsampled Part Model	44.7 ± 2.6
Bounding-Box Part Model	**45.7** ± 2.7

Table 5: Clean and adversarial accuracy of part-based models trained with and without the part segmentation labels compared to the ResNet-50 baseline. The improvement from the segmentation labels is highlighted.

Models	Seg. Labels?	Clean	Adv.
ResNet-50	N/A	74.7	37.7
Downsampled Part Model	No	76.9	39.6
	Yes	85.6	39.4
		(↑ 8.7)	(↓ 0.2)
Bounding-Box Part Model	No	78.1	39.9
	Yes	86.5	39.2
		(↑ 8.4)	(↓ 0.7)

D Non-Adversarial Robustness

Part-based models improve adversarial robustness, but what about robustness to non-adversarial distribution shift? We evaluate the models on three scenarios: common corruptions, foreground-vs-background spurious correlation, and shape-vs-texture bias. We generate benchmarks from Part-ImageNet following the same procedure as ImageNet-C [Hendrycks and Dietterich, 2019] for common corruptions, ImageNet-9 [Xiao et al., 2021] for foreground-vs-background spurious correlation, and Stylized ImageNet [Geirhos et al., 2019] for shape-vs-texture bias. For the common corruptions, the benchmark is composed of 15 corruption types and five severity levels. The spurious correlation benchmark is generated from a subset of foreground ("Only-FG") and background ("Only-BG-T") of ImageNet-9, filtering out classes not present in Part-ImageNet. Each foreground image is paired with a randomly chosen background image of another class. For shape-vs-texture bias, the data are generated by applying styles/textures using neural style transfer.

We train a ResNet-50 model and two part-based models using conventional training (not adversarial training) on clean Part-ImageNet samples. We tune the hyperparameters as described in Appendix B.1. For each benchmark, the best-performing set of hyperparameters is used to train 10 randomly initialized models to compute the confidence interval.

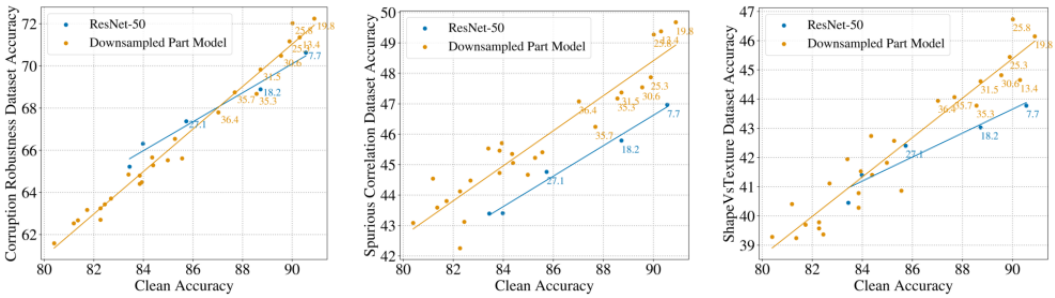
On all of the benchmarks, the part-based models outperform the baseline by 3–7 percentage points (see Tables 2, 3, and 4). The improvement over the ResNet-50 baseline is statistically significant (two-sample t-test, p-values below 10^{-6}). We note that these robustness gains do *not* come at a cost of clean accuracy as the clean accuracy of our part models is about 1% higher on average than that of the ResNet-50. This suggests that part-based models are more robust to common corruptions, better disentangle foreground and background information, and have higher shape bias compared to typical convolutional neural networks.

We also evaluate adversarially-trained models on the generalized robustness datasets, in addition to the normally trained ones reported earlier. Fig. 8 shows *the robust accuracy*[6] on the three benchmarks with respect to the clean accuracy of the models. The number next to each data point represents the adversarial accuracy, and due to the (adversarial) robustness-accuracy trade-off, the points on the top right corner generally have higher clean accuracy but lower adversarial accuracy.

[6]In this section, we will refer to the accuracy on the generalized robustness benchmarks as *the robust accuracy*. On the other hand, *the adversarial accuracy* still denotes the accuracy under adversarial attacks.

Table 6: Comparisons of the models on their generalized robustness. Higher is better. For each of the model types, we report two models, (A) and (B), trained with a different set of hyperparameters. Model (A) is the one with the highest accuracy on the shape-vs-texture benchmark, and model (B) is the one with the highest accuracy on both the spurious correlation and the common corruption benchmarks. All models are trained on Part-ImageNet without adversarial training.

Models	Shape-vs-Texture Bias	Spurious Correlation	Corruption Robustness
ResNet-50 (A)	**40.6** $\pm$ **1.8**	57.7 $\pm$ 2.0	81.5 $\pm$ 1.2
ResNet-50 (B)	40.5 $\pm$ 1.9	**58.6** $\pm$ **4.2**	**82.3** $\pm$ **1.6**
Downsampled Part Model (A)	**44.7** $\pm$ **2.6**	62.9 $\pm$ 2.1	84.3 $\pm$ 0.4
Downsampled Part Model (B)	43.4 $\pm$ 2.1	**65.1** $\pm$ **0.8**	**85.5** $\pm$ **0.8**
Bounding-Box Part Model (A)	**45.7** $\pm$ **2.7**	60.0 $\pm$ 2.0	82.0 $\pm$ 1.1
Bounding-Box Part Model (B)	44.4 $\pm$ 3.2	**65.1** $\pm$ **2.1**	**85.8** $\pm$ **0.7**



(a) Common Corruptions (b) Spurious Correlation (c) Shape vs Texture Bias

Figure 8: Plots of the robust accuracy on each of the three generalized robustness benchmark with respect to the clean accuracy. Each data point represents one adversarially trained model. The number next to each point is the adversarial accuracy (AutoAttack, $\epsilon = 8/255$). Generally, in the region where the clean accuracy is high, the part-based models outperform the ResNet-50 baseline on all accuracy metrics.

It is clear that there is a strong correlation between the clean accuracy and the robust accuracy on all three benchmarks. A similar trend is also observed in Taori et al. [2020]. In contrast, we do not find that adversarial training improves the common corruption robustness, spurious correlation robustness, or shape bias. Nonetheless, we emphasize that **the part models still outperform the ResNet-50 at almost all levels of clean accuracy across all types of robustness studied.**

Table 6 depicts the full results of the generalized robustness evaluations on the part-based models and the baseline. We conduct a hyperparameter search in order to find the best model for each of the benchmarks we test on. In this section, we report the robust accuracy of these models on all the benchmarks, not only the one that they perform the best in. Generally, we would have three rows per model architecture, one per dataset. However, interestingly, the best-performing models on the spurious correlation benchmark and the best-performing models on the corruption robustness benchmark are coincidentally the same models, i.e., model (B) in Table 6. On the other hand, the models (A) are only the best in the shape-vs-texture bias benchmark. This trend is consistent on the ResNet-50 as well as our part models. This phenomenon could be some sort of trade-off behavior. However, more experiments are needed to make further conclusions.

Table 7 shows a breakdown of the corruption robustness accuracy for each corruption type. This result confirms that the two part-based models outperform the ResNet-50 baseline on **all corruption types**, not only the mean. The bounding-box part model also achieves very slightly higher robust accuracy than the downsampled one across most of the corruption types.

Table 7: Accuracy for each corruption type from the common corruption benchmark, averaged across 10 random seeds during training. The highest number on each row is bold.

Corruption Type	ResNet-50	Downsampled Part Model	Bounding Box Part Model
Gaussian Noise	82.3	84.3	**84.7**
Shot Noise	82.4	84.1	**84.5**
Impluse Noise	80.8	83.6	**84.2**
Defocus Blur	81.7	86.1	**86.3**
Glass Blur	80.3	**84.0**	83.5
Motion Blur	79.1	**83.5**	**83.5**
Zoom Blur	67.4	70.1	**70.8**
Snow	75.1	80.1	**80.7**
Frost	78.8	83.4	**83.6**
Fog	86.7	90.5	**90.9**
Brightness	94.4	96.2	**96.4**
Contrast	71.0	74.5	**75.2**
Elastic Transform	88.2	92.3	**92.4**
Pixelate	92.6	94.6	**94.8**
JPEG Compression	93.4	95.1	**95.2**

Table 8: Clean and adversarial accuracy of *the ResNet-50 baseline* obtained over our hyperparameter sweep on Part-ImageNet.

Training Method	Learning Rate	Weight Decay	TRADES β	Clean	AutoAttack	PGD
Normal	0.1	5×10^{-4}	N/A	92.9	0.0	0.0
PGD	0.1	5×10^{-4}	N/A	74.7	37.7	43.3
		1×10^{-4}	N/A	69.2	36.5	40.7
	0.05	5×10^{-4}	N/A	76.4	36.6	42.2
		1×10^{-4}	N/A	74.8	34.4	40.3
	0.02	5×10^{-4}	N/A	73.0	33.6	39.6
		1×10^{-4}	N/A	70.5	32.0	37.6
TRADES	0.1	5×10^{-4}	0.05	91.6	1.2	2.2
			0.1	90.6	7.7	10.3
			0.15	89.6	13.0	16.0
			0.2	88.7	18.2	21.5
			0.3	87.9	22.9	26.0
			0.4	86.6	25.0	28.5
			0.5	85.7	27.1	29.8
			0.6	85.4	27.5	31.5
			0.7	84.7	29.0	32.2
			0.8	84.0	29.0	32.5
			0.9	83.8	30.6	34.2
			1.0	83.4	31.2	35.1
			2.0	74.4	30.5	34.9

E Additional Robustness Results

E.1 Hyperparameter Sweep Results

In this section, we include detailed results from our hyperparameter sweep on the ResNet-50 baseline (Table 8), the downsampled (Table 9), and the bounding-box part models (Table 10). The results suggest that all of the adversarially trained models are, to some degree, sensitive to the training hyperparameters, e.g., learning rate and weight decay. Nevertheless, the best setting is rather consistent across most of the models as well as the datasets, i.e., a learning rate of 0.1 and weight decay of 5×10^{-4}.

Table 9: Clean and adversarial accuracy of *the downsampled part models* obtained over our hyperparameter sweep on Part-ImageNet.

Training Method	Learning Rate	Weight Decay	c_{seg}	TRADES β	Clean	AutoAttack	PGD
Normal	0.1	5×10^{-4}	0.5	N/A	95.2	0.0	0.0
PGD	0.1	5×10^{-4}	0.5	N/A	83.9	39.9	45.3
		1×10^{-4}	0.5	N/A	79.1	39.6	45.8
	0.05	5×10^{-4}	0.5	N/A	85.1	38.8	44.7
		1×10^{-4}	0.5	N/A	82.3	37.5	43.7
	0.02	5×10^{-4}	0.5	N/A	80.4	36.9	43.4
		1×10^{-4}	0.5	N/A	82.3	35.1	42.4
TRADES	0.1	5×10^{-4}	0.5	0.05	90.9	19.8	23.8
				0.1	90.0	25.8	29.5
				0.2	89.6	30.6	34.7
				0.3	88.7	33.0	37.5
				0.4	88.4	33.7	37.7
				0.5	87.7	35.7	40.0
				0.8	85.3	37.2	41.2
				1.0	83.4	38.0	42.2

Table 10: Clean and adversarial accuracy of *the bounding-box part models* obtained over our hyperparameter sweep on Part-ImageNet.

Training Method	Learning Rate	Weight Decay	c_{seg}	TRADES β	Clean	AutoAttack	PGD
Normal	0.1	5×10^{-4}	0.5	N/A	95.4	0.0	0.0
PGD	0.1	5×10^{-4}	0.5	N/A	83.1	37.0	43.7
		1×10^{-4}	0.5	N/A	84.4	39.5	45.2
	0.05	5×10^{-4}	0.5	N/A	86.2	37.7	43.6
		1×10^{-4}	0.5	N/A	83.1	37.5	43.2
	0.02	5×10^{-4}	0.5	N/A	84.1	36.0	42.3
		1×10^{-4}	0.5	N/A	81.6	37.1	43.3
TRADES	0.1	5×10^{-4}	0.5	0.05	91.8	16.7	19.4
				0.1	90.8	24.1	27.5
				0.2	89.8	29.7	33.5
				0.3	89.6	32.5	36.2
				0.4	89.2	38.0	38.0
				0.5	87.8	35.3	39.3
				0.6	87.9	36.1	40.1
				0.8	86.1	37.4	41.5
				1.0	85.3	39.0	43.0

E.2 Effects of the c_{seg} hyperparameter

To test the effect of the segmentation loss, we train multiple part-based models with c_{seg} varied from 0 to 1. With c_{seg} closer to 1, the loss function prioritizes the pixel-wise segmentation accuracy. With c_{seg} closer to 0, less emphasis is put on the accuracy of segmentation masks. Fig. 9 shows the accuracy with respect to different c_{seg} values for both downsampled and bounding-box part models. It is, however, inconclusive whether the smaller or the larger value of c_{seg} is most preferable in this case. There is a vague trend that larger c_{seg} improves the clean accuracy but reduces the adversarial accuracy, exhibiting some form of trade-off. This overall trend can be explained by the fact that smaller c_{seg} places more weight on the adversarial classification loss and hence, improves the robustness.

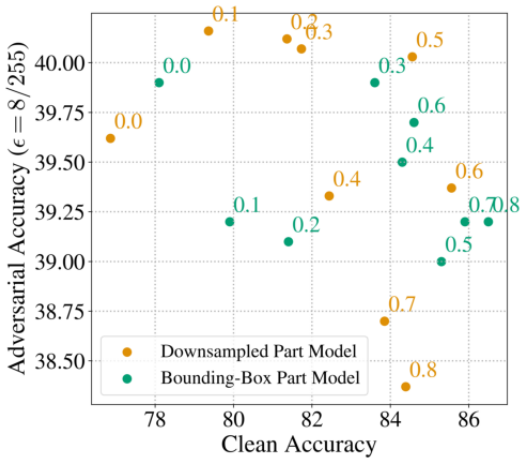


Figure 9: Clean and adversarial accuracy of the downsampled (orange) and the bounding-box (green) part models trained on Part-ImageNet. The number on the top right of each data point indicates the value of c_{seg} that model is trained with. All models are trained with a learning rate of 0.1 and weight decay of 5×10^{-4}.

Table 11: Adversarial accuracy of our part-based models at different values of ϵ. This table shows that the adversarial accuracy does reach zero as ϵ becomes larger which confirms that our part models are unlikely to experience the gradient obfuscation.

Datasets	Part-Based Models	Adversarial Accuracy			
		$\epsilon = 8/255$	$\epsilon = 16/255$	$\epsilon = 24/255$	$\epsilon = 32/255$
Part-ImageNet	Downsampled	39.4	13.6	3.5	1.1
	Bounding-Box	39.2	12.6	3.9	1.7
Cityscapes	Downsampled	70.2	24.3	2.8	0.4
	Bounding-Box	69.9	16.6	0.9	0.0
PASCAL-Part	Downsampled	38.5	24.8	8.3	1.8
	Bounding-Box	38.5	20.1	4.3	0.7

E.3 Effects of Part Segmentation Labels vs Architecture

Where does the robustness improvement come from? Does it come from the additional information provided by part annotations, or from the new architecture we introduce?

To answer these questions, we train part-based models on Part-ImageNet without using the part segmentation labels while keeping the model architecture and hyperparameters fixed (i.e., setting L_{seg} in Eqn. 5 to zero). We found that most of the improvement comes from the additional supervision provided by part annotations. In particular, the architecture provides 2–4 percentage points of improvement over ResNet-50, while the additional supervision provides another 8–9 points of improvement in clean accuracy (see Table 5). This experiment confirms that most of the gain comes from the additional information provided by fine-grained segmentation labels.

E.4 Optimality of the Attacks

Gradient Obfuscation. We do not believe our models suffer from gradient obfuscation. First, our models do not use any non-differentiable operations or randomization; they use only standard neural network layers.

Second, we conduct a sanity check suggested by Carlini et al. [2019] by making sure that a simple PGD attack can reduce the accuracy close to zero when the perturbation norm increases. Our new experiment, reported in Table 11, confirms this as the adversarial accuracy of our part-based models does drop to $< 2\%$ at $\epsilon = 32/255$.

Table 12: Effects of the c_{seg} parameter in the loss function of PGD attack on the Downsample part model trained on Part-ImageNet. We emphasize that this is the value of c_{seg} used during the evaluation attack, not during adversarial training.

Values of c_{seg} in PGD Attack	PGD Accuracy
0 (normal PGD)	45.4
0.1	45.9
0.3	48.0
0.5	50.4
0.7	53.7
0.9	57.5

Table 13: Adversarial accuracy measured by the two-staged attack on our part-based models compared to PGD and AutoAttack (AA). "MC" denotes the most-confident strategies.

Datasets	Part-Based Models	Adversarial Accuracy					
		PGD	AA	Untargeted	Random	MC (Random)	MC (Sorted)
Part-ImageNet	Downsampled	45.4	39.4	45.1	44.0	47.5	47.5
	Bounding-Box	45.7	39.2	45.3	43.3	47.3	53.4
Cityscapes	Downsampled	73.8	70.2	75.4	75.5	75.4	75.5
	Bounding-Box	73.4	69.9	74.7	74.8	74.7	74.6
PASCAL-Part	Downsampled	40.6	38.5	40.3	39.9	44.6	44.6
	Bounding-Box	40.6	38.5	40.6	41.0	43.9	43.2

Third, we have also experimented with decision-based black-box attacks that do not rely on gradient information. We use AutoAttack [Croce and Hein, 2020], which incorporates Square Attack [Andriushchenko et al., 2020] which does not rely on gradients and only uses the output scores. We also use the state-of-the-art ℓ_∞-attack, RayS [Chen and Gu, 2020], to evaluate our Downsampled part model on the Part-ImageNet dataset. RayS manages to reduce the accuracy to 71.0 (at 10k steps), which is still much higher than that achieved by the PGD attack and AutoAttack (45.4 and 39.4). This confirms that the non-gradient attacks are not better than the gradient-based ones, suggesting that there is no gradient obfuscation problem.

Single-staged attack. We have experimented with multiple ways to attack the part models, including attempts to fool the segmenter by using both losses in the attack objective. However, these alternatives actually decrease the attack success rate. In our experiments, using only classification loss always yields the strongest attack.

In particular, we consider PGD attack with an objective that is a linear combination of the classification loss and the segmentation loss, i.e., $L = (1 - c_{\text{seg}})L_{\text{clf}} + c_{\text{seg}}L_{\text{seg}}$, as in Eqn. 7. Table 12 reports the adversarial accuracy under this attack with varying values of c_{seg}. This shows that using the segmentation loss does not improve the attack. In fact, a larger c_{seg} (more weight on the segmentation loss) actually results in a worse attack.

Two-staged attack. Since we previously found that optimizing over both losses at the same time results in a worse attack, we separate the attack into two stages and make sure that the second stage only optimizes over the classification loss. The difference now lies in the first stage which we use to generate a "guiding sample" to initialize the second attack by focusing on fooling the segmenter first. We experiment with four strategies for the first-stage attack:

1. *Untargeted*: Maximize the loss of the segmenter directly with an untargeted PGD.
2. *Random*: Pick a random target mask from a random incorrect class and run PGD to fool the segmenter into predicting this target mask.
3. *Most-confident (random)*: Similar to the random strategy, but instead of sampling from a random class, only sample target masks from the most-confident class predicted by the part model, excluding the ground-truth class.

Table 14: Comparison of accuracy of part models trained using different types of auxiliary labels. The part bounding-box and centroid models are PGD adversarially trained. We select the part segmentation model with similar accuracy from Section 4 for comparison.

Types of Labels	Part-ImageNet		Cityscapes		PASCAL-Part	
	Clean	Adv.	Clean	Adv.	Clean	Adv.
Segmentation	85.6	39.4	94.8	70.2	77.3	34.5
Bounding Boxes	84.1	39.7	95.4	69.1	66.2	33.5
Centroids	82.6	39.7	94.0	70.9	62.9	33.5
ResNet-50	74.7	37.7	79.5	68.4	54.0	29.1

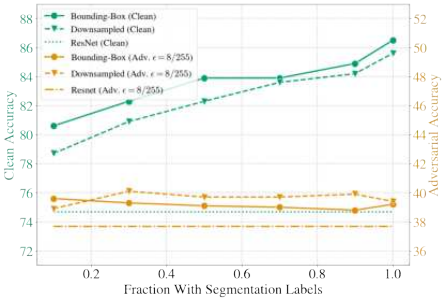


Figure 10: Performance of the part models when only a fraction of training samples are accompanied by a segmentation label.

4. *Most-confident (sorted)*: Similar to the most-confident (random) strategy, but instead of randomly choosing the masks, we run each mask in the test set through the classifier and choose the ones that the model assigns the highest score/confidence to the target class.

We note that similarly to PGD, we repeat all the two-staged attacks five times with different random seeds and select only the best out of five. This means that the first stage of the attacks uses five different target masks, apart from the untargeted strategy, and produces five different initialization points. Table 13 demonstrates that the two-staged attacks are about as effective as the normal PGD. The untargeted and the random strategies usually perform the best and can be slightly ($\sim$1% lower adversarial accuracy) better than the normal PGD attack. Nevertheless, no attack beats AutoAttack in any setting. This suggests that it is likely sufficient to use the AutoAttack alone for evaluation.

E.5 Cheaper Forms of Auxiliary Labels

We additionally explore two labeling strategies for reducing labelling costs: (1) bounding box segmentations for each part, or (2) keypoints or centroids for each part (Fig. 11, Appendix E.5).[7] These annotations provide less precise spatial information about each part but are much faster to label. Bounding-box labels are nearly as effective as segmentation masks on Part-ImageNet and Cityscapes (within $\sim$1% difference in accuracy; see Table 14). However, the difference is much larger on PASCAL-Part where the clean accuracy is 11% lower. Models trained on centroid labels perform slightly worse than the ones trained on bounding-box labels, which is unsurprising as centroids are even more coarse-grained. Nonetheless, all part models trained on any kind of part label still outperform the ResNet-50 baseline. We hope our work draws attention to the opportunity for stronger robustness through rich supervision and stimulates research into reducing the labeling cost. Below we explain how the labels are generated as well as the model training in detail.

Bounding-box labels. The part bounding boxes are generated directly from the part segmentation by drawing a tight box around all the pixels that belong to each part. Fig. 11 provides examples of the bounding box labels from the Part-ImageNet dataset. We want to keep the segmenter unchanged so we train the Downsampled part models with unmodified L_{seg}, as described in Eqn. 7, on the new bounding-box labels. We note that our bounding-box labels are still pixel-wise masks unlike the typical bounding boxes used in the object detection task. In practice, it is likely more efficient to replace the segmenter with an object detection model that outputs bounding boxes directly.

Centroid labels. Similarly to the bounding boxes, the centroid labels are also directly derived from the segmentation mask. We go through the same calculation in Eqn. 2 to generate the centroids from the ground-truth, instead of predicted, segmentation masks. Here, we train the bounding-box part model on the centroid labels, but instead of calculating the segmentation loss, we compute the loss directly on the dense features excluding the standard deviations. More precisely, the loss function,

[7] Here, we refer to the labels provided for training. This should not be confused with the architecture of the Bounding-Box Part Model.

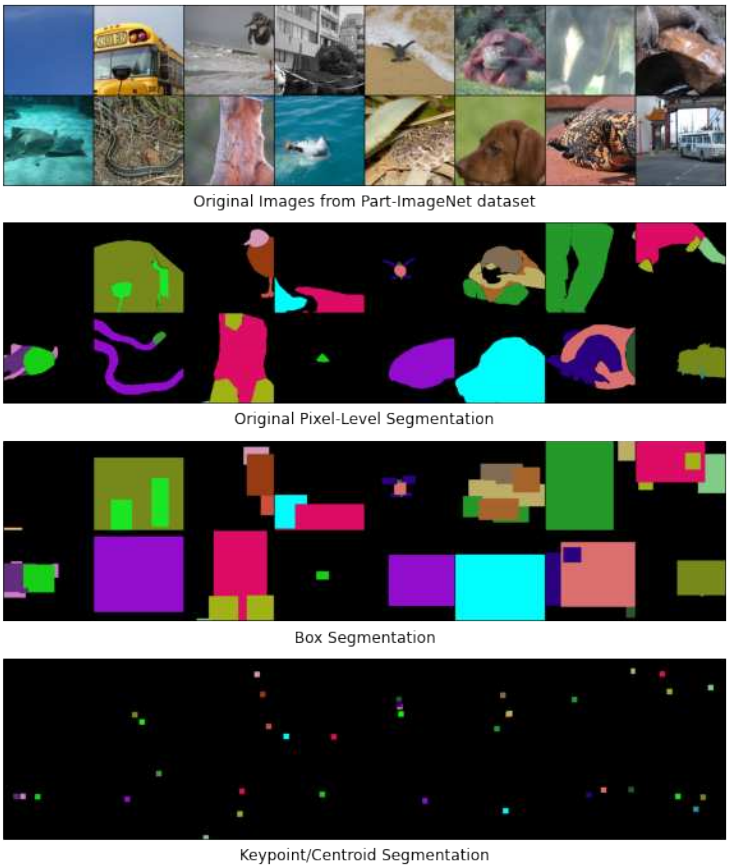


Figure 11: Random examples of part bounding-box labels and centroid labels used in the experiment in Appendix E.5.

L_{cen}, can be written as follows:

$$L_{\text{cen}} = \frac{1}{K}\sum_{k=1}^{K} \left[(c_k^1(f_{\text{seg}}(x)) - c_k^1(M_k))^2 + (c_k^2(f_{\text{seg}}(x)) - c_k^2(M_k))^2\right] \tag{12}$$

$$+ L_{\text{CE}}\left(\frac{\sum_{k\in c}\sum_{j=1}^{H\times W} f_{\text{seg}}(x)}{\sum_{c=1}^{C}\sum_{k\in c}\sum_{j=1}^{H\times W} f_{\text{seg}}(x)}, y\right). \tag{13}$$

The first term is the mean squared error of the predicted centroids and the ground truth. The second ensures that the segmenter predicts masks of the correct class. For this, we use the cross-entropy loss with the logits being the sum of pixel-wise predictions across all parts of each object class.

E.6 Training with Fewer Part Segmentation Labels

The main limitation of our approach is the extra labeling cost to obtain part segmentation labels. We investigate how the performance of our models changes when fewer part annotations are available. We train part models with the same number of training samples and class labels but a reduced number of segmentation labels, so some (10–90%) of training samples have both class and segmentation labels while the rest have *only* the class label. As expected, the clean accuracy degrades when the model receives less part-level supervision; see Fig. 10. Adversarial accuracy, however, remains more or less the same, and all of the part models still outperform the ResNet-50. We leave it to future work to improve performance when only a small fraction of training samples are part-annotated.

Table 15: Clean and adversarial accuracy of the downsampled part models trained with *object-level segmentation labels instead of part-level*. The model is adversarially trained (PGD) on Part-ImageNet with different values of c_{seg}. The adversarial accuracy is computed by AutoAttack and PGD attack.

Models	c_{seg}	Clean Accuracy	AutoAttack Accuracy	PGD Accuracy
Downsampled Part Model (Best)	-	85.6	39.4	45.4
Downsampled Part Model w/ Object Segmentation	0.1	**83.5**	39.2	45.4
	0.3	81.3	37.9	44.2
	0.5	82.8	**39.3**	**45.5**
	0.7	81.6	38.0	45.1
	0.9	82.0	37.9	44.9

Table 16: Accuracy on the three generalized robustness benchmarks comparing the Downsampled part models with and without the background channel.

Models	Common Corruptions	Background-vs-Foreground	Shape-vs-Texture
ResNet-50	82.3 ± 1.6	58.6 ± 4.2	40.6 ± 1.8
Downsampled Part Models (w/ Background)	85.5 ± 0.8	65.1 ± 0.8	44.7 ± 2.6
Downsampled Part Models (w/o Background)	85.5 ± 1.8	64.2 ± 2.2	45.1 ± 2.3

E.7 Part Segmentation vs Object Segmentation

We conduct an ablation study to test whether the part-level annotation is necessary to improve the adversarial training. Can it be substituted with an object-level annotation which is cheaper to label? To answer this question, we train downsampled "part" models using *the object-segmentation labels* instead of the part-level annotation. Table 15 clearly indicates that the models trained on the object-level annotation achieve lower clean and adversarial accuracy compared to ones trained on the part-level annotation. This experiment suggests that training with object segmentation does improve adversarial training compared to the baseline, but using part segmentation can achieve even better results. Intuitively, the part annotation is more fine-grained and contains more information than the object one. So it is likely that stronger learning signals lead to higher robustness.

E.8 Effects of Background Removal

We repeat the same experiments, measuring both adversarial and generalized robustness, on the Downsampled part models that remove the background. Specifically, we drop the background channel of the predicted segmentation mask by the segmenter before passing it to the second-stage classifier.

In summary, our results show that whether the predicted background channel is included or not has little effect on the accuracy. The model without background has 0.8% lower clean accuracy and the same adversarial accuracy as the one with the background channel. The result on the generalized robustness benchmarks in Table 16 also portrays a similar story: the Downsampled part models with and without the background perform similarly (within margins of error) but are still clearly better than ResNet-50. This experiment suggests that the second-stage classifier can learn to ignore the background pixels automatically. So there is no clear benefit to dropping them.

E.9 Effects of the Downsampled Size

Table 17 shows the performance of the downsampled part model when the output size of the pooling layer changes. Across all the sizes from 1 to 128, both the clean and the adversarial accuracy barely change; the gap between the largest and the smallest numbers is under 1.3 percentage points. This suggests that the performance of the downsampled part model is insensitive to the choice of the downsampling output size. We use the downsampling size of 4×4 throughout this paper, but almost any other number can be used since the difference is not significant.

Table 17: Clean and adversarial accuracy of the downsampled part models trained on Part-ImageNet with different values of downsampling output sizes. All of the models here are trained with a learning rate of 0.1, weight decay of 5×10^{-4}, and c_{seg} of 0.5. The adversarial accuracy is computed by AutoAttack and PGD attacks.

Downsampling Output Sizes	Clean Accuracy	AutoAttack Accuracy	PGD Accuracy
1×1	83.9	39.9	**45.9**
2×2	84.0	39.4	45.5
4×4	83.9	39.9	45.3
8×8	83.0	39.5	**45.9**
32×32	83.0	38.7	45.4
128×128	**84.3**	**40.0**	45.7

Table 18: Clean and adversarial accuracy of the part model variants adversarially trained (PGD) on Part-ImageNet with different values of c_{seg}. The adversarial accuracy is computed by AutoAttack and PGD attack ($\epsilon = 8/255$). For comparison, we add the first two rows for the two best part models we reported in the main paper. The highest accuracy in each column of each model is bold.

Models	c_{seg}	Clean Accuracy	AutoAttack Accuracy	PGD Accuracy
Downsampled Part Model (Best)	-	85.6	39.4	45.4
Bounding-Box Part Model (Best)	-	86.5	39.2	45.7
Two-Headed Part Model	0.1	**86.1**	38.9	44.7
	0.3	84.6	38.2	44.5
	0.5	85.4	39.2	44.6
	0.7	84.6	38.9	44.7
	0.9	85.7	**39.4**	**44.9**
Pixel Part Model	0.1	**84.5**	39.6	45.4
	0.3	83.0	38.5	45.1
	0.5	83.1	37.8	45.0
	0.7	83.3	**39.7**	**46.0**
	0.9	84.3	39.6	45.5

E.10 Adversarial Robustness Results on the Remaining Part Models

In this section, we include the robustness results on the other two part-based models we omit from the main text, i.e., the two-headed and the pixel part models. We report the accuracy of the models trained with five different values of c_{seg} for completeness and for displaying a minor trade-off between the clean and the adversarial accuracy. However, comparing the best models alone would be sufficient.

Table 18 suggests that the two-headed part models perform similarly to the downsampled variant and slightly worse than the bounding-box one when all of them are adversarially trained with PGD. On the other hand, the pixel part models have consistently lower accuracy than the other part models by roughly 1–2 percentage points. This result confirms our hypothesis on the importance of the spatial information as mentioned in Section 3.1 as well as Appendix C.3.

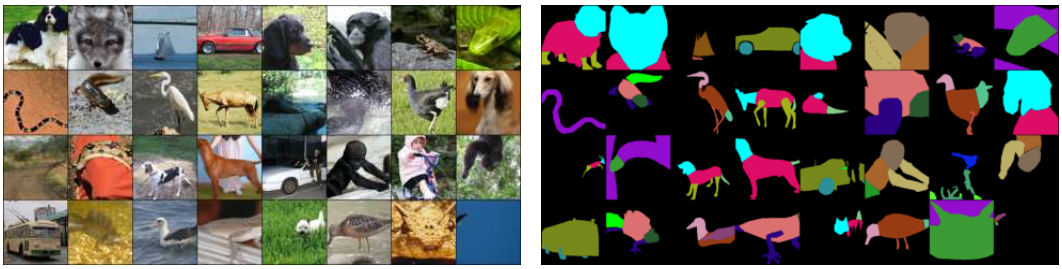
E.11 Feeding Input Images to the Part Model

In Section 3.1, we suggest that the classifier stage of the part models should not see the input image directly. We hypothesize that doing so opens up an opportunity for the attacker to bypass the more robust segmenter and influence the small and less robust classifier. This essentially defeats the purpose of the segmentation and the part model overall. However, there is also a counterargument to this hypothesis. In theory, if the model is fed with both the image and the predicted segmentation mask, it strictly receives more information. When adversarially trained, the model can then learn to ignore the image if it is deemed non-robust. Hence, this model should be strictly better or at least the same as the one that sees only the segmentation mask.

To find out which hypothesis holds, we create a variant of the downsampled part model by concatenating the input image to the predicted segmentation mask before being fed to the classifier stage. We then compare this model to the original downsampled part model. The empirical results support our

Table 19: Clean and adversarial accuracy of *the downsampled part models with concatenated input images* (see Appendix E.11). The model is adversarially trained (PGD) with different values of c_{seg} on Part-ImageNet. The adversarial accuracy is computed by AutoAttack and PGD attack ($\epsilon = 8/255$).

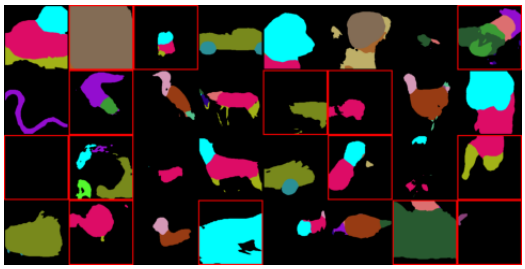
Models	c_{seg}	Clean Accuracy	AutoAttack Accuracy	PGD Accuracy
Downsampled Part Model (Best)	-	85.6	39.4	45.4
Downsampled Part Model w/ Concat. Input	0.1	82.2	37.7	44.4
	0.3	**82.6**	38.7	45.0
	0.5	79.9	38.7	44.8
	0.7	76.9	39.1	**45.3**
	0.9	72.7	**39.5**	44.1



(a) Benign test samples.

(b) Groundtruth segmentation.

(c) Predicted segmentation from clean samples.



(d) Predicted segmentation from PGD-attack samples.

Figure 12: Visualization of *the downsampled part model on Part-ImageNet*: (a) randomly selected clean test samples, (b) the corresponding groundtruth segmentation mask, (c) their predicted segmentation mask from the segmenter, and (d) the predicted segmentation mask when the samples are perturbed by PGD attack ($\epsilon = 8/255$). Segmentation masks corresponding to misclassified samples are indicated by a red box.

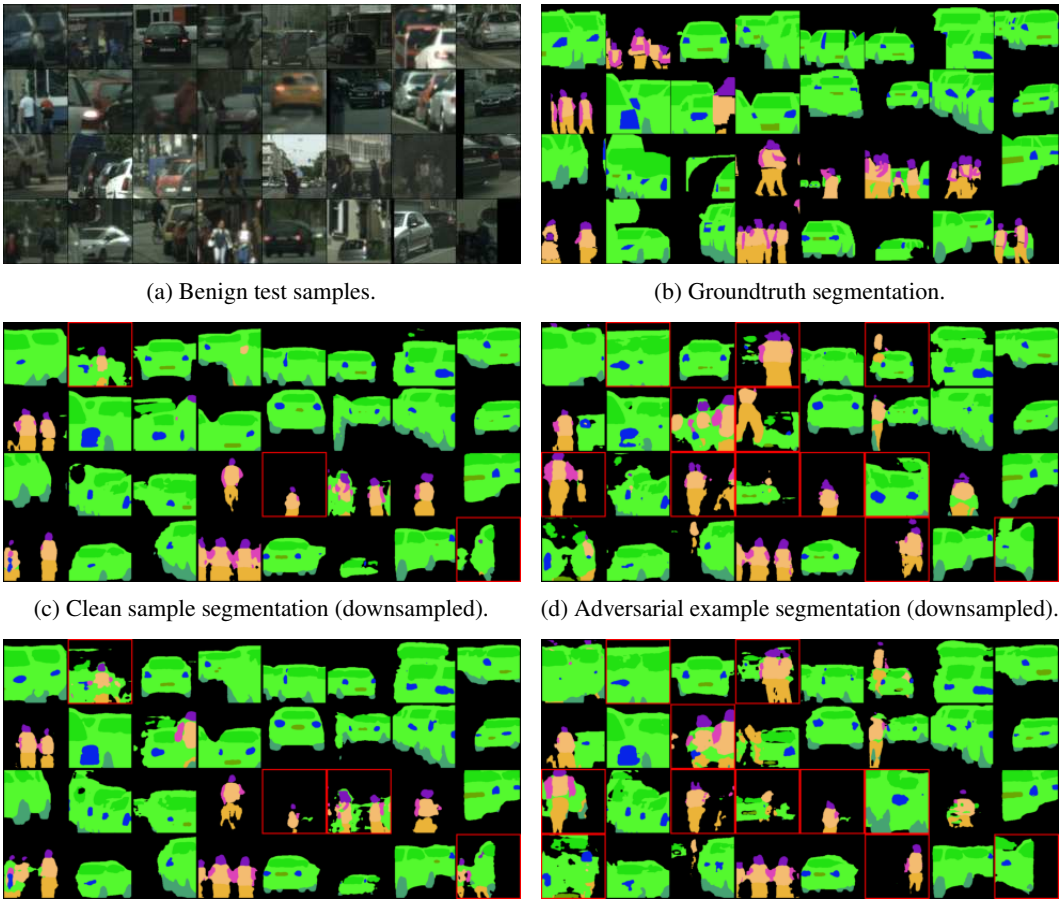
hypothesis. Table 19 shows that this input-concatenated downsampled part model performs slightly worse compared to the original version. We leave it to future work to unveil the underlying reasons that make the model less robust when more information is presented to it.

E.12 Additional Visualization of the Part Models

We provide additional visualization of the outputs of our part-based models on all three datasets. Fig. 12 shows a similar visualization to Fig. 5 but for the downsampled part model. The same visualization for Cityscapes (resp. PASCAL-Part) on the downsampled and the bounding-box part models can be found in Fig. 13 (resp. Fig. 14).

Apart from the ones trained on PASCAL-Part, our part-based models segment the object part fairly well even though some amount of detail and small parts are sometimes missed. In most of the misclassified samples, the predicted segmentation masks are also incorrect. This is particularly true for the PGD adversarial images. This observation qualitatively confirms that the classifier stage of the part model depends on and agrees with the segmentation mask as expected.

We suspect that the poor prediction of the segmentation on the PASCAL-Part dataset may be attributed to the small number of training samples. PASCAL-Part has about one order of magnitude fewer



(a) Benign test samples.

(b) Groundtruth segmentation.

(c) Clean sample segmentation (downsampled).

(d) Adversarial example segmentation (downsampled).

(e) Clean sample segmentation (bounding-box).

(f) Adversarial example segmentation (bounding-box).

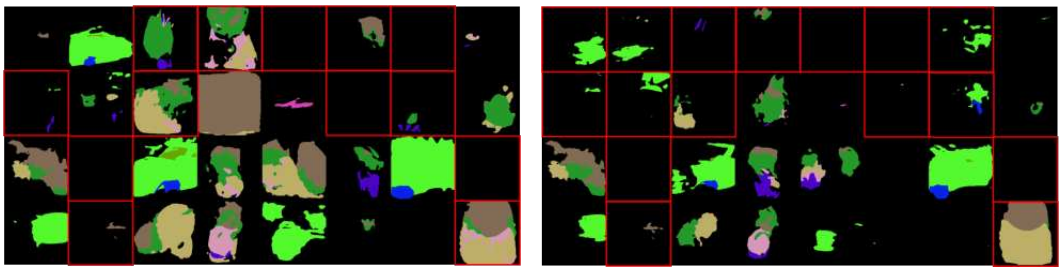
Figure 13: Visualization of the part model trained on *Cityscapes*: (a) randomly selected clean test samples, (b) the corresponding groundtruth segmentation mask. (c) and (d) are the predicted segmentation mask from the downsampled part model on clean and adversarial samples (PGD attack with $\epsilon = 8/255$), respectively. (e) and (f) are the segmentation masks from the bounding-box model. Segmentation masks corresponding to misclassified samples are indicated by a red box.

training samples compared to the other two datasets. Nevertheless, the segmentation labels still prove to be very helpful in improving the adversarial training, potentially also due to the fact that the number of training data is small. One interesting future direction is to study the relationship between the numbers of class labels and segmentation labels with respect to robustness.

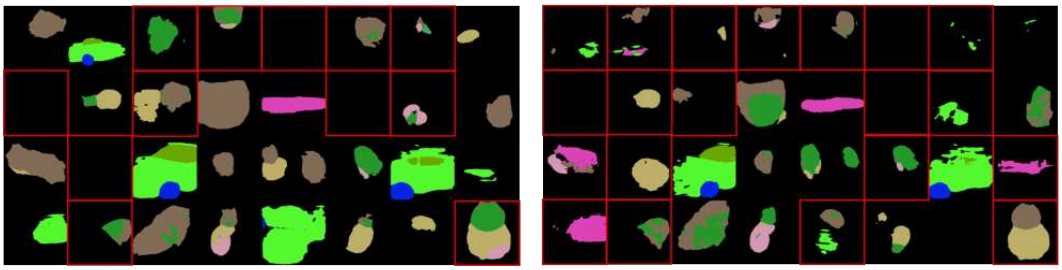
F Societal Impact

Our work focuses on improving the adversarial robustness of neural networks with the goal of creating secure and reliable models. We strictly propose a new "defense" technique and do not contribute to any attack algorithm. We believe that our work will benefit not only the research community but also machine learning practitioners and eventually, society overall. We hope that our work will be extended to improve the effectiveness of adversarial training in practice, leading to broader adoption of deep learning as well as preventing potential vulnerabilities and failures in the future.

(a) Benign test samples. (b) Groundtruth segmentation.



(c) Clean sample segmentation (downsampled). (d) Adversarial example segmentation (downsampled).



(e) Clean sample segmentation (bounding-box). (f) Adversarial example segmentation (bounding-box).

Figure 14: Visualization of the part model trained on *PASCAL-Part*: (a) randomly selected clean test samples, (b) the corresponding groundtruth segmentation mask. (c) and (d) are the predicted segmentation mask from the downsampled part model on clean and adversarial samples (PGD attack with $\epsilon = 8/255$), respectively. (e) and (f) are the segmentation masks from the bounding-box model. Segmentation masks corresponding to misclassified samples are indicated by a red box.

G X-Risk Sheet

G.1 Long-Term Impact on Advanced AI Systems

In this section, please analyze how this work shapes the process that will lead to advanced AI systems and how it steers the process in a safer direction.

1. **Overview.** How is this work intended to reduce existential risks from advanced AI systems?
   Answer: Our work proposes a new technique to improve the robustness of neural networks, both adversarial and non-adversarial. We show a simple architecture that can effectively utilize richer annotations in a form of part labels. Our technique is compatible with other advancements in adversarial training but is also beneficial without it.

2. **Direct Effects.** If this work directly reduces existential risks, what are the main hazards, vulnerabilities, or failure modes that it directly affects?
   Answer: This work directly attempts to reduce the risk of evasion attacks, or more generally vulnerability to adversaries, on machine learning models. Our models also exhibit increased robustness to common corruptions, spurious correlation, and texture biases without any data augmentation or specialized training procedure. Common corruptions and spurious correlation (e.g., objects appear with rare a background or other correlated but not task-related features)

represent the tail distribution. Spurious correlation and texture biases can also be categorized as misspecification or misalignment between the intended objective and how the model actually solves the task.

3. **Diffuse Effects.** If this work reduces existential risks indirectly or diffusely, what are the main contributing factors that it affects?
   Answer: Our work mostly involves the risks directly as mentioned above.
4. **What's at Stake?** What is a future scenario in which this research direction could prevent the sudden, large-scale loss of life? If not applicable, what is a future scenario in which this research direction be highly beneficial?
   Answer: We hope that our work will lead to models that are robust to attacks and better aligned with human perception systems in the future. The focus of our work is on the image recognition tasks so we hope to bring safety to any system that uses some forms of images as inputs. Important examples include self-driving vehicles and robotics that interact with humans directly.
5. **Result Fragility.** Do the findings rest on strong theoretical assumptions; are they not demonstrated using leading-edge tasks or models; or are the findings highly sensitive to hyperparameters? ☐
6. **Problem Difficulty.** Is it implausible that any practical system could ever markedly outperform humans at this task? ☒
7. **Human Unreliability.** Does this approach strongly depend on handcrafted features, expert supervision, or human reliability? ☐
8. **Competitive Pressures.** Does work towards this approach strongly trade off against raw intelligence, other general capabilities, or economic utility? ☐

G.2 Safety-Capabilities Balance

In this section, please analyze how this work relates to general capabilities and how it affects the balance between safety and hazards from general capabilities.

9. **Overview.** How does this improve safety more than it improves general capabilities?
   Answer: Our work is intended to explore a new direction for improving adversarial robustness. We believe that the usage of the part-level annotation and the architecture we introduce are particularly necessary for pushing the current state-of-the-art robust training methods. This is because adversarial training is believed to require more data compared to normal training [Schmidt et al., 2018]. This means that it may be impossible with scaling up alone to make adversarial training reach the same level of generalization as the non-robust counterpart. At least, it could take many more years before we bridge this gap by scaling things up alone.
10. **Red Teaming.** What is a way in which this hastens general capabilities or the onset of x-risks?
    Answer: No, our work essentially proposes defenses to prevent future exploitation.
11. **General Tasks.** Does this work advance progress on tasks that have been previously considered the subject of usual capabilities research? ☐
12. **General Goals.** Does this improve or facilitate research towards general prediction, classification, state estimation, efficiency, scalability, generation, data compression, executing clear instructions, helpfulness, informativeness, reasoning, planning, researching, optimization, (self-)supervised learning, sequential decision making, recursive self-improvement, open-ended goals, models accessing the Internet, or similar capabilities? ☐
13. **Correlation With General Aptitude.** Is the analyzed capability known to be highly predicted by general cognitive ability or educational attainment? ☐
14. **Safety via Capabilities.** Does this advance safety along with, or as a consequence of, advancing other capabilities or the study of AI? ☒

G.3 Elaborations and Other Considerations

15. **Other.** What clarifications or uncertainties about this work and x-risk are worth mentioning?
    Answer: Our proposed technique increases the supervision and the data labeling cost compared to other works in the domain of image recognition as well as adversarial machine learning in general. This presents a trade-off between the cost and the robustness our models achieve. Regardless, we

also conduct a series of experiments to reduce such labeling costs. We believe that an interesting future direction is to completely eliminate this cost altogether while also keeping the benefits.