Track 1:

Ensemble everything everywhere: Multi-scale aggregation for adversarial robustness

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Adversarial examples pose a significant challenge to the robustness, reliability and alignment of deep neural networks. We propose a novel, easy-to-use approach to achieving high-quality representations that lead to adversarial robustness through the use of multi-resolution input representations and dynamic self-ensembling of intermediate layer predictions. We demonstrate that intermediate layer predictions exhibit inherent robustness to adversarial attacks crafted to fool the full classifier, and propose a robust aggregation mechanism based on Vickrey auction that we call *CrossMax* to dynamically ensemble them. By combining multi-resolution inputs and robust ensembling, we achieve significant adversarial robustness on CIFAR-10 and CIFAR-100 datasets without any adversarial training or extra data, reaching an adversarial accuracy of ≈72% (CIFAR-10) and ≈48% (CIFAR-100) on the RobustBench AutoAttack suite ($L_\infty = 8/255$) with a finetuned ImageNet-pretrained ResNet152. This represents a result comparable with the top three models on CIFAR-10 and a +5 % gain compared to the best current dedicated approach on CIFAR-100. Adding simple adversarial training on top, we get ≈78% on CIFAR-10 and ≈51% on CIFAR-100, improving SOTA by 5 % and 9 % respectively and seeing greater gains on the harder dataset. We validate our approach through extensive experiments and provide insights into the interplay between adversarial robustness, and the hierarchical nature of deep representations. We show that simple gradient-based attacks against our model lead to human-interpretable images of the target classes as well as interpretable image changes. As a byproduct, using our multi-resolution prior, we turn pre-trained classifiers and CLIP models into controllable image generators and develop successful transferable attacks on large vision language models.

1 Introduction

Adversarial examples in the domain of image classification are small, typically human-imperceptible perturbations P to an image X that nonetheless cause a classifier, $f : X \rightarrow y$, to misclassify the perturbed image $X + P$ as a target class t chosen by the attacker, rather than its correct, ground truth class. This is despite the perturbed image $X + P$ still looking clearly like the ground truth class to a human, highlighting a striking and consistent difference between machine and human vision (first described by Szegedy et al. [2013]). Adversarial vulnerability is ubiquitous in image classification, from small models and datasets [Szegedy et al., 2013] to modern large models such CLIP [Radford et al., 2021], and successful attacks transfer between models and architectures to a surprising degree [Goodfellow et al., 2015] without comparable transfer to humans.

Submitted to AdvML-Frontiers'24: The 3rd Workshop on New Frontiers in Adversarial Machine Learning@NeurIPS'24, Vancouver, CA. Do not distribute.

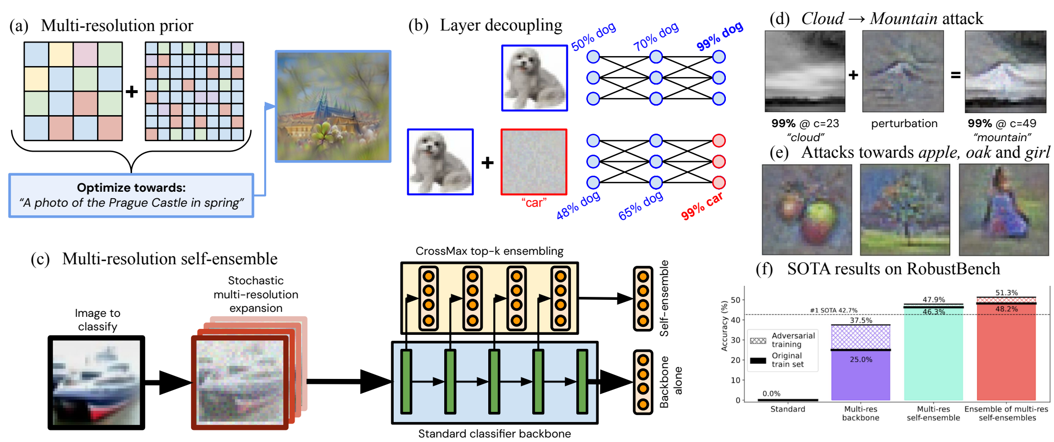


Figure 1: We use a multi-resolution decomposition (a) of an input image and a partial decorrelation of predictions of intermediate layers (b) to build a classifier (c) that has, by default, adversarial robustness comparable or exceeding state-of-the-art (f), even without any adversarial training. Optimizing inputs against it leads to interpretable changes (d) and images generated from scratch (e).

We hypothesize that the existence of adversarial attacks is due to the significant yet subtle mismatch between what humans do when they classify objects and how they learn such a classification in the first place (the *implicit* classification function in their brains), and what is conveyed to a neural network classifier explicitly during training by associating fixed pixel arrays with discrete labels (the learned machine classification function).

In this paper, we take a step towards aligning the implicit human and explicit machine classification functions, and consequently observe very significant gains in adversarial robustness against standard attacks as a result of a few, simple, well-motivated changes, and without any explicit adversarial training. While, historically, the bulk of improvement on robustness metrics came from adversarial training [Chakraborty et al., 2018], comparably little attention has been dedicated to improving the model backbone, and even less to rethinking the training paradigm itself. Our method can also be easily combined with adversarial training, further increasing the model's robustness cheaply. Beyond benchmark measures of robustness, we show that if we optimize an image against our models directly, the resulting changes are human interpretable, suggesting at least much-harder-to-find instances of noise-like superstimuli that we usually find by attacking a model. This suggests an overall higher-quality, natural representations being learned by the model.

We operate under what what we call the **Interpretability-Robustness Hypothesis:** ***A model whose adversarial attacks typically look human-interpretable will also be adversarially robust.*** The aim of this paper is to support this hypothesis and to construct first versions of such robust classifiers, without necessarily reaching their peak performance via extensive hyperparameter tuning.

Firstly, inspired by biology, we design an active adversarial defense by constructing and training a classifier whose input, a standard $H \times W \times 3$ image, is stochastically turned into a $H \times W \times (3N)$ channel-wise stack of multiple downsampled and noisy versions of the same image. The classifier itself learns to make a decision about these N versions *at once*, mimicking the effect of microsaccades in the human (and mammal) vision systems. Secondly, we show experimentally that hidden layer features of a neural classifier show significant decorrelation between their representations under adversarial attacks – an attack fooling a network to see a *dog* as a *car* does not fool the intermediate representations, which still see a *dog*. We aggregate intermediate layer predictions into a self-ensemble dynamically, using a novel ensembling technique that we call a *CrossMax* ensemble. Thirdly, we show that our Vickrey-auction-inspired *CrossMax* ensembling yields significant gains in adversarial robustness when ensembling predictors as varied as 1) independent brittle models, 2) predictions of intermediate layers of the same model, 3) predictions from several checkpoints of the same model, and 4) predictions from several self-ensemble models. We use the last option to gain $\approx 5\%$ in adversarial accuracy at the $L_\infty = 8/255$ RobustBench's AutoAttack on top of the best models on CIFAR-100. When we add light adversarial training on top, we outperform current best models by $\approx 5\%$ on CIFAR-10, and by $\approx 9\%$ on CIFAR-100, showing a promising trend where the harder the dataset, the more useful our approach compared to brute force adversarial training (see Figure 4).

2 Key Observations and Techniques

In this section we will describe the three key methods that we use in this paper. In Section 2.1 we introduce the idea of multi-resolution inputs, in Section 2.2 we introduce our robust *CrossMax* ensembling method, and in Section 2.3 we showcase the de-correlation between adversarially induced mistakes at different layers of the network and how to use it as an active defense.

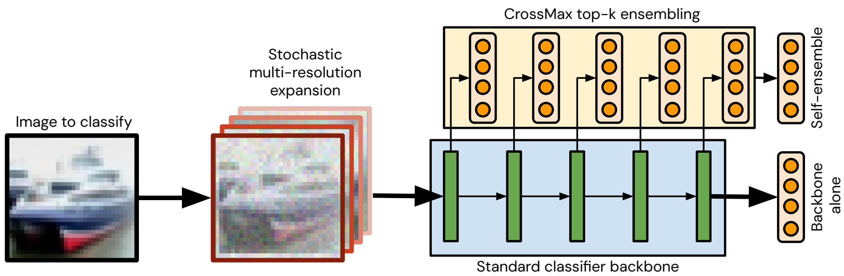


Figure 2: Combining channel-wise stacked augmented and down-sampled versions of the input image with robust intermediate layer class predictions via *CrossMax* self-ensemble. The resulting model gains a considerable adversarial robustness without any adversarial training or extra data.

2.1 The multi-resolution prior

Drawing inspiration from biology, we use multiple versions of the same image at once, down-sampled to lower resolutions and augmented with stochastic jitter and noise. We train a model to classify this channel-wise stack of images simultaneously. We show that this by default yields gains in adversarial robustness without any explicit adversarial training.

We turn an input image X of full resolution $R \times R$ and 3 channels (RGB) into its N variations of different resolutions $r \times r$ for $r \in \rho$. For CIFAR-10 and CIFAR-100, we are (arbitrarily) choosing resolutions $\rho = \{32, 16, 8, 4\}$ and concatenating the resulting image variations $\mathrm{rescale}_{\mathrm{R}}\left(\mathrm{rescale}_{r}(X)\right)$ channel-wise to a $R \times R \times (3|\rho|)$ augmented image $\bar{X}$. This is shown in Figure 7. Similar approaches have historically been used to represent images, such as Gaussian pyramids introduced in [Burt and Adelson, 1983]. To each variant we add 1) random noise both when downsampled and at the full resolution $R \times R$ (in our experiments of strength 0.1 out of 1.0), 2) a random jitter in the $x - y$ plane (± 3 in our experiments), 3) a small, random change in contrast, and 4) a small, random color-grayscale shift. This can also be seen as an effective reduction of the input space dimension available to the attacker, as discussed in [Fort, 2023].

2.2 *CrossMax* robust ensembling

The standard way of ensembling predictions of multiple networks is to either take the mean of their logits, or the mean of their probabilities. This increases both the accuracy as well as predictive uncertainty estimates of the ensemble [Lakshminarayanan et al., 2017, Ovadia et al., 2019]. Such aggregation methods are, however, susceptible to being swayed by an outlier prediction by a single member of the ensemble or its small subset. This produces a single point of failure. The pitfalls of uncertainty estimation and ensembling have been highlighted in [Ashukha et al., 2021], while the effect of ensembling on the learned classification function was studied by Fort et al. [2022].

We draw our intuition from *Vickrey auctions* [Wilson, 1977] which are designed to incentivize truthful bidding. Viewing members of ensembles as individual bidders, we can limit the effect of wrong, yet overconfident predictions by using the 2^{nd} highest, or generally k^{th} highest prediction per class. This also produces a cat-and-mouse-like setup for the attacker, since *which* classifier produces the k^{th} highest prediction for a particular class changes dynamically as the attacker tries to increase that prediction. A similar mechanism is used in balanced allocation [Azar et al., 1999] and specifically in the *k random choices* algorithm for load balancing [Mitzenmacher et al., 2001]. Our *CrossMax* aggregation is shown in Algorithm 1.

2.3 Only partial overlap between the adversarial susceptibility of intermediate layers

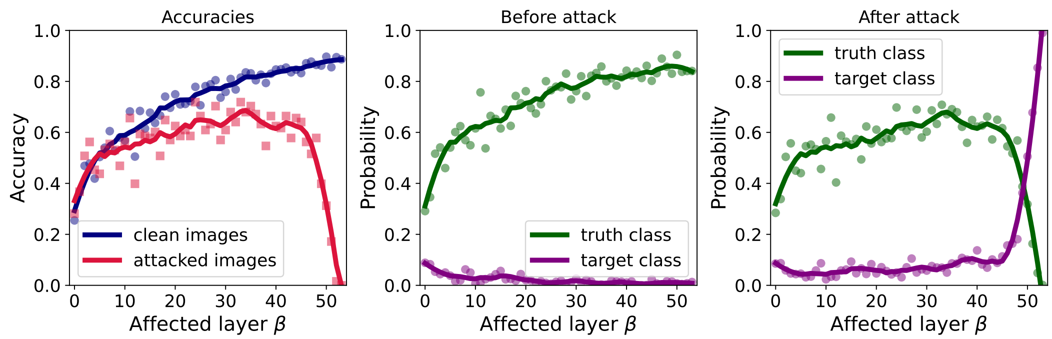


Figure 3: The impact of adversarial attacks ($L_\infty = 8/255$, 128 attacks) against the full classifier on the accuracy and probabilities at all intermediate layers for an ImageNet-1k pretrained ResNet152 finetuned on CIFAR-10 via trained linear probes.

A key question of both scientific and immediately practical interest is whether an adversarially modified image X' that looks like the target class t to a classifier $f : X \to y$ also has intermediate layer representations that look like that target class. In [Olah et al., 2017], it is shown via feature visualization that neural networks build up their understanding of an image hierarchically starting from edges, moving to textures, simple patterns, all the way to parts of objects and full objects themselves. This is further explored by Carter et al. [2019]. Does an image of a *car* that has been adversarially modified to look like a *tortoise* to the final layer classifier carry the intermediate features of the target class *tortoise* (e.g. the patterns on the shell, the legs, a tortoise head), of the original class *car* (e.g. wheels, doors), or something else entirely? We answer this question empirically.

In Figure 3 we showcase this effect using an ImageNet-pretrained ResNet152 [He et al., 2015] finetuned on CIFAR-10. Images attacked to look like some other class than their ground truth (to the final layer classification) do not look like that to intermediate layers, as shown by the target class probability only rising in the very last layers (see Figure 3). We can therefore confirm that indeed the activations of attacked images do not look like the target class in the intermediate layers, which offers two immediate use cases: 1) as a warning flag that the image has been tempered with and 2) as an active defense, which is strictly harder.

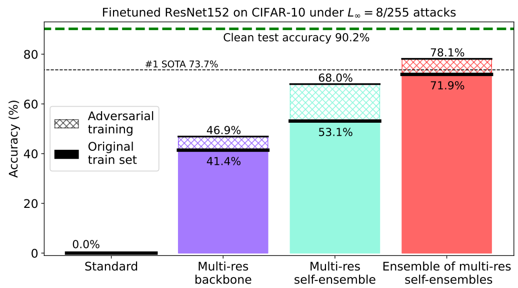
3 Training and Experimental Results

In this section we present in detail how we combine the previously described methods and techniques into a robust classifier on CIFAR-10 and CIFAR-100. We start both with a pretrained model and finetune it, as well as with a freshly initialized model. It turns out that finetuning a pre-existing model for robustness is technically easier and faster, therefore we predominantly focus on this approach. However, to demonstrate that the success of our technique does not simply come from massive pretraining, we also train a model from scratch. The concrete details of the model and training can be found in Appendix A.

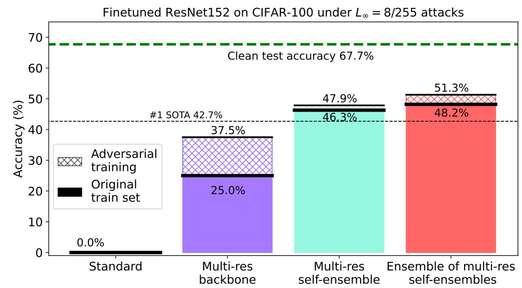
3.1 Adversarial vulnerability evaluation

To make sure we are using as strong an attack suite as possible to measure our networks' robustness and to be able to compare our results to other approaches, we use the `RobustBench` [Croce et al., 2020] library and its `AutoAttack` method, which runs a suite of four strong, consecutive adversarial attacks on a model in a sequence and estimates its adversarial accuracy (e.g. if the attacked images were fed back to the network, what would be the classification accuracy with respect to their ground truth classes). To evaluate our models using the hardest method possible, we ran the `AutoAttack` with the `rand` flag that is tailored against models using randomness. The results without adversarial training are shown in Table 1 and with adversarial training at Table 2. The visual representation of the results is presented in Figure 4.

3.2 Multi-resolution finetuning of a pretrained model



(a) CIFAR-10



(b) CIFAR-100

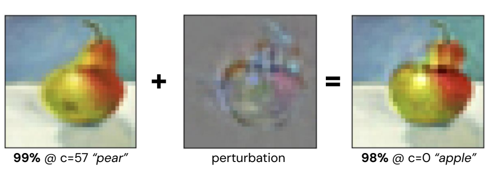
Figure 4: Adversarial robustness evaluation for finetuned ResNet152 models under $L_\infty = 8/255$ attacks of RobustBench AutoAttack (*rand* version, which is stronger against our models). On CIFAR-10, a CrossMax 3-ensemble of our self-ensemble multi-resolution models reaches #3 on the leaderboard, while on CIFAR-100 a 3-ensemble of our multi-resolution models is #1, leading by ≈+5 % in adversarial accuracy. When we add light adversarial training, our models surpass SOTA on CIFAR-10 by ≈+5 % and on CIFAR-100 by a strong ≈+9 %.

We demonstrate that this quickly leads to very significant adversarial robustness that matches and in some cases (CIFAR-100) significantly improves upon current best, dedicated approaches, without using any extra data or adversarial training. We see stronger gains on CIFAR-100 rather than CIFAR-10, suggesting that its edge might lie at harder datasets, which is a very favourable scaling compared to brute force adversarial training.

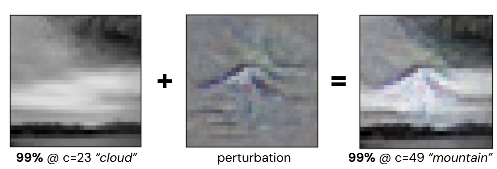
The steps we take are as follows: 1) Take a pretrained model (in our case ResNet18 and ResNet152 pretrained on ImageNet) 2) Replace the first layer with a fresh initialization that can take in $3N$ instead of 3 channels 3) Replace the final layer with a fresh initialization to project to 10 (for CIFAR-10) or 100 (for CIFAR-100) classes 4) Train the full network with a *small* (this is key) learning rate for a few epochs.

We find that using a small learning rate is key, which could be connected to the effects described for example in Thilak et al. [2022] and Fort et al. [2020]. While the network might reach a good clean test accuracy for high learning rates as well, only for small learning rates will it also get significantly robust against adversarial attacks, as shown in Figure 9. In Table 1 we present our results of finetuning an ImageNet pretrained ResNet152 on CIFAR-10 and CIFAR-100 for 10 epochs at the constant learning rate of 3.3×10^{-5} with Adam followed by 3 epochs at 3.3×10^{-6}. The details of our light adversarial finetuning are discussed in Appendix B.

3.3 Visualizing attacks against multi-resolution models



(a) *Pear* to *apple*



(b) *Cloud* to *mountain*

Figure 5: Examples of an adversarial attack on an image towards a target label. We use simple gradient steps with respect to our multi-resolution ResNet152 finetuned on CIFAR-100. The resulting attacks use the underlying features of the original image and make semantically meaningful, human-interpretable changes to it. Additional examples available in Figure 21.

We wanted to visualize the attacks against our multi-resolution models. In Figure 5 we start with a test set image of CIFAR-100 (a *pear*, *cloud*, *camel* and *elephant*) and over 400 steps with SGD and $\eta = 1$ minimize the loss with respect to a target class (*apple*, *mountain*, *rabbit* and *dinosaur*). We allow for large perturbations, up to $L_\infty = 128/255$, to showcase the alignment between our model and the implicit human visual system classification function. In case of the *pear*, the perturbation uses the underlying structure of the fruit to divide it into 2 apples by adding a well-placed edge. The resulting image is very obviously an apple to a human as well as the model itself. In case of the cloud, its white color is repurposed by the attack to form the snow of a mountain, which is drawn in by a dark sharp contour. In case of the elephant, it is turned into a dinosaur by being recolored to green and made spikier – all changes that are very easily interpretable to a human.

(a) $c = 0$ *apple* (b) $c = 35$ *girl* (c) $c = 46$ *man* (d) $c = 47$ *maple* (e) $c = 49$ *mountain*

Figure 6: Examples of adversarial attacks on our multi-resolution ResNet152 finetuned on CIFAR-100. The attacks are generated by starting from a uniform image (128,128,128) and using gradient descent of the cross-entropy loss with SGD at $\eta = 1$ for 400 steps towards the target label.

In Figure 6 we start with a uniform gray image of color (128, 128, 128) and by changing it to maximize the probability of a target class with respect to our model, we generate an image. The resulting images are very human-interpretable. We also generated 4 examples per CIFAR-100 class for all 100 classes in Figure 23 to showcase that we do not cherrypick the images shown.

4 Discussion and Conclusion

Our work demonstrates that taking inspiration from biology and stochastically translating an input image into a multi-resolution stack of inputs that are classified *simultaneously* by a model leads to higher-quality, natural representations, significant adversarial robustness, and human-interpretable attacks. Combining this with a novel, robust ensembling method inspired by Vickrey auctions that we call *CrossMax*, we demonstrate that we can further improve the model's adversarial robustness by combining its intermediate layer predictions into a *self-ensemble*. This is due to our empirical observation that intermediate layer representations are not fooled by attacks against the classifier as a whole, and that their induced errors are only partially correlated.

We are able to match the current state-of-the-art adversarial accuracy results on CIFAR-10 and surpass them by $\approx 5\%$ CIFAR-100 on a strong adversarial benchmark `RobustBench` without any extra data or dedicated adversarial training, that is usually needed to produce a robust model. When we add light adversarial training on top, we see that our methods are complementary to it and that we surpass the best models on CIFAR-10 by $\approx 5\%$ and by a very significant $\approx 9\%$ on CIFAR-100, taking it from $\approx 40\%$ to $\approx 50\%$ in a single step. Our methods seem to perform better on the harder dataset, suggesting a favourable scaling compared to the usual brute force adversarial training.

Our approach not only enhances robustness but also aligns the learned representations more closely with human visual processing, leading to more interpretable and reliable models. We demonstrate this by optimizing images against the outputs of our classifier directly and obtaining either human-interpretable changes, when applied to an existing image, or completely new, interpretable images, when starting from a uniform, empty image. This is in stark contrast to the usual result of such a procedure which would be a noise-like picture that would look very convincing to the network but would not resemble anything to a human.

References

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2013.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey, 2018. URL https://arxiv.org/abs/1810.00069.

P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. doi: 10.1109/TCOM.1983.1095851.

Stanislav Fort. Scaling laws for adversarial attacks on language model activations, 2023.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019.

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning, 2021.

Stanislav Fort, Ekin Dogus Cubuk, Surya Ganguli, and Samuel S. Schoenholz. What does a deep neural network confidently perceive? the effective dimension of high certainty class manifolds and their low confidence boundaries, 2022. URL https://arxiv.org/abs/2210.05546.

Robert B. Wilson. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 31(3):1106–1115, 1977.

Yossi Azar, Andrei Z Broder, Anna R Karlin, and Eli Upfal. Balanced allocations. *SIAM Journal on Computing*, 29:180–200, 1999.

Michael Mitzenmacher, Andrea W. Richa, and Ramesh Sitaraman. The power of two random choices: A survey of techniques and results. *Harvard University*, 2001.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 2019. doi: 10.23915/distill.00015. https://distill.pub/2019/activation-atlas.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark, 2020.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon, 2022.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M. Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel, 2020. URL https://arxiv.org/abs/2010.15110.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. URL https://ieeexplore.ieee.org/abstract/document/5206848/.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 770–778. IEEE, June 2016. doi: 10.1109/CVPR.2016.90. URL http://ieeexplore.ieee.org/document/7780459.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

ShengYun Peng, Weilin Xu, Cory Cornelius, Matthew Hull, Kevin Li, Rahul Duggal, Mansi Phute, Jason Martin, and Duen Horng Chau. Robust principles: Architectural design principles for adversarially robust cnns, 2023.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

Jiequan Cui, Zhuotao Tian, Zhisheng Zhong, Xiaojuan Qi, Bei Yu, and Hanwang Zhang. Decoupled kullback-leibler divergence loss, 2023. URL https://arxiv.org/abs/2305.13948.

Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training, 2023. URL https://arxiv.org/abs/2302.04638.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.

Brian R. Bartoldson, James Diffenderfer, Konstantinos Parasyris, and Bhavya Kailkhura. Adversarial robustness limits via scaling-law and human-alignment studies, 2024.

A van der Schaaf and J H van Hateren. Modelling the power spectra of natural images: Statistics and information. *Vision Research*, 36(17):2759–2770, September 1996. ISSN 0042-6989. Relation: http://www.rug.nl/informatica/organisatie/overorganisatie/iwi Rights: University of Groningen. Research Institute for Mathematics and Computing Science (IWI).

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Robert G Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.

Stanislav Fort. Pixels still beat text: Attacking the openai clip model with text patches and adversarial pixel perturbations. *URL https://stanislavfort. github. io/blog/OpenAI_CLIP_stickers_and_adversarial_examples*, 2021a.

Stanislav Fort. Adversarial examples for the openai clip in its zero-shot classification regime and their semantic generalization, jan 2021b. *URL https://stanislavfort. github. io/2021/01/12/OpenAI_CLIP_adversarial_examples. html*, 2021b.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL https://doi.org/10.5281/zenodo.5143773. If you use this software, please cite it as below.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

Rylan Schaeffer, Dan Valentine, Luke Bailey, James Chua, Cristóbal Eyzaguirre, Zane Durante, Joe Benton, Brando Miranda, Henry Sleight, John Hughes, Rajashree Agrawal, Mrinank Sharma, Scott Emmons, Sanmi Koyejo, and Ethan Perez. When do universal image jailbreaks transfer between vision-language models?, 2024. URL https://arxiv.org/abs/2407.15211.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, 2020.

A Model and training details

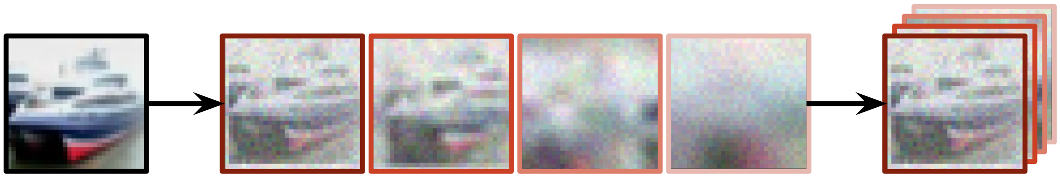


Figure 7: An image input being split into N progressively lower resolution versions that are then stacked channel-wise, forming a $3N$-channel image input to a classifier.

The pretrained models we use are the ImageNet [Deng et al., 2009] trained ResNet18 and ResNet152 [He et al., 2016]. Our hyperparameter search was very minimal and we believe that additional gains are to be had with a more involved search easily. The only architectural modification we make is to change the number of input channels in the very first convolutional layer from 3 to $3N$, where N is the number of channel-wise stacked down-sampled images we use as input. We also replaced the final linear layer to map to the correct number of classes (10 for CIFAR-10 and 100 for CIFAR-100). Both the new convolutional layer as well as the final linear layer are initialized at random. The batch norm [Ioffe and Szegedy, 2015] is on for finetuning a pretrained model (although we did not find a significant effect beyond the speed of training).

We focused on the CIFAR-* datasets [Krizhevsky, 2009, Krizhevsky et al.] that comprise 50,000 $32 \times 32 \times 3$ images. We arbitrarily chose $N = 4$ and the resolutions we used are 32×32, 16×16, 8×8, 4×4 (see Figure 7). We believe it is possible to choose better combinations, however, we did not run an exhaustive hyperparameter search there. The ResNets we used expect 224×224 inputs. We therefore used a `bicubic` interpolation to upsample the input resolution for each of the 12 channels independently.

To each image (the $32 \times 32 \times 3$ block of RGB channels) we add a random jitter in the $x - y$ plane in the ± 3 range. We also add a random noise of standard deviation 0.2 (out of 1.0). We believe that the biological jitter and noise are key aspects of a successful robust classifier, and therefore want to mimic their function here as well.

For training from scratch, we use a standard ResNet18 with the modifications above. We chose it since we primarily wanted to show the effect of multi-resolution inputs and multi-layer prediction aggregation rather than to find the maximum possible performance. We turn off batch normalization [Ioffe and Szegedy, 2015] not to conflate the effects we are exploring. While it is possible that additional architectural choices could lead to more robustness (as convincingly demonstrated in Peng et al. [2023]), we wanted to show the effect of our multi-resolution and self-ensemble choices in isolation.

All training is done using the `Adam` [Kingma and Ba, 2015] optimizer at a flat learning rate η that we always specify. Optimization is applied to all trainable parameters and the batch norm is turned on in case of finetuning, but turned off for training from scratch.

Linear probes producing predictions at each layer are just single linear layers that are trained on top of the pre-trained and frozen backbone network, mapping from the number of hidden neurons in that layer (flattened to a single dimension) to the number of classes (10 for CIFAR-10 and 100 for CIFAR-100). We trained them using the same learning rate as the full network for 1 epoch each.

B Adversarial finetuning

Adversarial training, which adds attacked images with their correct, ground truth labels back to the training set, is a standard brute force method for increasing models' adversarial robustness. [Chakraborty et al., 2018] It is ubiquitous among the winning submissions on the RobustBench leader board, e.g. in Cui et al. [2023] and Wang et al. [2023]. To verify that our technique does not only somehow replace the need for dedicated adversarial training, but rather that it can be productively combined with it for even stronger adversarial robustness, we re-ran all our finetuning experiments solely on adversarially modified batches of input images generated on the fly.

In all cases, we see an additive benefit of adversarial training on top of our techniques. In particular, for CIFAR-10 we outperform current SOTA by approximately 5 % while on CIFAR-100 and by approximately 9 % on CIFAR-100, which is a very large increase. The fact that our techniques benefit even from a very small amount of additional adversarial training (units of epochs of a single step attack) shows that our multi-resolution inputs and intermediate layer aggregation are a good prior for getting broadly robust networks.

B.1 Training from scratch

We train a ResNet18 from scratch on CIFAR-10 as a backbone, and then train additional linear heads for all of its intermediate layers to form a CrossMax self-ensemble. We find that, during training, augmenting our input images X with an independently drawn images X' with a randomly chosen mixing proportion p as $(1-p)X + pX'$ increases the robustness of the trained model. This simple augmentation technique is known as `mixup` and is described in Zhang et al. [2018]. We believe that this works well due to our multi-resolution inputs that are the correct prior for robustness, and show that without them such mixing does not increase robustness. For finetuning a pretrained model, however, this is not needed.

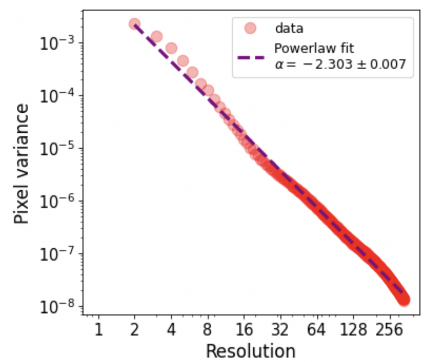
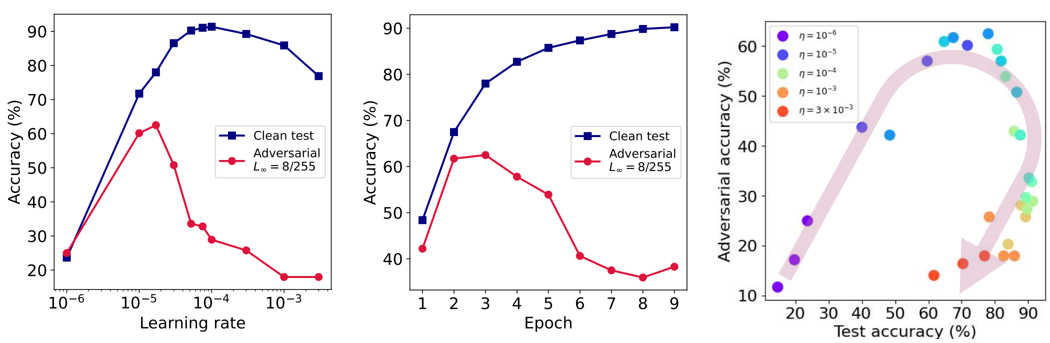


Figure 8: The image spectrum of generated multi-resolution attacks. The adversarial attacks generated over multiple resolutions at once end up showing very white-noise-like distribution of powers over frequencies (the slope for natural images is ≈ -2). This is in contrast with standard noise-like attacks.

For our ResNet18 model trained from scratch on CIFAR-10, we keep the pairs of images that are mixed in `mixup` fixed for 20 epochs at a time, producing a characteristic pattern in the training accuracies. Every 5 epochs we re-draw the random mixing proportions in the $[0, 1/2]$ range. We trained the model for 380 epochs with the Adam optimizer [Kingma and Ba, 2015] at learning rate 10^{-3} and dropped it to 10^{-4} for another 120 epochs. The final checkpoint is the weight average of the last 3 epochs. The training batch size is 512. These choices are arbitrary and we did not run a hyperparameter search over them.

The results on the full `RobustBench` AutoAttack suite of attacks for CIFAR-10 are shown in Table 1 for self-ensemble constructed on top of the multi-resolution ResNet18 backbone (the linear heads on top of each layer were trained for 2 epochs with Adam at 10^{-3} learning rate).



(a) Learning rate effects (b) Epoch effect (c) Accuracy vs robust accuracy

Figure 9: Finetuning a pretrained model with multi-resolution inputs. The left panel shows the test accuracy and adversarial accuracy after the first two attacks of RobustBench AutoAttack at $L_\infty = 8/255$ after 3 epochs of finetuning an ImageNet pretrained ResNet152. The middle panel shows the effect of training epoch for a single finetuning run at the learning rate $\eta = 1.7 \times 10^{-5}$. The right panel shows a hysteresis-like curve where high test accuracies are both compatible with low and high adversarial accuracies. The test accuracies are over the full 10,000 images while the adversarial accuracies are evaluated on 128 test images.

C Results tables

Dataset	Adv. train	Model	Method	#	Test acc	rand RobustBench AutoAttack $L_\infty = 8/255$ # samples (%) Adv acc	APGD→ CE	APGD DLR
CIFAR-10	×	ResNet18*	Self-ensemble	1024	76.94	64.06	51.56	44.53
CIFAR-10	×	ResNet152	Multi-res backbone	128	89.17	41.44	32.81	21.88
CIFAR-10	×	ResNet152	3-ensemble	128	91.06	67.97	61.72	59.38
CIFAR-10	×	ResNet152	Self-ensemble	128	87.14	53.12	50.00	43.75
CIFAR-10	×	ResNet152	3-ensemble of self-ensembles	128	90.20	**71.88**	68.75	68.75
CIFAR-10	✓	[30]	SOTA #1			73.71		
CIFAR-100	×	ResNet152	Multi-res backbone	128	65.70	25.00	21.88	13.28
CIFAR-100	×	ResNet152	3-ensemble	128	66.63	47.66	39.06	37.50
CIFAR-100	×	ResNet152	Self-ensemble	512	65.71	**46.29** ±2.36	34.77 ±2.09	30.08 ±2.13
CIFAR-100	×	ResNet152	3-ensemble of self-ensembles	512	67.71	**48.16** ±2.65	40.63 ±2.11	37.32 ±1.98
CIFAR-100	✓	[28]	SOTA #1			42.67		

Table 1: Full *randomized* (=the strongest against our approach) RobustBench AutoAttack adversarial attack suite results for 128 test samples at the $L_\infty = 8/255$ strength. In this table we show the results of attacking our multi-resolution ResNet152 models finetuned on CIFAR-10 and CIFAR-100 from an ImageNet pretrained state without any adversarial training or extra data for 20 epochs with Adam at $\eta = 3.3 \times 10^{-5}$. We use our *CrossMax* ensembling on the model itself (self-ensemble), the final 3 epochs (3-ensemble), and on self-ensembles from 3 different runs (3-ensemble of self-ensembles). We also include results for a ResNet18 trained from *scratch* on CIFAR-10. Despite its simplicity, our method gets adversarial robustness of $\approx 72\%$ on CIFAR-10 (ranking #3 on RobustBench leaderboard) and $\approx 48\%$ on CIFAR-100, surpassing current best models by $+5\%$. Unlike other approaches, we do not use any extra data or adversarial training and our models gain adversarial robustness by default. Additional adversarial training helps, as shown in Table 2.

Dataset	Adv. train	Model	Method	#	Test acc	rand RobustBench AutoAttack $L_\infty = 8/255$ # samples (%) Adv acc	APGD→ CE	APGD DLR
CIFAR-10	✓	ResNet152	Multi-res backbone	128	87.19	46.88	34.38	32.03
CIFAR-10	✓	ResNet152	Self-ensemble	128	84.58	67.94	64.06	54.69
CIFAR-10	✓	ResNet152	3-ensemble of self-ensembles	128	87.00	**78.13**	73.44	72.65
CIFAR-10	✓	[30]	SOTA #1			73.71		
CIFAR-100	✓	ResNet152	Multi-res backbone	128	62.72	37.50	32.03	22.66
CIFAR-100	✓	ResNet152	Self-ensemble	512	58.93	**47.85** ±2.66	36.72 ±3.01	33.98 ±2.72
CIFAR-100	✓	ResNet152	3-ensemble of self-ensembles	512	61.17	**51.28** ±1.95	44.60 ±2.00	43.04 ±1.97
CIFAR-100	✓	[28]	SOTA #1			42.67		

Table 2: Full *randomized* (=the strongest against our approach) RobustBench AutoAttack adversarial attack suite results for 128 test samples at the $L_\infty = 8/255$ strength. In this table we show the results of attacking our multi-resolution ResNet152 models finetuned on CIFAR-10 and CIFAR-100 from an ImageNet pretrained state **with** light adversarial training.

D Additional Insights and Applications

We want to support our multi-resolution input choice as an active defense by demonstrating that by reversing it and representing an adversarial perturbation *explicitly* as a sum of perturbations at different resolutions, we get human-interpretable perturbations by default.

E Single-resolution adversarial attacks

Natural images contain information expressed on all frequencies, with an empirically observed power-law scaling. The higher the frequency, the lower the spectral power, as $\propto f^{-2}$ [van der Schaaf and van Hateren, 1996].

While having a single perturbation P of the full resolution $R \times R$ theoretically suffices to express anything, we find that this choice induces a specific kind of high frequency prior. Even simple neural networks can theoretically express any function [Hornik et al., 1989], yet the specific architecture matters for what kind of a solution we obtain given our data, optimization, and other practical choices. Similarly, we find that an alternative formulation of the perturbation P leads to more natural looking and human interpretable perturbations despite the attacker having access to the highest-resolution perturbation as well and could in principle just use that.

F Multi-resolution attacks

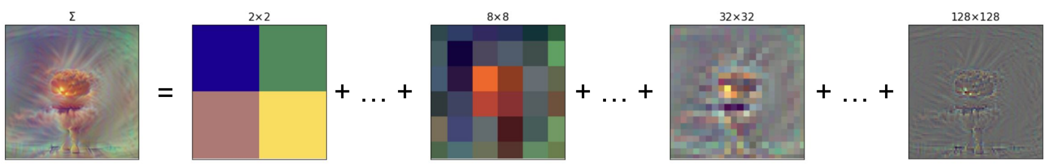


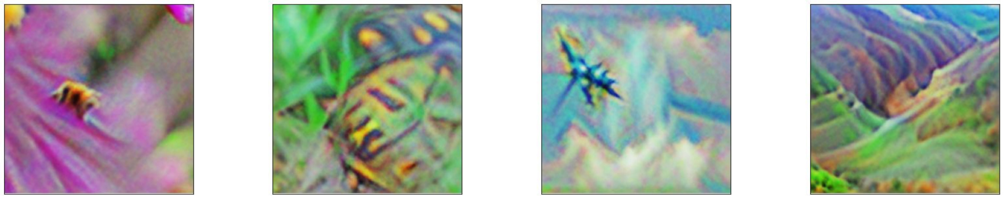
Figure 10: The result of expressing an image as a set of resolutions and optimizing it towards the CLIP embedding of the text '*a photo of a nuclear explosion*'. The plot shows the resulting sum of resolutions (left panel, marked with ρ) and selected individual perturbations P_r of resolutions 2×2, 8×8, 32×32 and 128×128. The intensity of each is shifted and rescaled to fit between 0 and 1 to be recognizable visually, however, the pixel values in the real P_r fall of approximately as r^{-1}.

We express the single, high resolution perturbation P as a sum of perturbations $P = \sum_{r \in \rho} \mathrm{rescale}_R(P_r)$, where P_r is of the resolution $r \times r$ specified by a set of resolutions ρ, and the $\mathrm{rescale}_R$ function rescales and interpolates an image to the full resolution $R \times R$. When we jointly optimize the set of perturbations $\{P_r\}_{r \in \rho}$, we find that: a) the resulting attacked image $X + \sum_{r \in \rho} \mathrm{rescale}_R(P_r)$ is much more human-interpretable, b) the attack follows a power distribution of natural images.

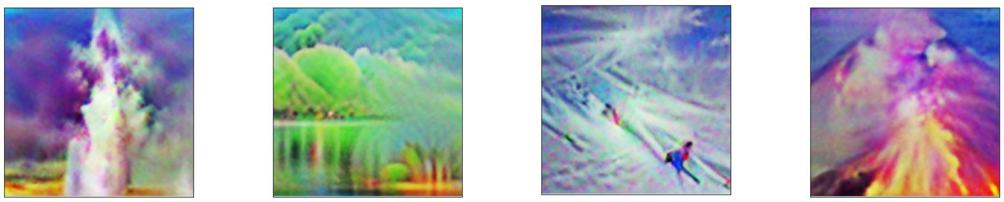
When attacking a classifier, we choose a target label t and optimize the cross-entropy loss of the predictions stemming from the perturbed image as if that class t were ground truth. To add to the robustness and therefore interpretability of the attack (as hypothesized in our *Interpretability-Robustness Hypothesis*), we add random jitter in the x-y plane and random pixel noise, and design the attack to work on a set of models.

Figure 11: An attack on vision language models. GPT-4 sees *Rick Astley from his famous "Never Gonna Give You Up" music video* tree. See Table 3 and 4 for details.

An example of the multi-resolution sum is show in Figure 12.
There we use a simple Stochastic Gradient Descent [Robbins and Monro, 1951] optimization with the learning rate of 5×10^{-3} and a cosine decay schedule over 50 steps. We add a random pixel noise of 0.6 (out of 1), jitter in the x-y plane in the ± 5 range and a set of all perturbations from 1×1 to 224×224 interpolated using `bicubic` interpolation [Keys, 1981]. In Figure 12 we see that despite the very limited expressiveness of the final layer class label, we can still recover images that look like the target class to a human. We also tested them using Gemini Advanced and GPT-4, asking what



(a) $c = 309$ *bee* (b) $c = 37$ *box turtle* (c) $c = 895$ *warplane* (d) $c = 979$ *valley*



(e) $c = 974$ *geyser* (f) $c = 975$ *lakeside* (g) $c = 795$ *ski* (h) $c = 980$ *volcano*

Figure 12: Examples of images generated as attacks on ImageNet-trained classifiers. These images were generated by minimizing the cross-entropy loss of seven pretrained classifiers with respect to the target ImageNet class. Spatial jitter in the ± 5 pixel range and pixel noise of standard deviation 0.6 were applied during SGD optimization with learning rate 5×10^{-3} over 50 steps with a cosine schedule. The perturbation was expressed as a sum of perturbations at all resolutions from 1×1 to 224×224 that were optimized at once.

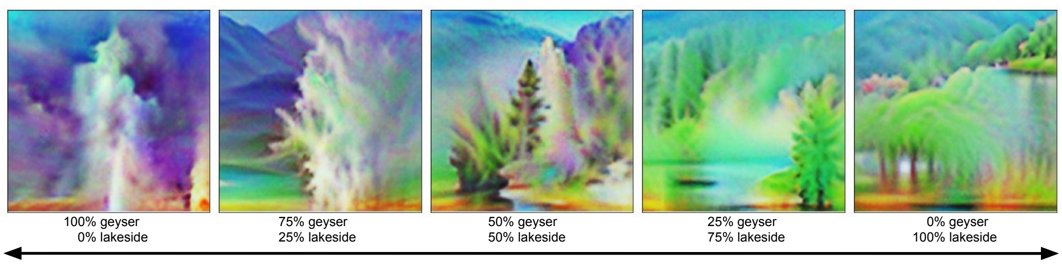


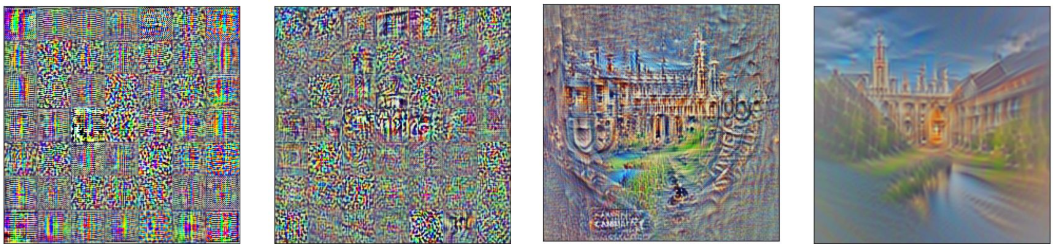
Figure 13: Optimizing towards a probability vector with a sliding scale between $c = 974$ *geyser* and $c = 975$ *lakeside*. Optimizing against pretrained classifiers generated semantically blended image of the two concepts.

the AI model sees in the picture, and got the right response in all 8 cases. To demonstrate that we can generate images beyond the original 1000 ImageNet classes, we experimented with setting the target label not as a one-hot vector, but rather with target probability p on class t_1 and $1 - p$ on t_2. For classes $c = 974$ (*geyser*) and $c = 975$ (*lakeside*) we show, in Figure 13 that we get semantically meaningful combinations of the two concepts in the same image as we vary p from 0 to 1. $p = 1/2$ gives us a *geyser* hiding beyond trees at a *lakeside*. This example demonstrates that in a limited way, classifiers can be used as controllable image generators.

G Multi-resolution attack on CLIP

The CLIP-style [Radford et al., 2021] models map an image I to an embedding vector $f_I : I \to v_I$ and a text T to an embedding vector $f_T : T \to v_T$. The cosine between these two vectors corresponds to the semantic similarity of the image and the text, $\cos(v_I, v_T) = v_I \cdot v_T / (|v_I||v_T|)$. This gives us $\mathrm{score}(I, T)$ that we can optimize.

Adversarial attacks on CLIP can be thought of as starting with a human-understandable image X_0 (or just a noise), and a target label text T^*, and optimizing for a perturbation P to the image that tries to increase the $\mathrm{score}(X_0 + P, T^*)$ as much as possible. In general, finding such perturbations is easy, however, they end up looking very noise-like and non-interpretable. [Fort, 2021a,b].



(a) Just a 224×224 perturbation alone. (b) Adding random noise to optimization. (c) Adding random jitter to optimization. (d) Adding all resolutions from 1×1 to 224×224.

Figure 14: The effect of adding noise, jitter, and a full set of resolutions to an adversarial attack on CLIP towards the text *'a beautiful photo of the University of Cambridge, detailed'*. While using just a plain perturbation of the full resolution in Figure 14a, as is standard in the typical adversarial attack setup, we get a completely noise-like image. Adding random noise to the pixels during optimization leads to a glimpse of a structure, but still maintains a very noise-like pattern (Figure 14b). Adding random jitter in the x-y plane on top, we can already see interpretable shapes of *Cambridge* buildings in Figure 14c. Finally, adding perturbations of all resolutions, $1 \times 1, 2 \times 2, \ldots, 224 \times 224$, we get a completely interpretable image as a result in Figure 14d.

If we again express $P = \mathrm{rescale}_{224}(P_1) + \mathrm{rescale}_{224}(P_2) + \cdots + P_{224}$, where P_r is a resolution $r \times r$ image perturbation, and optimize $\mathrm{score}(X_0 + \mathrm{rescale}_{224}(P_1) + \mathrm{rescale}_{224}(P_2) + \cdots + P_{224}, T^*)$ by simultaneously updating all $\{P_r\}_r$, the resulting image $X_0 + \sum_{r \in [1,224]} \mathrm{rescale}_R(P_r)$ looks like the target text T^* to a human rather than being just a noisy pattern. Even though the optimizer could choose to act only on the full resolution perturbation P_{224}, it ends up optimizing all of them jointly instead, leading to a more natural looking image. To further help with natural-looking attacks, we introduce pixel noise and the *x-y* plane jitter, the effect of which is shown in Figure 14.

We use SGD at the learning rate of 5×10^{-3} for 300 steps with a cosine decay schedule to maximize the cosine between the text description and our perturbed image. We use the `OpenCLIP` models [Ilharco et al., 2021, Cherti et al., 2023] (an open-source replication of the CLIP model [Radford et al., 2021]). Examples of the resulting "adversarial attacks", starting with a blank image with 0.5 in its RGB channels, and optimizing towards the embedding of specific texts such as *"a photo of Cambridge UK, detailed*, and *"a photo of a sailing boat on a rough sea"* are shown in Figure 16. The image spectra are shown in Figure 8, displaying a very natural-image-like distribution of powers. The resulting images look very human-interpretable.

(a) Original (b) *Albert Einstein* (c) *Queen Elizabeth* (d) *Nikola Tesla*

Figure 15: Starting with an image of Isaac Newton and optimizing a multi-resolution perturbation towards text embeddings of *Albert Einstein*, *Queen Elizabeth* and *Nikola Tesla* leads to a change in the face of the person depicted. This demonstrates how semantically well-targeted such multi-resolution attacks are. All 4 images are recognizable as the target person to humans as well as GPT-4o and Gemini Advanced.

Starting from a painting of Isaac Newton and optimizing towards the embeddings of *"Albert Einstein"*, *"Queen Elizabeth"* and *"Nikola Tesla"*, we show that the attack is very semantically targeted, effectively just changing the facial features of Isaac Newton towards the desired person. This is shown in Figure 15. This is exactly what we would ideally like adversarial attacks to be – when changing the content of what the model sees, the same change should apply to a human. We use a similar method to craft transferable attacks (see Figure 11 for an example) against commercial, closed source vision language models (GPT-4, Gemini Advanced, Claude 3 and Bing AI) in Table 3, in which a *turtle* turns into a *cannon*, and in Table 4, where *Stephen Hawking* turns into the music video *Never Gonna Give You Up* by *Rick Astley*. The attacks also transfer to Google Lens, demonstrating that the multi-resolution prior also serves as a good *transfer* prior and forms an early version of a transferable image vision language model jailbreak. This is a constructive proof to the contrary of the non-transferability results in Schaeffer et al. [2024].

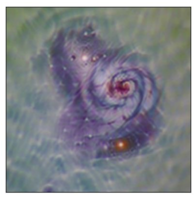
(a) *Ancient Rome*

(b) *Cambridge, UK*

(c) *Prague Castle in spring*

(d) *Oxford, UK*

(e) *sailing ship on stormy sea*



(f) *the Whirlpool Galaxy, M51*

(g) *a large ship cannon firing*

(h) *African savanna with animals and trees*

Figure 16: Examples of images generated with the multi-resolution prior, jitter and noise with the OpenCLIP models. The text whose embedding the image optimizes to approach is of the form '*A beautiful photo of [X], detailed*' for different values of [X].

H Additional details on attack transfer between layers

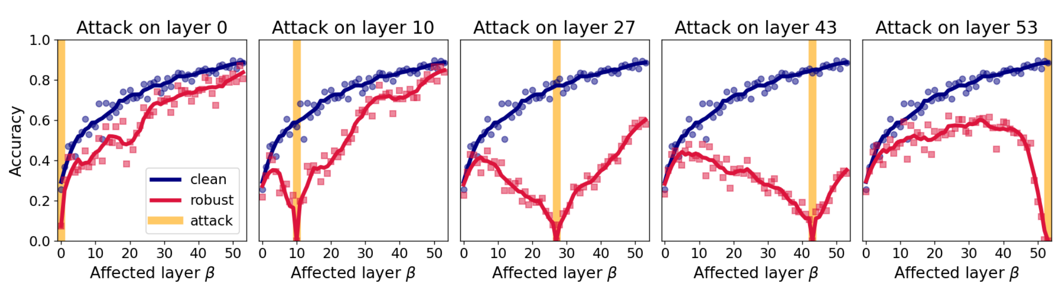


Figure 17: Transfer of adversarial attacks ($L_\infty = 8/255$, 512 attacks) against the activations of layer α on the accuracy of layer β for $\alpha = 0, 10, 27, 43, 53$ on ImageNet-1k pretrained ResNet152 finetuned on CIFAR-10 via trained linear probes. Each panel shows the effect of designing a pixel-level attack to confuse the linear probe at a particular layer. The blue curve is the test accuracy on the unperturbed data, and the red line shows the accuracy on the attacked images. The accuracy drops to 0 at the layer that is directly attacked (marked in orange), showing a successful attack. The effect is localized: attacking early layers mainly affects early layer predictions, middle layer attacks primarily affect middle layers, and likewise attacks on the final layers (the standard regime) primarily influence late layer performance. For more details, see Figure 19.

I Transfer to massive commercial models

In Table 3 we show the results of asking *"What do you see in this photo?"* and adding the relevant picture to four different, publicly available commercial AI models: GPT-4[1], Bing Copilot[2], Claude 3 Opus[3] and Gemini Advanced[4]. We find that, with an exception of Gemini Advanced, even a $L_\infty = 30/255$ attack generated in approximately 1 minute on a single A100 GPU (implying a cost at most in cents) fools these large models into seeing a *cannon* instead of a *turtle*. The attack also transfers to Google Lens.

	Original	$L_\infty = 20/255$	$L_\infty = 30/255$	$L_\infty = 40/255$	$L_\infty = 70/255$	$L_\infty = 100/255$
						
GPT-4	sea **turtle** swimming	**turtle** swimming in water	**cannon** mounted on stone base, firing	**cannon** with a notably ornate and rusted appearance	**cannon** mounted on a brick platform	stylized or artistically rendered depiction of a **cannon**
Bing Copilot	sea **turtle** gracefully swimming	sea **turtle** gracefully swimming	a **cannon** mounted on a stone base	**cannon** with a wheel, mounted on a stone base	old **cannon** mounted on a brick platform	color-saturated **cannon** mounted on wheels
Claude 3 Opus	sea **turtle** swimming in clear, turquoise water	sea **turtle** swimming underwater	old **cannon** submerged underwater	old decorative **cannon** sitting on a stone or concrete platform	old naval **cannon** set on a stone or brick platform	artistic painting or illustration of an old **cannon**
Gemini Advanced	sea **turtle** swimming underwater	sea **turtle** swimming underwater	sea **turtle** swimming	sea **turtle** swimming in a pool	**cannon** being fired by a **turtle** wearing a red jacket	artistic interpretation of a **cannon** firing

Table 3: Multi-resolution adversarial attacks of increasing L_∞ using OpenCLIP on an image of a sea turtle towards the text *"a cannon"* tested on GPT-4, Bing Copilot (Balanced), Claude 3 Sonnet and Gemini Advanced. All models we tested the images on were publicly available. The conversation included a single message *"What do you see in this photo?"* and an image. We chose the most relevant parts of the response.

Figure 18 compares attacks on robust and brittle models.

J Attack transfer between layers

This setup also allows us not only to investigate what the intermediate classification decision would be for an adversarially modified image X' that confuses the network's final layer classifier, but also to generally ask what the effect of confusing the classifier at layer α would do to the logits at a layer β. The results are shown in Figure 17 for 6 selected layers to attack, and the full attack layer $\times$ read-out layer is show in Figure 19.

[1] chatgpt.com
[2] bing.com/chat
[3] claude.ai/
[4] gemini.google.com

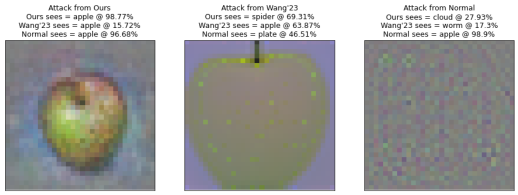
	Original	$L_\infty = 20/255$	$L_\infty = 30/255$	$L_\infty = 40/255$	$L_\infty = 70/255$	$L_\infty = 100/255$
						
GPT-4	Stephen Hawking	Stephen Hawking	Never Gonna Give You Up	Never Gonna Give You Up	Never Gonna Give You Up	singer or performer, possibly Rick Astley
Bing Copilot	individual sitting in a wheelchair	individual sitting on a bench	individual sitting down, holding a microphone, singing	person seated, holding a musical instrument	two individuals in an indoor setting	person in front of a microphone, singing
Claude 3 Opus	elderly man in a wheelchair	man in a wheelchair, smiling	young man with blonde hair, vintage-style microphone, singing	young man with blond hair, 1980s pop music	music video, 1980s, singer	music video, 1980s fashion
Gemini Advanced	Refused to answer.	Refused to answer.	Refused to answer.	Refused to answer.	Refused to answer.	Refused to answer.

Table 4: Multi-resolution adversarial attacks of increasing L_∞ using OpenCLIP on an image of *Stephen Hawking* towards the embedding of an image from the famous *Rick Astley's* song *Never Gonna Give You Up* from the 1980s tested on GPT-4, Bing Copilot (Balanced), Claude 3 Sonnet and Gemini Advanced. All models we tested the images on were publicly available. The conversation included a single message *"What do you see in this photo?"* and an image. We chose the most relevant part of the response. Unfortunately, Gemini refused to answer, likely due to the presence of a human face in the photo.

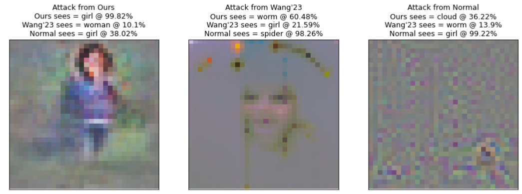
We find that attacks designed to confuse early layers of a network do not confuse its middle and late layers. Attacks designed to fool middle layers do not fool early nor late layers, and attacks designed to fool late layers do not confuse early or middle layers. In short, there seems to be roughly a 3-way split: early layers, middle layers, and late layers. Attacks designed to affect one of these do not generically generalize to others. We call this effect the *adversarial layer de-correlation*. This de-correlation allows us to create a *self-ensemble* from a single model, aggregating the predictions resulting from intermediate layer activations. To make sure that the ensemble is robust, we use the *CrossMax* method described in Section 2.2 and Algorithm 1. While ensembling multiple equivalent models, we did not have to care about their different quality, however, here early layers are typically less accurate than late layers, as shown in Figure 3.

In Figure 24 we show the self-ensemble robustness under adversarial attacks of different strength for an ImageNet pretrained ResNet152 and ViT-B/16, with linear heads at each layer separately finetuned on CIFAR-10. The aggregation method in Algorithm 1 provides non-zero robust accuracy for attacks of even $L_\infty = 5/255$, while standard ensembling using mean logits as well as just the last layer prediction loses robust accuracy around 3/255. This is an early indication that CrossMax self-ensembling can actively use the decorrelation of intermediate layer adversarial susceptibilities for an active, white-box defense.

K Visualizing attacks on multi-resolution models



(a) Apple ($c = 0$): The image generated from our model looks like an *apple* to itself, the Wang et al. [2023] robust model, and a brittle ResNet152 alike. The attacks against Wang et al. [2023] and standard ResNet152, on the other hand, convince only themselves.



(b) Girl ($c = 35$): The image generated from our model looks like a *girl* to itself, a brittle ResNet152 alike, and as a *woman* to the Wang et al. [2023] robust model. The attacks against them, on the other hand, convince only themselves.

Figure 18: Examples of adversarial attacks on our multi-resolution ResNet152 finetuned on CIFAR-100 (left), the previous best model on CIFAR-100 $L_\infty = 8/255$ on RubustBench from Wang et al. [2023] (middle), and standard ResNet152 finetuned on CIFAR-100. The attacks are generated by starting from a uniform image (128,128,128) and using gradient descent of the cross-entropy loss with SGD at $\eta = 1$ for 400 steps towards the target label. The prediction results for each of the models are shown above the images.

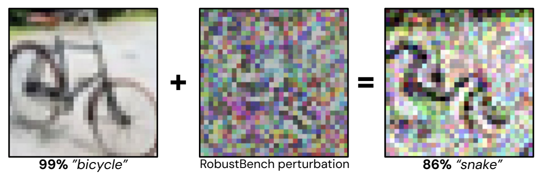


Figure 20: An example of a $L_\infty = 64/255$ RobustBench AutoAttack on our model, changing a *bicycle* into a *snake* in an interpretable way.

Figure 22 shows 6 examples of successfully attacked CIFAR-100 test set images for an ensemble of 3 self-ensemble models – our most adversarially robust model. When looking at the misclassifications caused, we can easily see human-plausible ways in which the attacked image can be misconstrued as the most probable target class. For example, a crab with a body resembling a mushroom cap gets a foot of a mushroom added by the attack, causing a misclassification as 40% mushroom from a 90% crab. A blurry picture of a sting ray gets 3D-like shading added by the attack, making it look mouse-like and being classified as 30% shrew from a 90% ray. Overall, we see that the changes that are induced by the attacker seem to have a human-understandable explanation. Figure 20 shows an example of a successful $L_\infty = 64/255$ (much larger than the standard 8/255 perturbations) RobustBench AutoAttack on a test image of a *bicycle* converting it, in a human-interpretable way, to a *snake* by re-purposing parts of the bicycle frame as the snake body.

We are also very interested in the existence of adversarial attacks on the human visual system and we believe that our work should be an update against their likelihood. We use biologically inspired methods (multiple resolutions, jitter, noise) that work as a defense against a white-box attacker. When flipped around, the same ideas generate human-interpretable images. The intermediate layer representations could also be viewed as using shallower circuits in the brain, and their partial robustness might suggest the same in humans. Given that moving closer (in a very rudimentary way) to the human visual system in these regards gave us both a practical defense and an image generator, we believe that we should update against adversarial vulnerability of humans.

L Additional experiments for CrossMax

To demonstrate experimentally different characteristics of prediction aggregation among several classifiers, we trained 10 ResNet18 models, starting from an ImageNet pretrained model, changing their final linear layer to output 10 classes of CIFAR-10. We then used the first 2 attacks of the RobustBench `AutoAttack` suite (APGD-T and APGD-CE; introduced by Croce and Hein [2020] as particularly strong attack methods) and evaluated the robustness of our ensemble of 10 models under adversarial attacks of different L_∞ strength. The results are shown in Figure 25.

The aggregation methods we show are 1) our CrossMax (Algorithm 1) (using *median* since the 10 models are expected to be equally good), 2) a standard logit mean over models, 3) median over

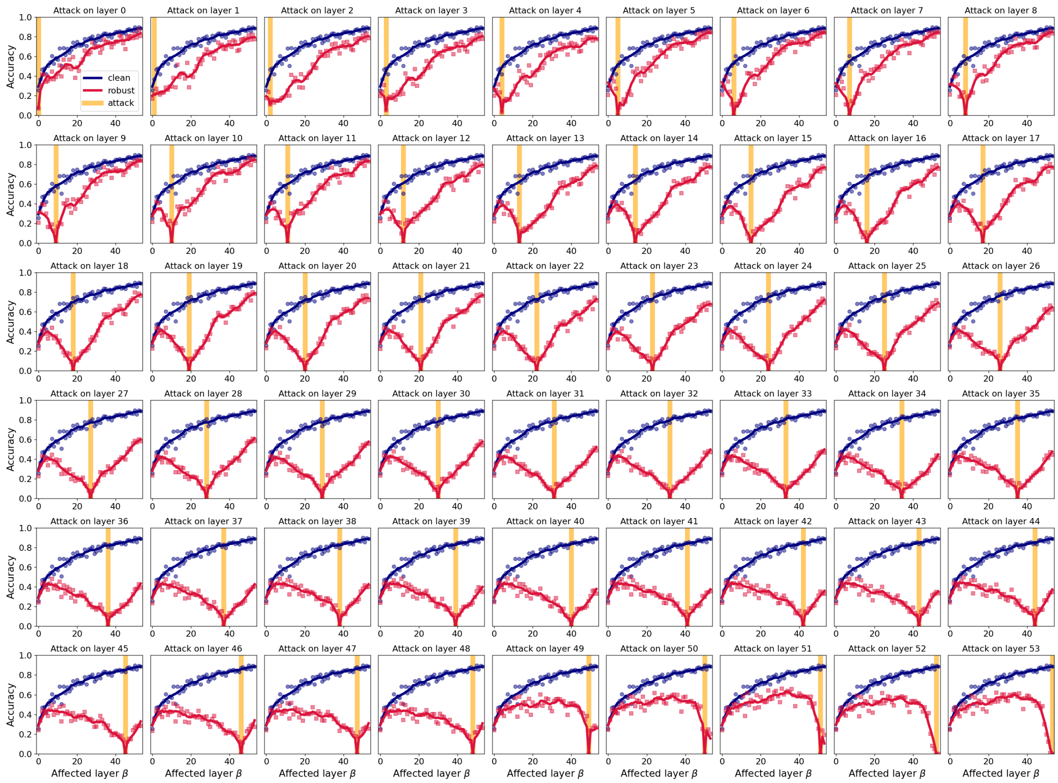
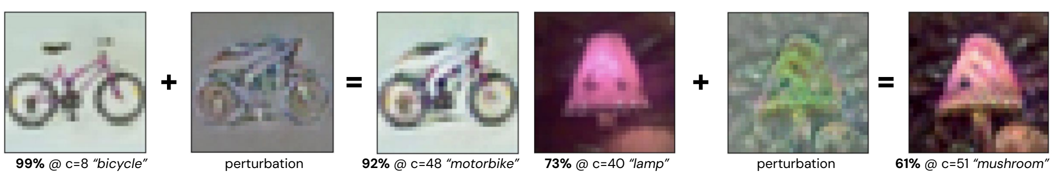
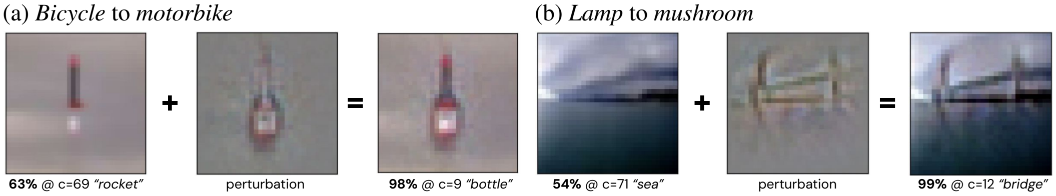


Figure 19: Attack transfer between layers of the ResNet154 model pre-trained on ImageNet-1k. The individual linear heads were finetuned on CIFAR-10 on top of the frozen model.



(a) *Bicycle* to *motorbike*

(b) *Lamp* to *mushroom*



(c) *Rocket* to *bottle*

(d) *Sea* to *bridge*

Figure 21: Additional examples of an adversarial attack on an image towards a target label. We use simple gradient steps with respect to our multi-resolution ResNet152 finetuned on CIFAR-100. The resulting attacks use the underlying features of the original image and make semantically meaningful, human-interpretable changes to it. Additional examples available in Figure 5.

models, and 4) the performance of the individual models themselves. While an ensemble of 10 models, either aggregated with a mean or median, is more robust than individual models at all attack strengths, it nonetheless loses robust accuracy very fast with the attack strength L_∞ and at the standard level of $L_\infty = 8/255$ it drops to ≈0%. Our *CrossMax* in Algorithm 1 provides > 0 robust accuracy even to $10/255$ attack strengths, and for $8/255$ gives a 17-fold higher robust accuracy than just plain mean or median. We use this aggregation for intermediate layer predictions (changing *median* to top_3) as well and see similar, transferable gains. We call this setup a *self-ensemble*.

Figure 22: Examples of successfully attacked CIFAR-100 images for an ensemble of self-ensembles – our most robust model. We can see human-plausible ways in which the attack changes the perceived class. For example, the skyscraper has a texture added to it to make it look tree-like.

Algorithm 1 CrossMax = An Ensembling Algorithm with Improved Adversarial Robustness

Require: Logits Z of shape $[B, N, C]$, where B is the batch size, N the number of predictors, and C the number of classes
Ensure: Aggregated logits
1: $\hat{Z} \leftarrow Z - \max(Z, \text{axis} = 2)$ {Subtract the max per-predictor over classes to prevent any predictor from dominating}
2: $\hat{Z} \leftarrow \hat{Z} - \max(\hat{Z}, \text{axis} = 1)$ {Subtract the per-class max over predictors to prevent any class from dominating}
3: $Y \leftarrow \text{median}(\hat{Z}, \text{axis} = 1)$ {Choose the median (or k^{th} highest for self-ensemble) logit per class}
4: **return** Y

As an ablation, we tested variants of the *CrossMax* method. There are two normalization steps: A) subtracting the per-predictor max, and B) subtracting the per-class max. We exhaustively experiment with all combinations, meaning $\{\_, A, B, AB, BA\}$, (robust accuracies at 4/255 are $\{4, 4, 0, 22, 0\}\%$) and find that performing A and then B, as in Algorithm 1, is by far the most robust method. We perform a similar ablation for a robust, multi-resolution self-ensemble model in Table 5 and reach the same verdict, in addition to confirming that the algorithm is very likely not accidentally masking gradients.

Aggregation fn	topk$_2$					mean				
Method	_	A	B	BA	AB	_	A	B	BA	AB
Test acc	57.08	59.86	0.82	1.27	58.92	60.31	59.89	1.1	1.05	**57.23**
Adv acc	46.88	46.88	1.56	0.00	57.81	40.62	48.44	0.00	0.00	39.06

Table 5: CrossMax algorithm ablation. The Algorithm 1 contains two subtraction steps: A = the per-predictor max subtraction, and B = the per-class max subtraction. This Table shows the robust accuracies of a self-ensemble model on CIFAR-100 trained with light adversarial training, whose intermediate layer predictions were aggregated using different combinations and orders of the two steps. We also look at the effect of using the final topk_2 aggregation vs just using a standard mean. The best result is obtained by the Algorithm 1, however, we see that not using the topk does not lead to a critical loss of robustness as might be expected if there were accidental gradient masking happening.

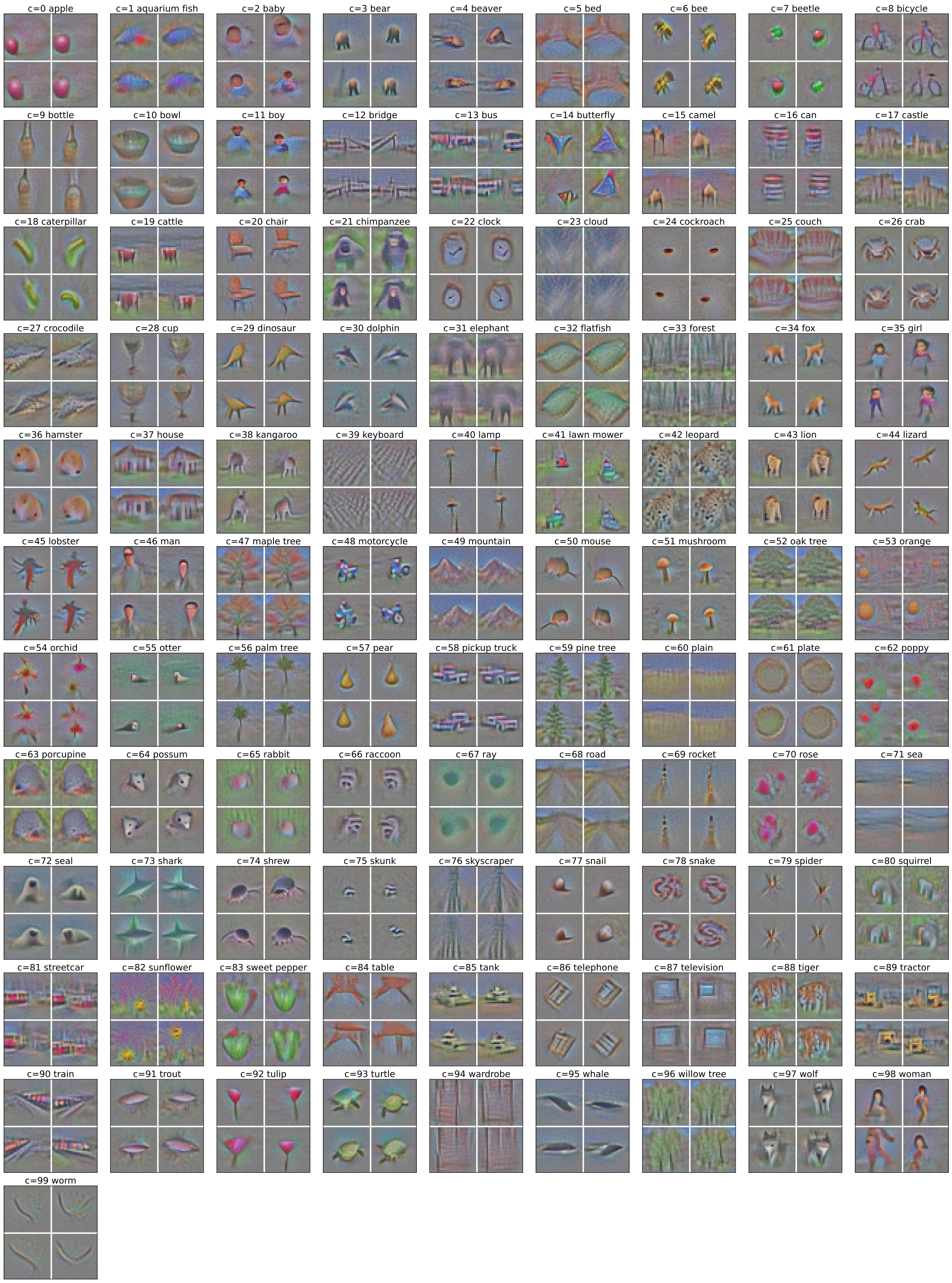
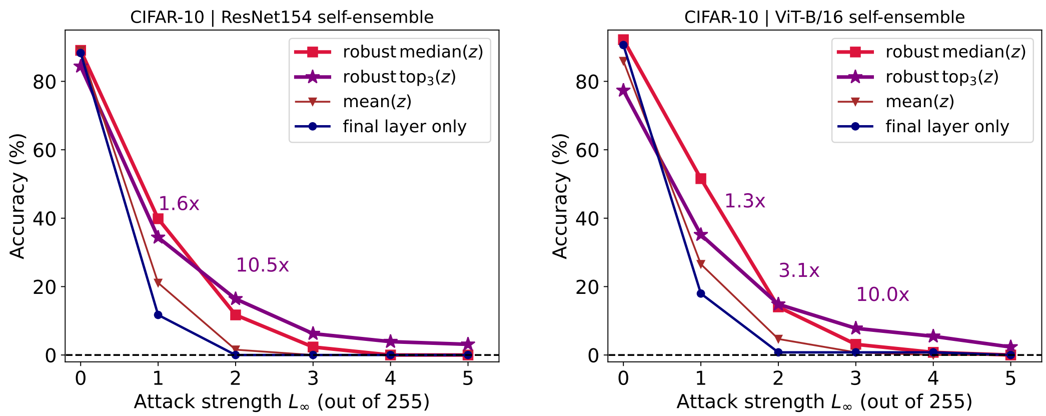


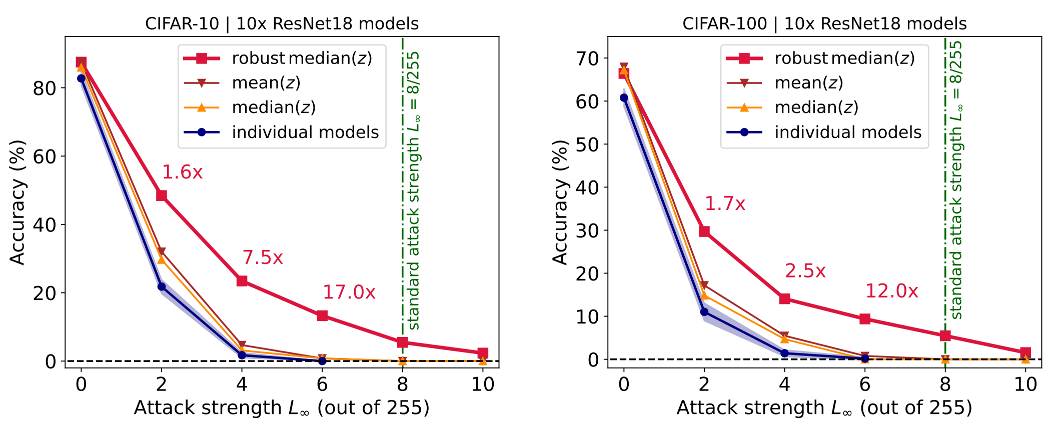
Figure 23: Examples of optimizing towards all 100 CIFAR-100 classes against our multi-resolution ResNet152 model, 4 examples for each. We use 400 simple gradient steps at learning rate $\eta = 1$ with SGD with respect to the model, starting from all grey pixels (128,128,128). The resulting attacks are easily recognizable as the target class to a human.



(a) ResNet154 self-ensemble on CIFAR-10

(b) ViT-B/16 self-ensemble on CIFAR-10

Figure 24: The robust accuracy of different types of self-ensembles of ResNet152 and ViT-B/16 with linear heads finetuned on CIFAR-10 under increasing L_∞ attack strength.



(a) CIFAR-10

(b) CIFAR-100

Figure 25: The robust accuracy of different types of ensembles of 10 ResNet18 models under increasing L_∞ attack strength. Our robust median ensemble, *CrossMax*, gives very non-trivial adversarial accuracy gains to ensembles of individually brittle models. For $L_\infty = 6/255$, its CIFAR-10 robust accuracy is 17-fold larger than standard ensembling, and for CIFAR-100 the factor is 12.