6 Appendix

6.1 Hosting and maintenance plan

This dataset will be hosted on MIT servers in perpetuity at https://objectnet.dev/flash/ with a backup on dropbox. Our dataset collection toolbox is hosted publicly on github at https://github.com/dmayo/MVT-difficulty. A datacard for this dataset will be available at https://objectnet.dev/flash/datacard.

6.2 Object classes

Test subjects were presented with images from 50 possible object classes and asked to select which object they saw. The 50 classes were hand-picked to minimize similarity between classes that could be confusing for experiment subjects. The object classes were:
Band-Aid, T-shirt, backpack, banana, cleaver, clothes iron, coffee mug, computer mouse, digital watch, doormat, dumbbell, envelope, hair dryer, hammer, lampshade, lemon, lipstick, match, mobile phone, necklace, padlock, paintbrush, paper towel, park bench, pill bottle, pillow, plastic bag, plunger, power drill, printer, racket, ruler, safety pin, salt shaker, sandal, screw, shovel, space heater, spatula, speaker, strainer, sunglasses, teddy bear, television, umbrella, vase, wallet, waste container, water bottle, whistle

6.3 Image Selection

After choosing object classes, we selected images for the experiment. We used all 50 images belonging to a class in the ImageNet Validation set with no additional selection step. For ObjectNet, we collected bounding box data for the images, and then randomly selected 50 images per class such that when cropped to the bounding box, the object in the image was centered and clear.

6.4 Image cropping procedure

1. We draw a bounding box around the object (we use existing bounding boxes for the ImageNet validation set and collect our own bounding boxes for ObjectNet from MTurk).
2. We initialize the cropping box to be the bounding box.
3. If the cropping box does not form a square, we extend the shorter side of the rectangular cropping box to form a square. If the image is not large enough to extend the shorter side of the cropping box, we pad it with black pixels to form a square.
4. We crop using the cropping box for the image. The cropped image will be a square.
5. We resize the cropped image to be 224x224 pixels.

6.5 Mask generation

The masks were generated following the procedure used by [41]. Specifically, a Fourier transform was applied to each image to obtain the magnitude and phase components. Then, a random array with elements sampled uniformly from [0, 1] was added to the image phase component after which the magnitude and phase components were recombined via an inverse Fourier transform to produce the mask. Each image was paired with its particular phase-scrambled mask in the experiments.

6.6 Experiment Procedure and Payment

Participants both in the lab and on Mechanical Turk were presented with a document informing them of the purpose, privacy, and risks associated with the experiment and soliciting their consent to participate (see fig. 10). Participants were then instructed as to how to carry out the experiment and were shown an example video as well as the list of image classes for their review before beginning. They were informed that they would not need to memorize the classes as the classes would be shown

Table 1: Dataset statistics

number of responses	200,382
number of images	4,771
number of presentaiton durations	6
number of response per image	42
number of objectnet images	2415
number of imagenet images	2356
number of participants	2647

to them after each video. Participants were also encouraged to take breaks should they feel fatigued or otherwise uncomfortable. Example instructions are shown in fig. 11

After giving consent and reading the experiment overview. participants then completed two calibration steps for to estimate the size of their monitor and their distance from the screen for us to then size the videos appropriately to 8 degrees of visual angle. First, the participants are shown an image of a credit card and are asked to use a card of their own to adjust a slider to change the size of the card on the screen to the size of their card. Since credit cards are the same size around the world, this allows us to measure the pixel-to-inches ratio of the participant's monitor. Next, the participant completes a blind-spot test [37] that allows us to estimate the distance they are sitting from their screen. Together, these two measurements are sufficient to compute the desired video eccentricity. See fig. 12 for images of the calibration steps.

The estimated hourly wage for participants on Mechanical Turk and in the lab was $10/hr and $20/hr respectively with approximately $15,000 spent in total on participant compensation.

6.7 In-Lab Experiment Results

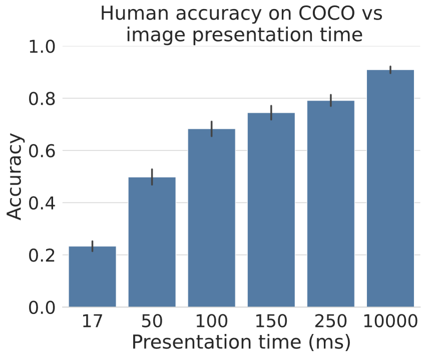
To corroborate our Amazon Mechanical Turk results, we selected 200 images shown to Turk workers to conduct the same experiment in a controlled laboratory setting. 12 individuals came to participate in the experiment in which they viewed and responded to all 200 images on our 144Hz refresh rate monitor with 1ms gray-to-gray time. After conducting the experiment, 3 individuals had seen each image at each of the 4 presentation times. When compared to the MTurk results for those same 200 images, the comparison is much as we would expect. The In-Lab accuracy with shortest image duration (17ms) is less than on MTurk which can likely be contributed to the use of our new, high refresh-rate monitor in the controlled environment. It is likely that MTurk workers' personal computers differ in their graphics presentation abilities which may result in the image being visible for slightly greater than 17ms on some monitors. On the other end, the in-lab experiments reported higher accuracy at the longest image duration (10s) which is also unsurprising as the in-lab participants completed the task in a controlled environment with no distractions and are likely more inclined to take the task seriously and stay focused. The results show no significant differences in accuracy at the intermediate image durations. See fig. 4 for a side-by-side comparison between MTurk and In-Lab results.

6.8 Dataset statistics

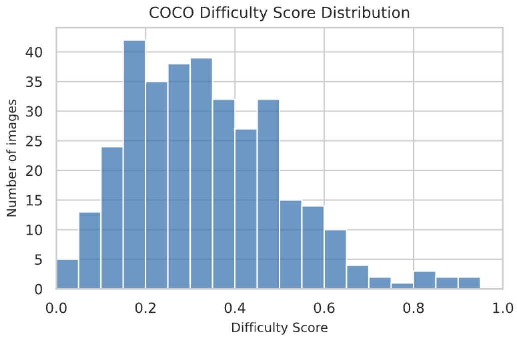
We collected 42 human responses for each of 5,000 images (2,500 from ImageNet and 2,500 from ObjectNet). After reviewing response, 229 images were removed due to either being unrecognizable, mislabeled, or having been seen by the same worker twice despite safeguards in place to disallow it. Additional dataset statistics are listed in table 1.

6.9 Preliminary COCO MVT Results

To bolster our claims about the difficulty of current datasets, we conducted the MTurk MVT experiment on a small subset of the COCO dataset. As COCO is a more visually complex dataset than



(a) Human accuracy per timing on COCO.



(b) Difficulty score for COCO images normalized by number of collected responses per image. A difficulty score of 1.0 here correspondes to 42 in figure 1.

many single object classification datasets, it provides a good litmus test for how our conclusions generalize to other kinds of datasets.

6.9.1 Image selection

To maximize the utility of our results for both computer science and neuroscience research we selected 732 images from the Natural Scenes Dataset[42], a subset of COCO for which fMRI data was collected from human participants. We used the image crops used in the NSD experiments. These crops ensure that the image is square.

6.9.2 Image classes

We selected a set of 41 classes such that no image contained more than one of the classes.

6.9.3 Experimental procedure

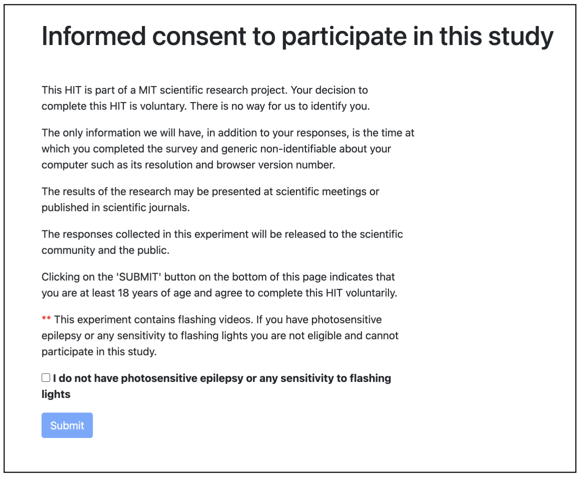
We conducted the MVT experiment as described in the text, asking participants to perform a 1-of-41 forced-choice single-object recognition task.

Below, we present preliminary results computed on 340 of the images. The final manuscript will include all 732 images. The results in section 6.9.3 are striking in their similarity to those presented for ImageNet and ObjectNet in the main text. The accuracy of human workers at each of the presentation times while performing the COCO classification task is almost the same as that of the ImageNet experiments. Given that both ImageNet and COCO originate from the same online pool of images, this is to be expected.

Similarly, the difficulty scores of COCO images (the counterpart of fig. 1) is skewed toward easy images, perhaps more so than either ImageNet or ObjectNet. These results indicate that our conclusions about the difficulty distributions of individual object recognition tasks in vision datasets generalizes. Of course, COCO has images where multiple object classes are present, which involves visual search in addition to recognizing individual image instances, but, for the quantity that we measure here, how hard are objects themselves to recognize, it COCO and ImageNet are essentially the same.

6.10 Finetuned Models

Here we list details regarding training/finetuning procedures for the model results reported in the paper.



Informed consent to participate in this study

This HIT is part of a MIT scientific research project. Your decision to complete this HIT is voluntary. There is no way for us to identify you.

The only information we will have, in addition to your responses, is the time at which you completed the survey and generic non-identifiable about your computer such as its resolution and browser version number.

The results of the research may be presented at scientific meetings or published in scientific journals.

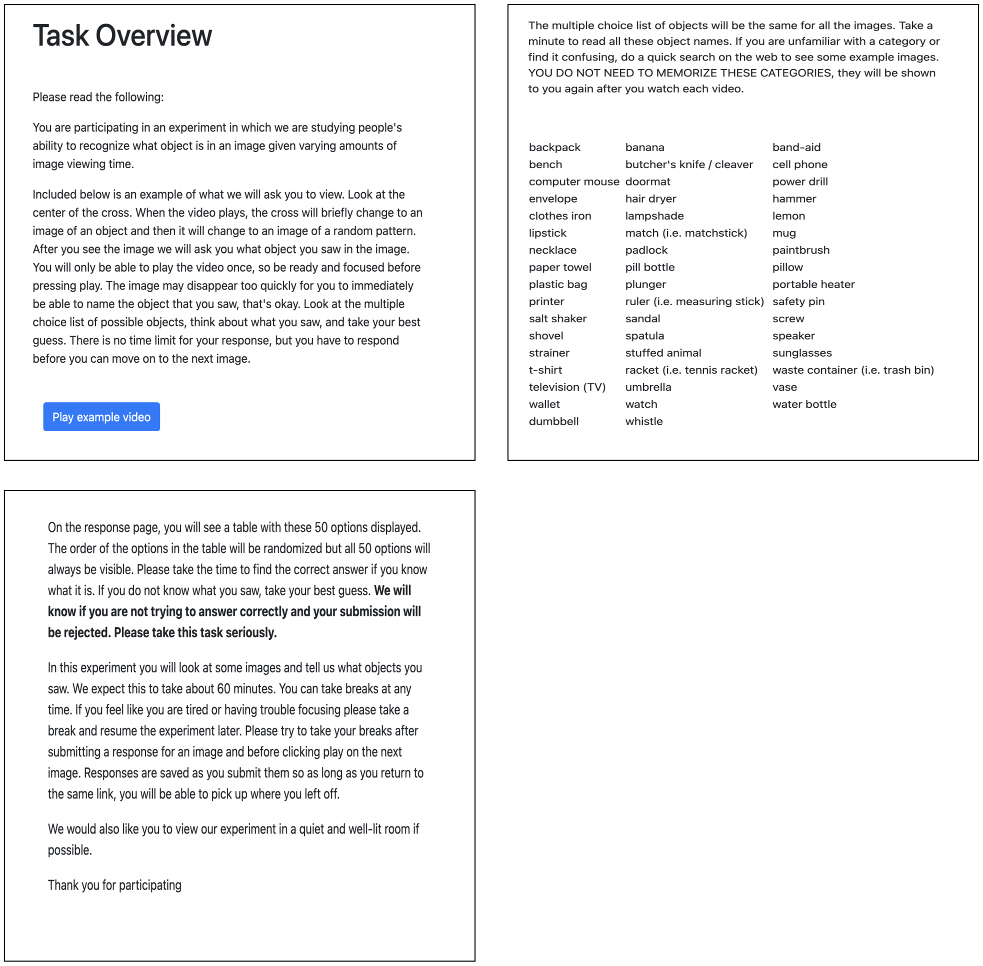
The responses collected in this experiment will be released to the scientific community and the public.

Clicking on the 'SUBMIT' button on the bottom of this page indicates that you are at least 18 years of age and agree to complete this HIT voluntarily.

** This experiment contains flashing videos. If you have photosensitive epilepsy or any sensitivity to flashing lights you are not eligible and cannot participate in this study.

☐ **I do not have photosensitive epilepsy or any sensitivity to flashing lights**

Figure 10: Informed consent page shown to participants before beginning the experiment.



Task Overview

Please read the following:

You are participating in an experiment in which we are studying people's ability to recognize what object is in an image given varying amounts of image viewing time.

Included below is an example of what we will ask you to view. Look at the center of the cross. When the video plays, the cross will briefly change to an image of an object and then it will change to an image of a random pattern. After you see the image we will ask you what object you saw in the image. You will only be able to play the video once, so be ready and focused before pressing play. The image may disappear too quickly for you to immediately be able to name the object that you saw, that's okay. Look at the multiple choice list of possible objects, think about what you saw, and take your best guess. There is no time limit for your response, but you have to respond before you can move on to the next image.

The multiple choice list of objects will be the same for all the images. Take a minute to read all these object names. If you are unfamiliar with a category or find it confusing, do a quick search on the web to see some example images. YOU DO NOT NEED TO MEMORIZE THESE CATEGORIES, they will be shown to you again after you watch each video.

backpack
bench
computer mouse
envelope
clothes iron
lipstick
necklace
paper towel
plastic bag
printer
salt shaker
shovel
strainer
t-shirt
television (TV)
wallet
dumbbell
banana
butcher's knife / cleaver
doormat
hair dryer
lampshade
match (i.e. matchstick)
padlock
pill bottle
plunger
ruler (i.e. measuring stick)
sandal
spatula
stuffed animal
racket (i.e. tennis racket)
umbrella
watch
whistle
band-aid
cell phone
power drill
hammer
lemon
mug
paintbrush
pillow
portable heater
safety pin
screw
speaker
sunglasses
waste container (i.e. trash bin)
vase
water bottle

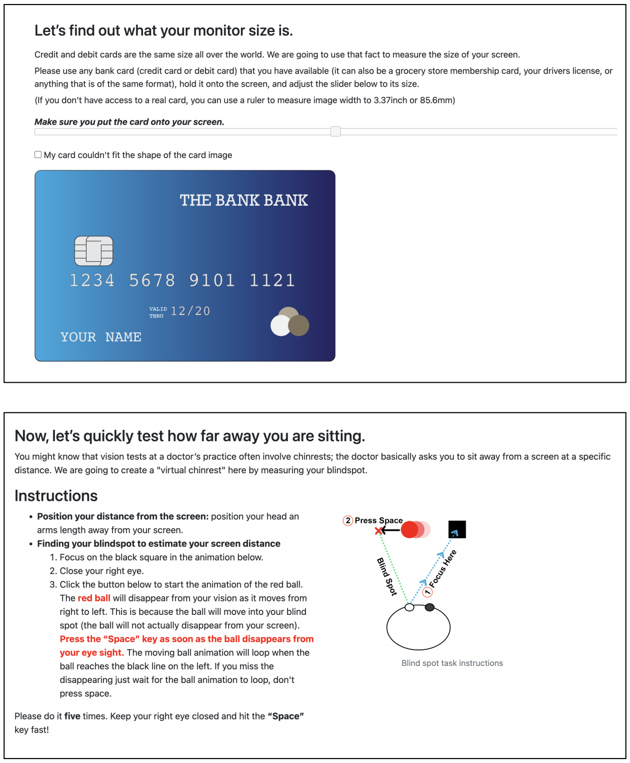
On the response page, you will see a table with these 50 options displayed. The order of the options in the table will be randomized but all 50 options will always be visible. Please take the time to find the correct answer if you know what it is. If you do not know what you saw, take your best guess. **We will know if you are not trying to answer correctly and your submission will be rejected. Please take this task seriously.**

In this experiment you will look at some images and tell us what objects you saw. We expect this to take about 60 minutes. You can take breaks at any time. If you feel like you are tired or having trouble focusing please take a break and resume the experiment later. Please try to take your breaks after submitting a response for an image and before clicking play on the next image. Responses are saved as you submit them so as long as you return to the same link, you will be able to pick up where you left off.

We would also like you to view our experiment in a quiet and well-lit room if possible.

Thank you for participating

Figure 11: Instructions given to participants before beginning the experiment.



Let’s find out what your monitor size is.

Credit and debit cards are the same size all over the world. We are going to use that fact to measure the size of your screen.

Please use any bank card (credit card or debit card) that you have available (it can also be a grocery store membership card, your drivers license, or anything that is of the same format), hold it onto the screen, and adjust the slider below to its size.

(If you don't have access to a real card, you can use a ruler to measure image width to 3.37inch or 85.6mm)

Make sure you put the card onto your screen.

☐ My card couldn't fit the shape of the card image

Now, let’s quickly test how far away you are sitting.

You might know that vision tests at a doctor’s practice often involve chinrests; the doctor basically asks you to sit away from a screen at a specific distance. We are going to create a "virtual chinrest" here by measuring your blindspot.

Instructions

- **Position your distance from the screen:** position your head an arms length away from your screen.
- **Finding your blindspot to estimate your screen distance**
  1. Focus on the black square in the animation below.
  2. Close your right eye.
  3. Click the button below to start the animation of the red ball. The **red ball** will disappear from your vision as it moves from right to left. This is because the ball will move into your blind spot (the ball will not actually disappear from your screen). **Press the “Space” key as soon as the ball disappears from your eye sight.** The moving ball animation will loop when the ball reaches the black line on the left. If you miss the disappearing just wait for the ball animation to loop, don't press space.

Please do it **five** times. Keep your right eye closed and hit the **“Space”** key fast!

Figure 12: Images of the experiment calibration steps. The credit card task was used to measure the pixel-to-inches ratio of the subject’s screen. The blind spot task provided an estimate of the subjects distance from their screen.

6.10.1 Model training procedure

Pretrained models weights were instantiated using publicly available model checkpoints, either using torchvision or found on the model’s source repository. The models—with the exception of CLIP—were then finetuned using subsets of the ImageNet training and validation sets containing only the 50 classes we chose to use in the psychophysics experiments. The models were finetuned for 90 epochs with an SGD optimizer and initial learning rate of 0.1 with momentum value of 0.9 and weight decay coefficient of 0.0001. The learning rate decayed by a factor of 2 every 9 epochs. Training, finetuning, and inference were performed on a cluster of 8 Nvidia TITAN RTX graphics cards.

6.10.2 Model Performance

We evaluate our finetuned models on the same cropped images used in our psychophysics experiments. See table 4 for model accuracy reports on the image difficulty reported in the paper and table 2 and table 3 for model performance on the full ImageNet and ObjectNet subsets of the experiment images.

6.11 Metric calculation procedure

In this section, we go through the details in computing c-score, prediction depth, and adversarial robustness for our experiment images.

6.11.1 C-score

C-score [7] identifies individual image difficulty by characterizing the expected accuracy or a held-out image given training sets of varying size sampled from the data distribution. In particular, c-score is the frequency of classifying an example correctly when it is omitted from the training set. However,

Table 2: Model accuracy on ImageNet per MVT subset. Models are named to include architecture, training objective, and training dataset where appropriate. ResNet-X-Y% indicates a ResNet with depth X and trained on a random Y% subset of the ImageNet-1k dataset. Model names ending in 21k were pretrained on ImageNet-21k. All other models with the exception of SWSL and CLIP models were pre-trained on the full ImageNet-1k dataset.

Subset	<= 17	<= 50	<= 100	<= 150	<= 250	<= 10000
ResNet-18	94.4	91.8	81.1	77.2	61.3	49.0
ResNet-18-20%	81.9	77.2	63.7	58.2	39.8	34.9
ResNet-18-40%	84.4	85.0	67.8	63.5	50.5	46.2
ResNet-18-60%	87.5	88.3	72.6	70.4	53.8	46.2
ResNet-18-80%	88.1	86.5	76.7	68.8	54.8	48.6
ResNet-50	94.4	95.5	85.2	84.7	79.6	64.7
ResNet-50-20%	86.9	82.9	72.2	65.1	48.4	41.0
ResNet-50-40%	93.8	89.1	74.1	74.1	60.2	48.2
ResNet-50-60%	91.2	90.2	78.5	79.4	60.2	59.4
ResNet-50-80%	90.6	91.5	83.0	80.4	68.8	59.0
ResNet-101	95.0	95.2	90.0	87.8	79.6	71.5
ResNet-101-20%	86.2	85.4	70.0	66.1	50.5	48.2
ResNet-101-40%	90.6	90.0	78.1	77.8	63.4	50.6
ResNet-101-60%	93.1	89.9	84.4	77.2	62.4	59.0
ResNet-101-80%	92.5	94.1	83.0	82.5	64.5	61.8
ResNet-152	93.8	96.4	93.7	86.8	78.5	72.7
ResNet-152-20%	86.9	84.4	73.0	71.4	52.7	44.2
ResNet-152-40%	93.1	88.4	76.3	76.7	62.4	52.6
ResNet-152-60%	93.1	90.4	82.2	78.8	66.7	59.4
ResNet-152-80%	90.6	91.9	86.3	85.7	76.3	60.6
CORNet-S	93.8	92.6	81.9	78.8	58.1	52.2
VOneNet-Resnet50	93.8	94.4	84.4	82.5	67.7	56.6
VOneNet-CORNet-S	91.9	92.3	82.2	77.2	63.4	53.4
VGG-19	91.9	90.2	80.7	79.4	62.4	55.4
Noisy Student (EfficientNet-L2)	95.0	93.3	87.8	86.8	68.8	65.5
DenseNet-121	94.4	93.3	83.3	80.4	72.0	58.6
MSDNet Classifier 0	78.8	76.0	60.0	54.0	40.9	33.7
MSDNet Classifier 1	89.4	86.2	73.7	67.7	53.8	45.8
MSDNet Classifier 2	91.9	89.9	77.8	72.5	62.4	51.4
MSDNet Classifier 3	91.9	90.4	79.3	69.8	63.4	51.4
MSDNet Classifier 4	94.4	91.3	79.3	78.8	62.4	52.2
SimCLR ResNet50	88.1	86.3	73.7	69.8	60.2	54.6
SimCLR ResNet101	93.1	89.6	79.3	83.6	72.0	57.8
SimCLR ResNet152	93.8	92.1	83.7	81.0	72.0	63.9
CLIP-ViT-B/32	95.6	90.8	79.3	74.6	67.7	48.6
CLIP-ViT-B/16	97.5	94.7	83.3	81.0	80.6	52.2
CLIP-ViT-L/14	98.1	97.1	92.6	91.0	86.0	72.3
CLIP-ViT-L/14@336px	98.1	96.9	91.5	92.6	89.2	73.9
CLIP-ResNet-50	92.5	84.4	69.6	67.7	55.9	34.5
CLIP-ResNet-101	94.4	88.1	71.9	68.8	67.7	41.0
CLIP-ResNet-50x4	93.8	88.8	75.6	72.5	71.0	41.8
CLIP-ResNet-50x16	94.4	91.5	81.1	77.2	68.8	43.8
CLIP-ResNet-50x64	98.8	95.9	87.0	85.2	77.4	59.0
EfficientNet-S	91.2	92.5	84.1	78.3	64.5	62.2
EfficientNet-M	90.6	91.5	80.7	73.5	69.9	61.4
EfficientNet-L	95.0	93.0	87.4	83.6	75.3	64.3
EfficientNet-S-21	96.9	95.6	92.6	87.8	81.7	71.5
EfficientNet-M-21	97.5	97.0	93.7	88.9	84.9	72.3
EfficientNet-L-21	98.1	96.7	93.0	90.5	86.0	73.5
ViT-T/16	67.5	72.3	57.8	54.5	38.7	34.5
ViT-S/16	95.0	94.5	82.6	85.2	68.8	58.6
ViT-B/16	96.2	95.6	85.2	87.3	67.7	63.5
ViT-L/16	98.8	97.5	97.4	96.8	84.9	80.7
MoCo-V3	92.5	92.6	85.6	84.7	75.3	64.7
SWSL-ResNext101-32x16d	96.9	98.1	97.4	95.8	87.1	85.5
SWSL-ResNet50	96.2	97.7	96.7	95.2	84.9	77.9
MAE-ViT-B/16	94.4	95.6	88.9	89.9	77.4	75.1

computing c-score for each image by brute force is computationally infeasible since we must train a separate model for each image. Instead, we computed the learning speed proxy as recommended by the authors. Learning speed measures the epoch at which an image is correctly classified by a model. Intuitively, a training example that is consistent with the training set should be learned quickly because the gradient step for all consistent examples should be similar. The authors found high Spearman rank correlation between c-score and cumulative learning speed based proxies.

We trained a ResNet-50 [43] from scratch on ImageNet1k [16] for 90 epochs with an SGD optimizer and initial learning rate of 0.1 with momentum value of 0.9 and weight decay coefficient of 0.0001. The learning rate decayed by a factor of 2 every 9 epochs and the batch size was 256. The standard ImageNet transforms were applied to all images, and the network was initialized randomly. We then evaluated our experiment images at each epoch and used the average of correct predictions as an estimated c-score for each image. fig. 13 shows the average c-scores for ImageNet and ObjectNet

Table 3: Model accuracy on ObjectNet per recognition time subset.

Subset	<= 17	<= 50	<= 100	<= 150	<= 250	<= 10000
ResNet-18	76.1	65.1	49.1	41.2	25.3	20.6
ResNet-18-20%	46.2	44.8	29.6	27.5	11.5	12.5
ResNet-18-40%	58.1	53.8	43.0	35.2	20.7	16.2
ResNet-18-60%	67.5	60.0	43.3	37.9	19.5	17.4
ResNet-18-80%	66.7	63.4	45.7	37.4	26.4	17.1
ResNet-50	80.3	79.7	62.9	53.3	44.8	27.0
ResNet-50-20%	58.1	51.1	36.4	35.2	24.1	15.7
ResNet-50-40%	70.1	61.7	45.4	42.9	29.9	20.3
ResNet-50-60%	70.9	70.1	54.0	45.1	33.3	19.7
ResNet-50-80%	76.9	70.4	53.3	49.5	31.0	22.0
ResNet-101	86.3	81.0	68.7	54.9	47.1	29.6
ResNet-101-20%	50.4	53.3	41.2	33.0	21.8	13.6
ResNet-101-40%	66.7	61.5	47.4	35.7	27.6	18.3
ResNet-101-60%	76.9	70.3	52.9	46.7	36.8	22.6
ResNet-101-80%	75.2	75.2	62.9	46.2	34.5	25.2
ResNet-152	85.5	83.8	68.7	60.4	46.0	30.7
ResNet-152-20%	58.1	53.7	39.2	34.6	19.5	13.6
ResNet-152-40%	66.7	65.1	49.5	42.9	25.3	19.1
ResNet-152-60%	72.6	68.4	57.0	41.8	35.6	22.6
ResNet-152-80%	74.4	73.5	55.7	50.0	35.6	24.6
CORNet-S	75.2	71.5	53.6	45.1	36.8	20.3
VOneNet-Resnet50	77.8	75.7	59.1	45.6	35.6	20.9
VOneNet-CORNet-S	72.6	67.0	51.9	42.9	31.0	16.5
VGG-19	76.1	66.3	50.9	46.7	34.5	18.3
Noisy Student (EfficientNet-L2)	76.9	68.7	54.3	45.1	26.4	20.9
DenseNet-121	77.8	74.9	57.0	49.5	33.3	22.3
MSDNet Classifier 0	45.3	39.0	31.3	28.0	17.2	10.7
MSDNet Classifier 1	62.4	56.4	41.9	36.3	23.0	15.9
MSDNet Classifier 2	70.9	64.3	52.9	44.0	28.7	22.3
MSDNet Classifier 3	76.9	68.7	51.5	46.7	29.9	21.7
MSDNet Classifier 4	73.5	70.9	51.5	47.3	37.9	22.3
SimCLR ResNet50	60.7	61.0	49.8	45.6	27.6	17.4
SimCLR ResNet101	75.2	70.3	59.5	55.5	32.2	25.8
SimCLR ResNet152	78.6	70.3	57.0	55.5	35.6	28.7
CLIP-ViT-B/32	88.9	80.5	61.2	61.0	43.7	33.6
CLIP-ViT-B/16	92.3	88.2	78.0	69.8	50.6	48.4
CLIP-ViT-L/14	97.4	93.8	88.7	81.3	78.2	70.1
CLIP-ViT-L/14@336px	96.6	94.2	91.1	85.2	80.5	70.1
CLIP-ResNet-50	78.6	73.5	58.1	54.9	34.5	27.5
CLIP-ResNet-101	86.3	76.9	63.6	58.2	41.4	32.5
CLIP-ResNet-50x4	83.8	79.3	67.4	64.8	43.7	36.8
CLIP-ResNet-50x16	92.3	85.0	74.2	69.2	52.9	48.7
CLIP-ResNet-50x64	89.7	90.3	84.2	81.9	69.0	55.9
EfficientNet-S	68.4	66.2	52.9	40.1	23.0	20.6
EfficientNet-M	71.8	66.3	47.8	39.0	20.7	18.8
EfficientNet-L	75.2	71.5	54.0	45.1	31.0	23.2
EfficientNet-S-21	94.0	83.9	72.9	64.8	41.4	30.4
EfficientNet-M-21	88.9	87.5	71.1	67.0	46.0	34.2
EfficientNet-L-21	89.7	86.2	70.1	68.7	50.6	36.5
ViT-T/16	43.6	43.4	29.6	25.3	10.3	10.7
ViT-S/16	83.8	76.8	57.7	52.2	31.0	23.5
ViT-B/16	86.3	80.2	65.6	58.2	37.9	26.4
ViT-L/16	96.6	92.5	84.9	81.3	58.6	44.9
MoCo-V3	82.9	73.3	59.1	54.9	34.5	24.6
SWSL-ResNext101-32x16d	94.0	94.5	89.3	85.7	62.1	57.4
SWSL-ResNet50	94.0	89.4	83.8	72.0	52.9	47.2
MAE-ViT-B/16	84.6	82.7	69.4	63.2	37.9	32.5

experiment images split by whether the ResNet-50 correctly predicted the image. C-score serves as an efficient predictor for human recognition difficulty only for images classified by the model in both ImageNet and ObjectNet. C-scores for images misclassified by the model do not reveal information about the human recognition difficulty and remain consistently low across all difficulty subsets.

6.11.2 Prediction depth

Prediction depth [8] represents the number of hidden layers after which the network's final prediction is already determined. The authors showed that prediction depth is larger for examples that visually appear to be more difficult and is consistent between architectures and random seeds.

We trained a linear decoder at the end of each block of a ResNet-50 on the 50 experiment classes using the ImageNet training and validation set. We used the same ResNet-50 used to calculate c-scores to ensure consistency of our results. There are 16 convolutional layers in a ResNet-50; and each linear decoder follows a convolution layer and consists of a pooling layer, flatten layer, and fully-connected layer. We use the same hyperparameters as section 6.11.1 and only updated the weights of the linear decoder.

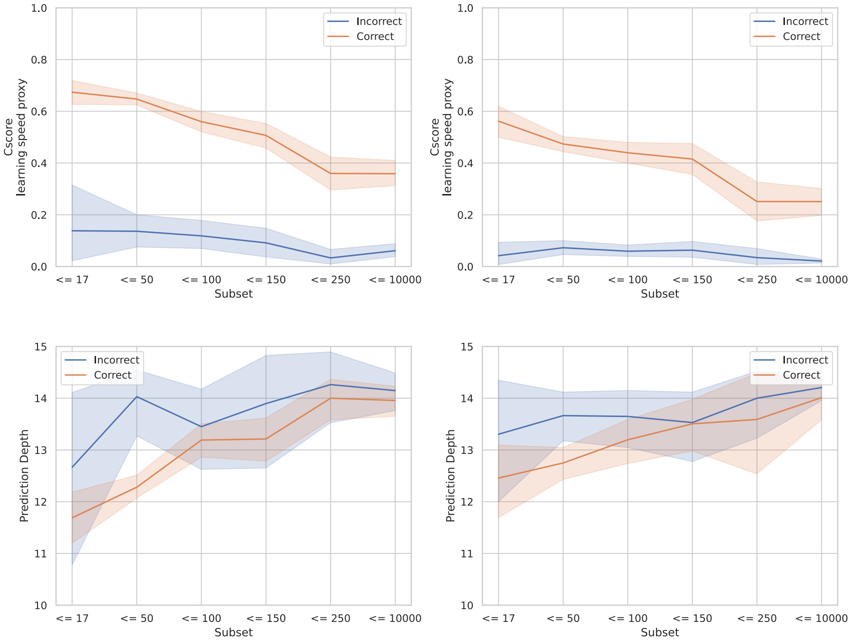


Figure 13: **Top**: left and right are average c-score over subsets for experiment ImageNet and ObjectNet images respectively. Orange shows the images that are correctly predicted by the ResNet-50 while blue shows the images that are incorrectly predicted. **Bottom**: prediction depth plots shown in the same way as top.

A prediction is defined to be made at depth $L = l$ if the linear classifier after layer $L = l - 1$ is different from the network's final prediction, but the classification of the linear decoder after every layer $L \geq l$ are equal to the final classification of the network. Images classified by all decoders are said to be predicted at layer 0. Note that prediction depth is independent of whether the final prediction is correct or not. It measures the layer at which an image's prediction converges.

Figure 13 shows the average c-scores for ImageNet and ObjectNet experiment images split by whether the ResNet-50 correctly predicted the image. Like c-score, prediction depth serves as an efficient predictor for human recognition difficulty only for images classified by the model in both ImageNet and ObjectNet.

6.11.3 Adversarial robustness

We measured an image's distance to the decision boundary of a network using fast gradient sign method (FGSM) [9]. FGSM creates an modified example that maximizes the loss using the gradients of loss with respect to the input image:

$$mod_x = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

where adv_x is the modified image, x is the original image, y is the original input label, ϵ is a multiplier adjusted accordingly to control the size of modification step, θ is the model parameters, and J is the loss function. Note that gradients are taken with respect to the input image, and model parameters remain constant.

For an image classified by a model, we define its distance to the closest decision boundary of the model as the minimum ϵ needed for the model to misclassify the modified image. On the other hand, for an image misclassified by a model, we define its distance to the closest decision boundary of the model as the minimum ϵ needed for the model to classify the modified image.

We used the same ResNet-50 used to calculate c-scores to ensure consistency of our results. We finetuned the ResNet-50 on the 50 experiment classes using the ImageNet training and validation set. We used the same hyperparameters as section 6.11.1 and only updated the weights of the final pooling, flatten, and fully-connected layer. We used this finetuned ResNet-50 as the backbone for adversarial perturbation and correction.

While perturbing each classified image, we searched for the smallest ϵ, from 0 to 0.02 incrementing by 1.25e-5 and from 0.02 to 2.5 incrementing by 0.005, that would result in a misclassification. We only applied only one gradient step when perturbing. While correcting each misclassified image, we searched for the smallest ϵ, from 0 to 0.001 incrementing by 1.25e-6 and from 0.001 to 0.05 incrementing by 1.25e-5. We applied two gradient steps when correcting because correction requires finer and more steps.

Note that the search range depends on the backbone model and the dataset. One must choose them through manual trial-and-errors to yield interesting and significant results. Recall that after removing images that were incorrectly annotated, incorrectly cropped, etc section 3, we reduced to 4,771 images from the original 5,000. Of these, 3,296 and 1,475 images were classified and misclassified by the finetuned ResNet-50 respectively. We were not able to find an ϵ for every image while perturbing and correcting in the corresponding search range. We omitted these images in our analysis. We were able to successfully perturb 2,815 out of 3,296 classified images and correct 1,114 out of 1,475 misclassified images.

We hypothesized that difficult images that are classified and misclassified would be closer and further from the decision boundary respectively. fig. 8 confirms the prior hypothesis. We could not confirm the latter hypothesis due to the smaller number of misclassified images across all subsets, as shown through the higher error bars in fig. 14

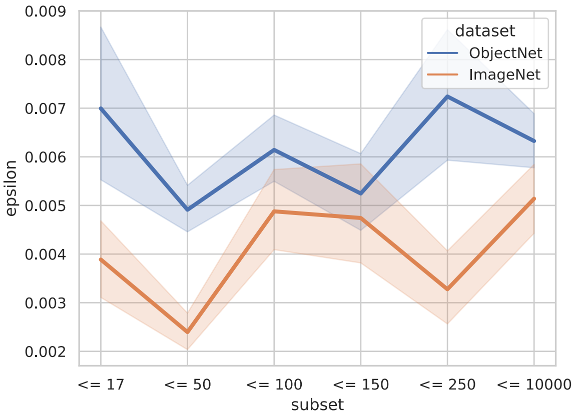


Figure 14: Average ϵ magnitude required to correct misclassified images back to their correct class per subset

6.12 What factors effect MVT? Imagenet-x analysis

We found no clear trends across MVT subsets for the 16 dimensions labeled in the imagenet-x dataset. The results of our analysis can be found in table 5.

6.13 Constructing a metric for image difficulty

We propose two metrics:

1. Difficulty score which provides an exact ranking from most difficult to recognize to least difficult to recognize based on each response
2. six minimum viewing time (MVT) subsets that quantify the minimum amount of time required for the majority of participants to reliably recognize an image.

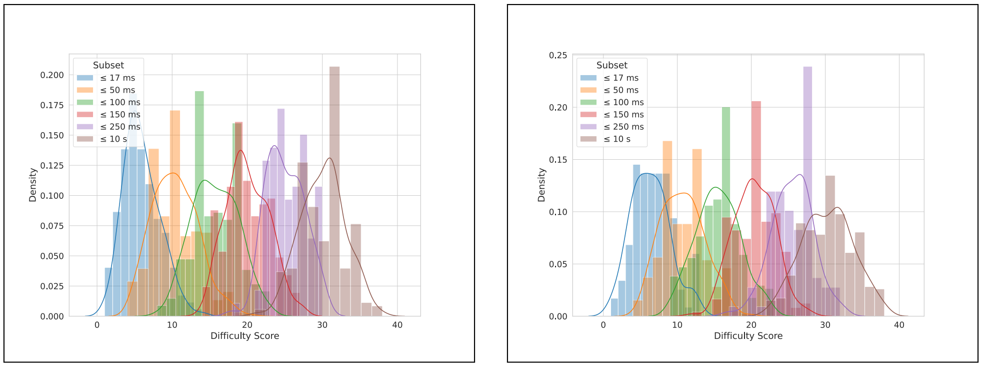


Figure 15: Distribution of difficulty score for each MVT subsets in ImageNet (left) and ObjectNet (right).

Difficulty score is a value from 0 to 42 that represents the number of incorrect predictions given by participants in our experiment across all timings for a particular image. Each image in our experiment was seen an equal number of times per timing and and only rarely were images that were recognizable at shorter timings also recognizable at longer timings. This results in a low difficulty score indicating that an image is easy to recognize and a high difficulty score indicating that an image is hard to recognize. These scores correlate well with the MVT difficulty subsets as shown in fig. 15. Difficulty score varies significantly by object class as well (see fig. 16).

6.14 Difficulty score distribution by object class

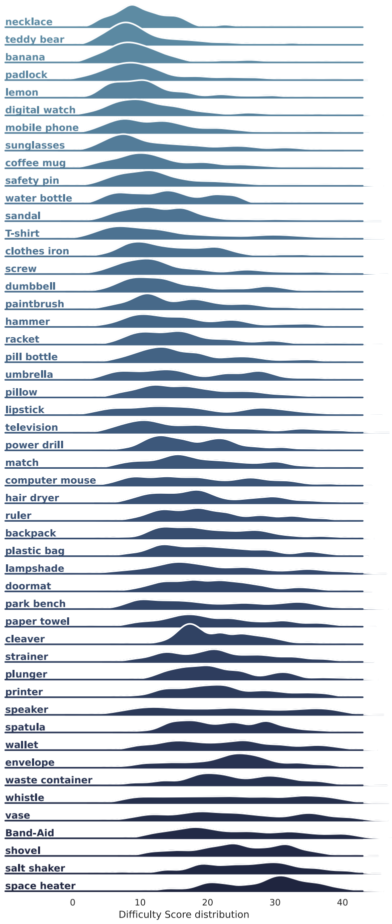


Figure 16: Difficulty distribution by object class sorted in order of increasing mean

Table 4: Model accuracy on the ImageNet and ObjectNet subsets of our 4,771 images.

ResNet-18	80.4	48.8
ResNet-18-20%	65.0	31.1
ResNet-18-40%	71.5	39.9
ResNet-18-60%	75.3	43.6
ResNet-18-80%	75.6	45.2
ResNet-50	87.1	60.0
ResNet-50-20%	71.0	38.3
ResNet-50-40%	77.7	46.4
ResNet-50-60%	80.4	51.6
ResNet-50-80%	82.9	52.6
ResNet-101	89.2	62.7
ResNet-101-20%	73.2	37.8
ResNet-101-40%	79.5	45.0
ResNet-101-60%	81.9	52.7
ResNet-101-80%	84.6	56.7
ResNet-152	90.3	64.5
ResNet-152-20%	73.4	38.5
ResNet-152-40%	78.7	47.9
ResNet-152-60%	82.3	51.4
ResNet-152-80%	84.6	54.8
CORNet-S	81.6	52.6
VOneNet-Resnet50	84.0	54.8
VOneNet-CORNet-S	81.3	48.8
VGG-19	81.1	50.2
Noisy Student (EfficientNet-L2)	86.2	51.2
DenseNet-121	83.7	55.7
MSDNet Classifier 0	62.8	29.3
MSDNet Classifier 1	74.6	41.6
MSDNet Classifier 2	78.6	49.6
MSDNet Classifier 3	78.9	51.6
MSDNet Classifier 4	81.0	52.8
SimCLR ResNet50	76.4	46.1
SimCLR ResNet101	82.1	55.4
SimCLR ResNet152	84.3	56.0
CLIP-ViT-B/32	80.0	63.3
CLIP-ViT-B/16	84.5	73.6
CLIP-ViT-L/14	92.1	85.8
CLIP-ViT-L/14@336px	92.0	86.9
CLIP-ResNet-50	71.6	56.8
CLIP-ResNet-101	75.3	61.4
CLIP-ResNet-50x4	77.5	64.6
CLIP-ResNet-50x16	80.1	72.4
CLIP-ResNet-50x64	87.1	79.4
EfficientNet-S	83.1	48.6
EfficientNet-M	81.7	47.2
EfficientNet-L	85.8	52.7
EfficientNet-S-21	90.2	66.4
EfficientNet-M-21	91.4	68.2
EfficientNet-L-21	91.5	68.8
ViT-T/16	59.4	29.9
ViT-S/16	84.4	57.1
ViT-B/16	86.5	61.3
ViT-L/16	94.1	78.1
MoCo-V3	85.5	56.6
SWSL-ResNext101-32x16d	95.1	82.3
SWSL-ResNet50	93.4	75.4
MAE-ViT-B/16	89.5	65.2

Table 5: ImageNet-x factors as a % of MVT subset. Each table entry represents the percentage of the images in MVT subset (row) that were labeled as containing a feature (column). This analysis is over the ImageNet images in our dataset.

MVT subset	multiple objects	background	color	brighter	darker	style	larger	smaller
17 ms	0.00	20.69	22.76	0.69	0.69	0.00	0.00	0.00
50 ms	0.15	25.23	20.39	0.00	0.15	0.15	0.15	3.32
100 ms	0.00	28.23	16.13	0.00	0.00	0.00	0.00	5.65
150 ms	0.00	25.56	16.67	0.00	1.11	0.00	0.56	4.44
250 ms	0.00	29.89	12.64	0.00	0.00	0.00	0.00	3.45
10 sec	0.00	27.27	16.94	0.00	1.24	0.00	0.41	4.13

MVT subset	object blocking	person blocking	partial view	pattern	pose	shape	subcategory	texture
17 ms	0.00	0.00	0.69	27.59	21.38	3.45	1.38	0.69
50 ms	0.00	0.00	1.06	23.26	21.75	1.36	2.27	0.76
100 ms	0.40	0.00	1.61	20.56	20.97	2.42	3.63	0.40
150 ms	0.56	0.00	1.67	22.78	20.56	3.89	1.11	1.11
250 ms	0.00	1.15	4.60	22.99	19.54	1.15	3.45	1.15
10 sec	0.00	0.00	2.07	19.42	22.73	4.13	0.83	0.83

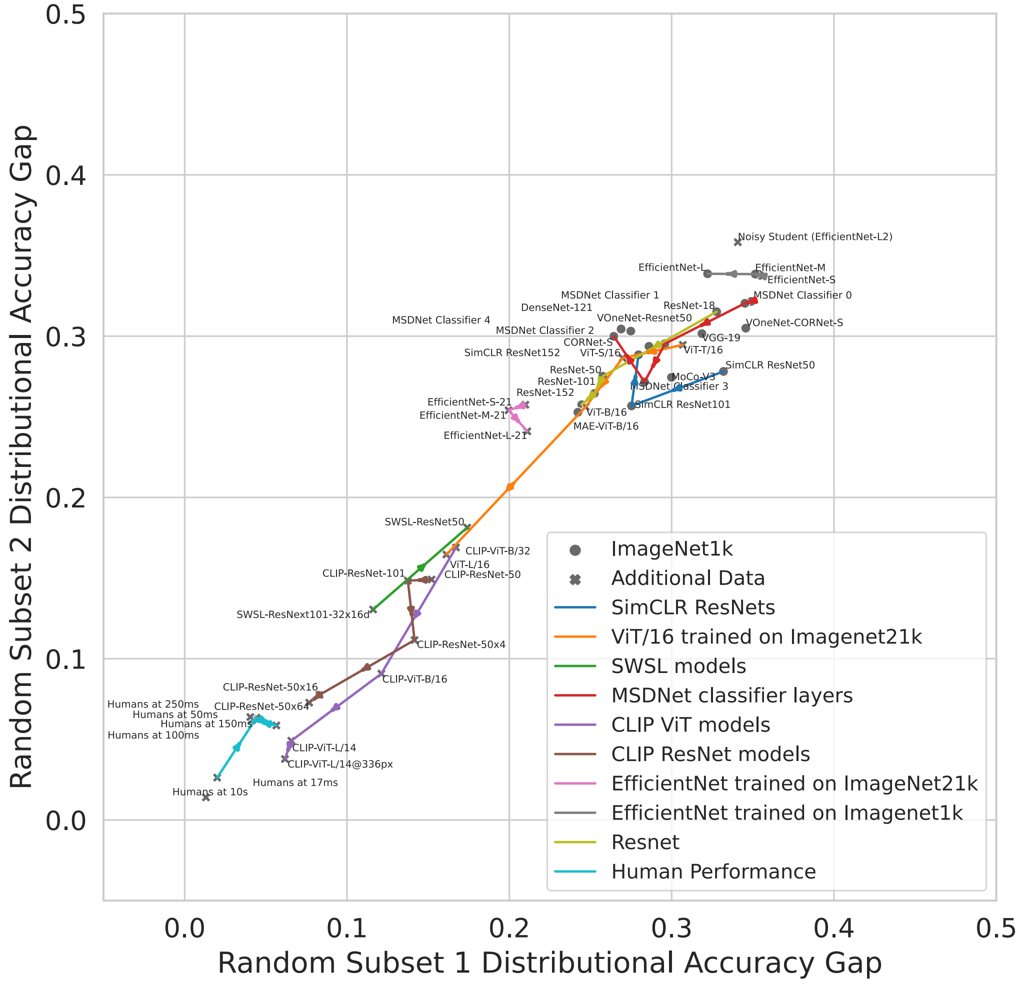


Figure 17: Robustness gap for our finetuned models on two randomly sampled subsets of our experiment data, balanced between ImageNet and ObjectNet. Lines connect model families with arrows pointing in direction of increasing model capacity. Compare with fig. 7.

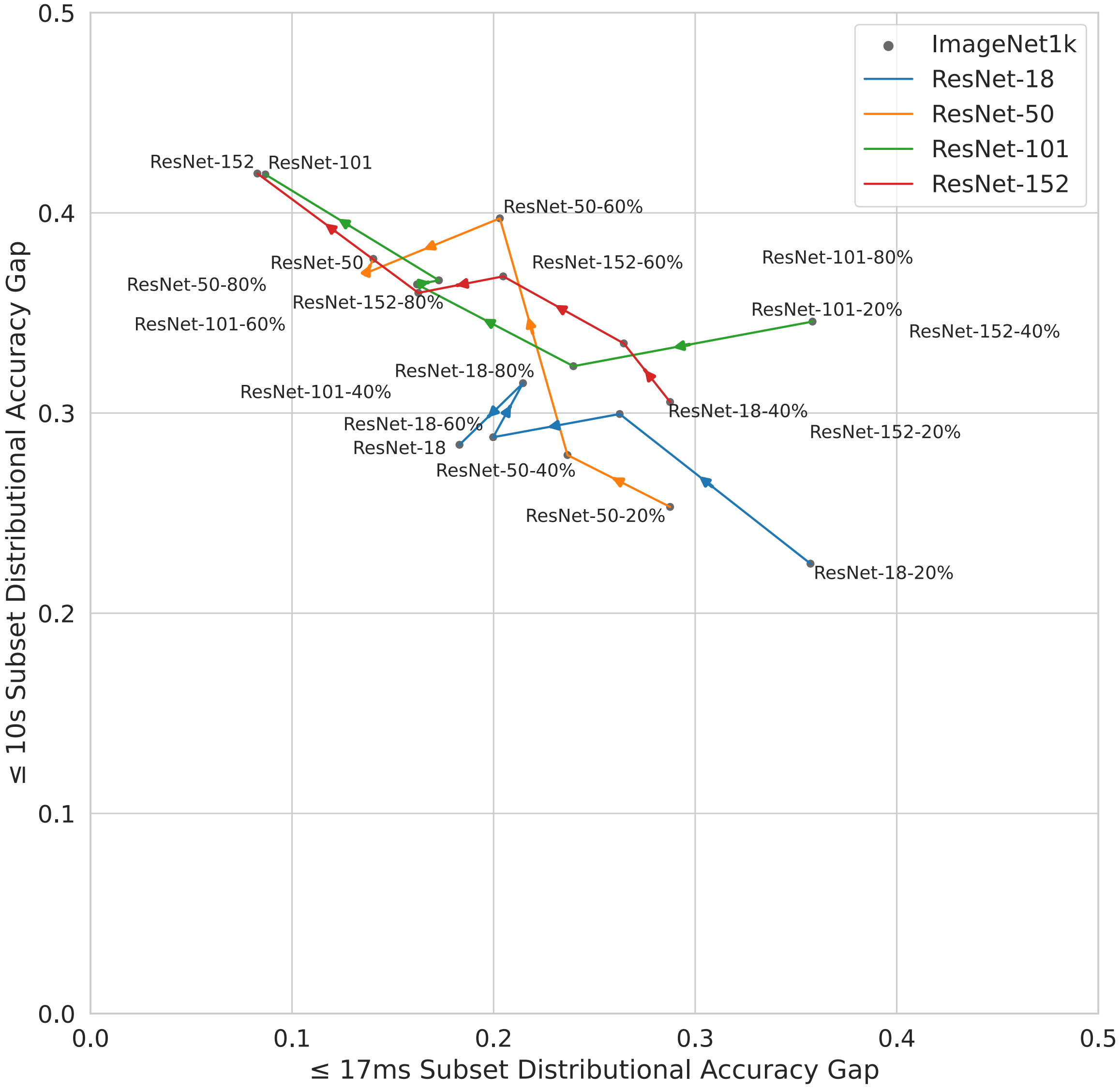


Figure 18: Robustness gap for our finetuned ResNets trained on varying percentages of the ImageNet training set. Lines connect the same architectures with arrows pointing in direction of increasing dataset percentage. Compare with fig. 7.

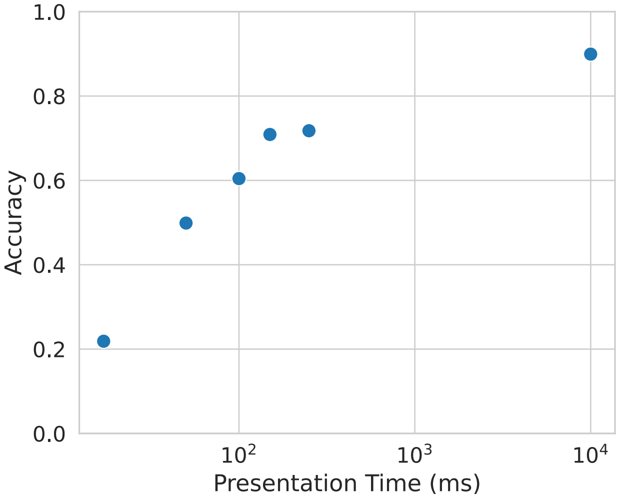


Figure 19: Human accuracy vs Image presentation time from Mechanical Turk results. Time is log-scale with a sigmoid fit

Massachusetts Institute of Technology
Committee on the Use of Humans as Experimental Subjects
77 Massachusetts Avenue Building E25-143B Cambridge, MA 02139-4307

Submission Date: Mar-29-2023

Title: E-4846, Multi-Modal Knowledge Tracking and Storytelling (mm-KTS) 80466CSDRP (in lab)
Principal Investigator: Barbu, Andrei
Department: CSAIL - PI Research
Faculty Sponsor: Katz, Boris
Start Date: Apr-01-2023
End Date: Apr-01-2026

Determination: Exempt

Your research activities meet the criteria for exemption as defined by Federal regulation 45 CFR 46 under the following:

Exempt Category 3 - Benign Behavioral Intervention

Research involving benign behavioral interventions where the study activities are limited to adults only and disclosure of the subjects' responses outside the research could not reasonably place the subjects at risk for criminal or civil liability or be damaging to the subjects' financial standing, employability, educational advancement, or reputation. Research does not involve deception or participants prospectively agree to the deception. 45 CFR 46.104(d)(3)

All members of the research team must adhere to the policies as outlined in the Investigator Responsibilities for Exempt Research. If the facts surrounding your evaluation change, you are required to submit a new Exempt Evaluation. Research records may be audited at any time during the conduct of the study.

email: couhes@mit.edu | phone: 617-253-6787 | website: couhes.mit.edu

Massachusetts Institute of Technology
Committee on the Use of Humans as Experimental Subjects
77 Massachusetts Avenue Building E25-143B Cambridge, MA 02139-4307

Submission Date: Sep-10-2019

Title: E-1632, Object recognition on Mechanical Turk
Principal Investigator: Katz, Boris
Department: Computer Science and Artificial Intelligence Laboratory
Faculty Sponsor:
Start Date: Sep-17-2019
End Date: Oct-01-2022

Determination: Exempt

Your research activities meet the criteria for exemption as defined by Federal regulation 45 CFR 46 under the following:

Exempt Category 3 - Benign Behavioral Intervention

Research involving benign behavioral interventions where the study activities are limited to adults only and disclosure of the subjects' responses outside the research could not reasonably place the subjects at risk for criminal or civil liability or be damaging to the subjects' financial standing, employability, educational advancement, or reputation. Research does not involve deception or participants prospectively agree to the deception. 45 CFR 46.104(d)(3)

All members of the research team must adhere to the policies as outlined in the Investigator Responsibilities for Exempt Research. If the facts surrounding your evaluation change, you are required to submit a new Exempt Evaluation. Research records may be audited at any time during the conduct of the study.

email: couhes@mit.edu | phone: 617-253-6787 | website: couhes.mit.edu