dimension $d = 256$. We found a margin of $m = \sqrt{10}$ experimentally resulted in the best encodings. A more complex architecture might yield stronger results, but to keep our initial design simple we use this CNN. We fine-tune this modified network to minimize the contrastive loss function described in Section 3.2, with a learning rate $\alpha = 1e^{-4}$, momentum $\mu = 0.9$, and batch size $b = 32$.

C ADAPTIVE ATTACKS/EVALUATION

C.1 PRADA Evaluation Parameters

PRADA sets a single parameter, the detection threshold, δ, according to a desired false positive rate (FPR). As described in [25], PRADA's FPR is evaluated by dividing a set of benign queries into chunks of 50 queries, inputting each chunk into PRADA, and computing the FPR as the fraction of chunks that trigger detection. Similar to tuning our scheme's threshold, we use the CIFAR10 training data as the set of benign queries, randomly divide it into 1,000 chunks, and tune the threshold to $\delta = 0.88$ to yield an FPR of 0.1%.

C.2 Additional Figures

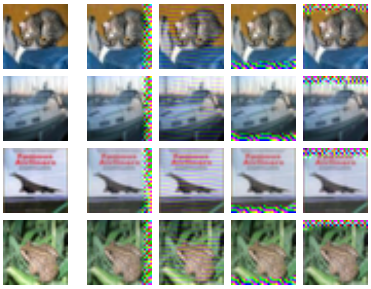

Figure 2: For each image in the test set (left) we show four possible images transformed by our auto-encoder (right) that have the same classification label but have high ℓ_2 distance from each other.

C.3 Extended Sybil Account Attack

Cyclic account reuse strategy. In the simplest Sybil attack strategy, an attacker creates C accounts and then, for a given attack instance, makes the i-th query through account $i \mod C$. The primary benefit of such a strategy is that any given detector is only allowed to observe every C-th query. An attacker may hope that this will increase the distance between consecutive queries seen by the detector (per account) and could require fewer accounts.

However, we find that this does not actually help. Upon investigation, we find the reason for this is that in practice, the randomness introduced through gradient estimation is roughly at the same scale as the *total* perturbation introduced throughout the process, and therefore observing sequential queries is much less important than observing queries from the same adversarial example generation process. However, this is just one attack strategy: a careful attacker may come up with a stronger one.

Near-optimal account reuse strategy. We now show that it is unlikely there exists a Sybil strategy which partitions a query sequence to reduce the detection rate by more than a factor of two.

We perform the following setup. We take the query sequence $\{x_i\}_{i=1}^{N}$ for a run of NES (using brightness for query blinding) to generate an adversarial example x_N and process each input with the encoder to obtain the embeddings $e_i = e(x_i)$. Then, we compute the pairwise distance between all pairs of points $d_{i,j} = \|e_i - e_j\|$.

We now ask: what is the largest set $S \subset [1..N]$ such that the k-nearest neighbor distance for each point is larger than the detection threshold. Because finding the largest set is NP Hard (it is easy to see through reduction from maximum clique), we approximate this quantity greedily, starting with the empty set and adding elements that are maximally far apart. On performing this experiment, we find that when a defense uses k-nearest neighbor distance, it is not possible to construct a set with $|S| > 2 \cdot k$. (For example, when $k = 5$ the largest set we can construct is $|S| = 9$; when $k = 25$ the largest set we can construct is $|S| = 44$.) Thus, if we conservatively assume that every set could be made this largest size, we are guaranteed one detection at least every $2k$ queries per Sybil account, reducing the detection rate by at most a factor of two. We now discuss further disadvantages of using Sybil accounts.

Drawback of Sybil Accounts. For anti-abuse reasons, most services already take measures to ensure that users do not create a large number of accounts. However, further, if a single user created multiple accounts and spread queries across each of these accounts, we believe this would only make it easier for the service provider to detect that one user was using multiple Sybil accounts. For example, if occasionally the service provider ever performed an across-user query-history analysis, it would be possible to discover the same user making highly-similar queries across different accounts.

D ENSEMBLE ADVERSARIAL TRAINING

We pre-train a ResNet50v1 on CIFAR-10 (accuracy 92.2%), then train it for 100 epochs on adversarial examples generated on an ensemble of different trained ResNets: ResNet44v1, ResNet56v2, and ResNet74v1. Examples were generated using FGSM [19] with $\epsilon = 0.05$, and each epoch's adversarial examples were generated from a randomly selected model from the ensemble and the defended model (one example generated per CIFAR-10 training image).

E VIDEO CLASSIFICATION

Videos were sampled at 30 frames per second, then downsized to $32x32x3$, yielding 7,000 frames per video. The threshold was tuned from $\delta = 1.44$ to $\delta = 0.7$. Decreasing the sampling rate by a factor of C decreased the FPR by a factor greater than C, (e.g. reducing the frame rate from 30fps to 15fps reduced the FPR by 0.6 instead of 0.5), as successive frames were less likely to have similar content.

The defender may need to further tune the threshold according to the natural workload that their system would expect (e.g. if the classification system is to be used for video frames, then a set of video frames, rather than static images, should be used to set the defense's threshold). The video classification setting may also be significantly expensive for users, as such frequent querying to an image classification API would incur high costs.