Evading Watermark Based Detection of AI Generated Content

Zhengyuan Jiang
Duke University
zhengyuan.jiang@duke.edu

Jinghuai Zhang
Duke University
jinghuai.zhang@duke.edu

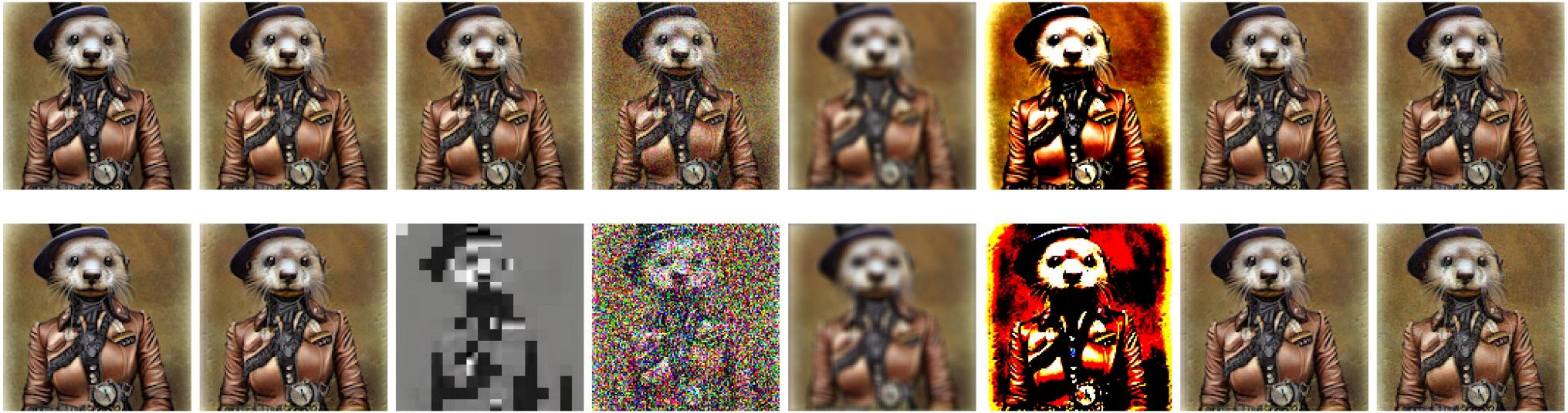
Neil Zhenqiang Gong
Duke University
neil.gong@duke.edu

Presented By: Samantha Tang

Image Watermarks

- Visible Watermarks
  - Dall-E
- Non-learning based Watermarks
  - Encoder and Decoder designed based on heuristics
  - Stable Diffusion uses Invisible Watermark
- Learning based Watermarks
  - Meta proposed to use
  - Encoder and Decoders are Neural Networks
  - HiDDeN and UDH
- In non-learning and learning, we have a watermark, encoder and decoder

Types of Post-Processing



(a) Original (b) Watermarked (c) JPEG (d) GN (e) GB (f) B/C (g) WEvade-W-II (h) WEvade-B-Q

Figure 1: Illustration of original image, watermarked image, and watermarked images post-processed by existing and our methods (last two columns) to evade detection. The watermarking method is HiDDeN. GN: Gaussian noise. GB: Gaussian blur. B/C: Brightness/Contrast. The encoder/decoder are trained via standard training (*first row*) or adversarial training (*second row*).

Learning-Based Watermarks are Not Robust Enough

- Previous studies do not cover robustness against adversarial post-processing
- WEvade developed to generate adversarial examples with small perturbations under multiple conditions

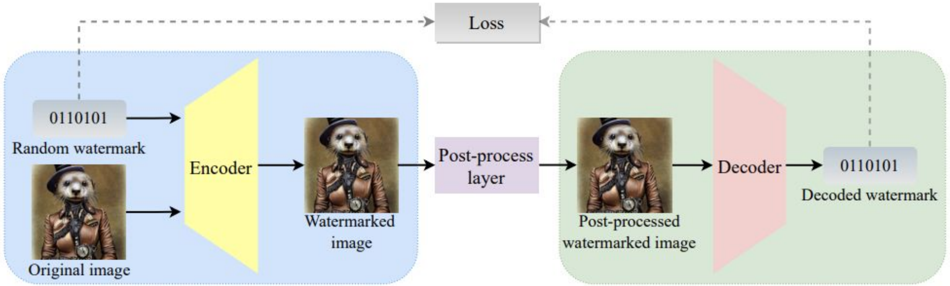
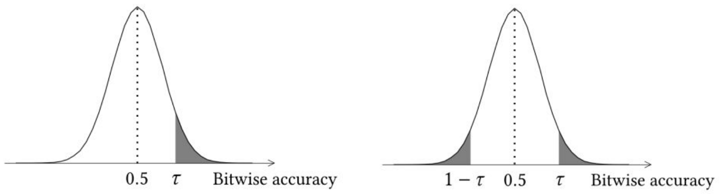


Figure 2: Illustration of training encoder and decoder in learning-based watermarking methods.

Standard and Adversarial Testing

- **Standard:**
  - Mini-batch training where a random watermark is sampled for an image I
  - Encoder makes the watermarked image
  - Decoder takes in this watermarked image and produces a watermark
  - Use SGD to minimize the loss $\sum_I loss(D(E(I,w_I)),w_I)$
- **Adversarial:**
  - For each image in the mini-batch, randomly select a post-processing method including WEvade
  - Same process as above but the loss has changed
  - Use SGD to minimize a loss function $\sum_I loss(D(E(I,w_I)+\delta_I),w_I)$, where δ_I is the perturbation



(a) Single-tail detector **(b) Double-tail detector**

Figure 3: Illustration of (a) single-tail detector and (b) double-tail detector with threshold τ. The bitwise accuracy of an original image I_o follows a binomial distribution divided by n, i.e., $BA(D(I_o), w) \sim B(n, 0.5)/n$. The area of the shaded region(s) is the *false positive rate (FPR)* of a detector.

Detectors

- Evaluations done via $BA(w1,w2)$, which is the fraction of bits that match in $w1$ and $w2$
- Single Tail Detector:
  - $BA(D(I),w) > \tau$
- Double Tail Detector:
  - watermarks decoded from original images have bitwise accuracy close to 0.5
  - watermarks decoded from watermarked images have large bitwise accuracy, e.g., close to 1
  - $BA(D(I),w) > \tau$ or $BA(D(I),w) < 1 - \tau$
- Note the concerns for FPR, select threshold with those in mind

$$FPR_s(\tau) = \Pr(BA(D(I_o), w) > \tau) = \Pr(m > n\tau) = \sum_{k=\lceil n\tau \rceil}^{n} \binom{n}{k} \frac{1}{2^n},$$

$$\tau^* = \arg\min_\tau \sum_{k=\lceil n\tau \rceil}^{n} \binom{n}{k} \frac{1}{2^n} < \eta.$$

$$FPR_d(\tau) = \Pr(BA(D(I_o), w) > \tau \text{ or } BA(D(I_o), w) < 1 - \tau) = \Pr(m > n\tau \text{ or } m < n - n\tau) = 2\sum_{k=\lceil n\tau \rceil}^{n} \binom{n}{k} \frac{1}{2^n},$$

$$\tau^* = \arg\min_\tau 2\sum_{k=\lceil n\tau \rceil}^{n} \binom{n}{k} \frac{1}{2^n} < \eta$$

White Box Techniques

- White Box Knowledge
  - Does not access the ground truth watermark or the encoder
  - Has access to the decoder, but does not know the threshold used by the target detectors
- WEvade-W-I
  - Given a watermarked image, add perturbation δ to it such that D outputs a different binary value for each bit of the watermark

$$\min_{\delta} l(D(I_w + \delta), \neg D(I_w)) \tag{4}$$

$$s.t. \ ||\delta||_\infty \leq r, \quad D(I_w + \delta) = \neg D(I_w), \tag{5}$$

- WEvade-W-II
  - find a small perturbation δ such that the decoded watermark $D(I_w + \delta)$ has a bitwise accuracy close to 0.5, compared to a uniformly at random chosen target watermark w_t
  - post-processed watermarked image is indistinguishable with original images with respect to bitwise accuracy

$$\min_{\delta} l(D(I_w + \delta), w_t) \tag{8}$$

$$s.t. \ ||\delta||_\infty \leq r, \quad BA(D(I_w + \delta), w_t) \geq 1 - \epsilon, \tag{9}$$

Solve with Projected Gradient Descent

Algorithm 1 WEvade-W-I and WEvade-W-II

Input: Watermarked image I_w and target watermark w_t
Output: Post-processed watermarked image I_{pw}

$r_b \leftarrow 2$
$r_a \leftarrow 0$
while $r_b - r_a > 0.001$ **do**
  $r \leftarrow (r_a + r_b)/2$
  $\delta' \leftarrow$ FindPerturbation (I_w, w_t, r)
  if ((WEvade-W-I & Equation 5 is satisfied) or (WEvade-W-II & Equation 9 is satisfied)) **then**
    $r_b \leftarrow r$
    $\delta \leftarrow \delta'$
  else
    $r_a \leftarrow r$
  end if
end while
return $I_w + \delta$

Algorithm 2 FindPerturbation (I_w, w_t, r)

Input: Decoder D, objective function l, learning rate α, and maximum number of iterations *max_iter*.
Output: Perturbation δ

$\delta \leftarrow 0$
for $k = 1$ to *max_iter* **do**
  $g \leftarrow \nabla_\delta l(D(I_w + \delta), w_t)$
  $\delta \leftarrow \delta - \alpha \cdot g$
  //Projection to satisfy the perturbation bound
  if $\|\delta\|_\infty > r$ **then**
    $\delta \leftarrow \delta \cdot \frac{r}{\|\delta\|_\infty}$
  end if
  //Early stopping
  if ((WEvade-W-I & Equation 5 is satisfied) or (WEvade-W-II & Equation 9 is satisfied)) **then**
    return δ
  end if
end for
return δ

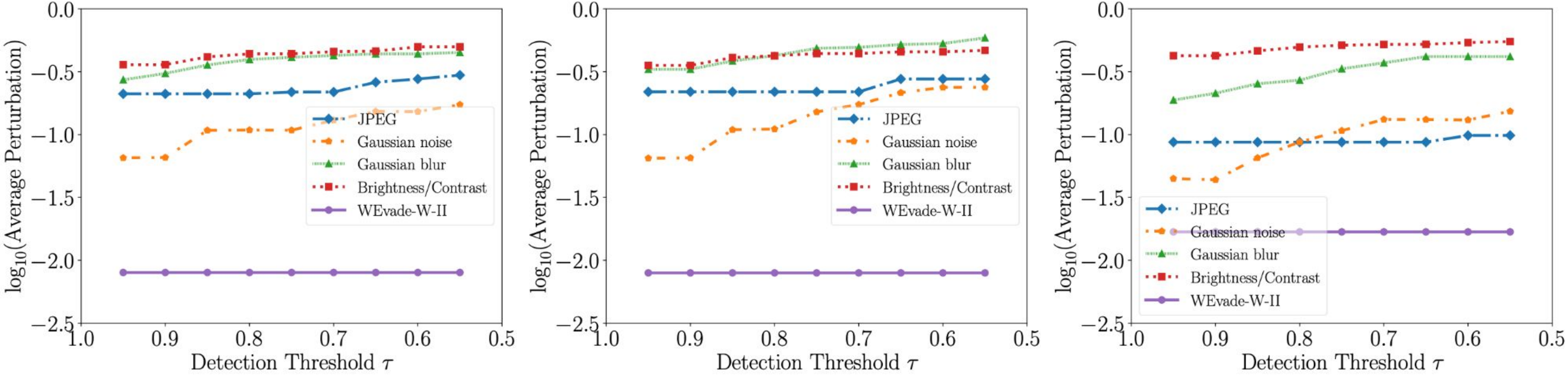
Black Box Techniques

- Black Box Knowledge
  - Does not access the ground truth watermark or the encoder
  - Only has access to the binary result of the detector
- WEvade-B-S
  - Attacker trains a surrogate encoder and decoder
  - Performs white-box attack, WEvade-W-II, on the surrogate decoder
  - Key assumption is the surrogate would output a similar decoded watermark to the target detector
- WEvade-B-Q
  - Directly queries the target detector
  - Extends HopSkipJump
    - Use JPEG compression, lowering quality until it evades, to post-process I_w as the initial I_{pw}
    - If nothing evades, we use the initial I_{pw} found by HopSkipJump
    - Early stop the iteration when the perturbation in I_{pw} increases in multiple consecutive iterations
  - Guarantees evasion at every step

Algorithm 3 WEvade-B-Q

Input: API of the target detector, a watermarked image I_w, query budget max_q, and early stop threshold ES.
Output: Post-processed image I_{pw}

$q \leftarrow 0$
//Initializing I_{pw}
for $Q \in [99, 90, 70, 50, 30, 10, 1]$ **do**
  $q \leftarrow q + 1$
  if API(JPEG(I_w, Q))=="non-AI-generated" **then**
    $I_{pw} \leftarrow$ JPEG(I_w, Q)
    break
  end if
end for
//Iteratively move I_{pw} towards I_w
$\delta_{min} \leftarrow I_{pw} - I_w$
$es \leftarrow 0$
while $q \leq max\_q$ and $es \leq ES$ **do**
  $I_{pw}, q' \leftarrow$ HopSkipJump(I_{pw})
  $q \leftarrow q + q'$
  if $\|I_{pw} - I_w\|_\infty < \|\delta_{min}\|_\infty$ **then**
    $\delta_{min} \leftarrow I_{pw} - I_w$
    $es \leftarrow 0$
  else
    $es \leftarrow es + 1$
  end if
end while
return $I_w + \delta_{min}$



(a) COCO (b) ImageNet (c) CC

Figure 7: Average perturbation added by each post-processing method to evade the double-tail detector with different threshold τ in the white-box setting. We set the parameters of existing post-processing methods such that they achieve the same evasion rate as our WEvade-W-II. The watermarking method is HiDDeN and the results for UDH are shown in Figure 24 in Appendix.

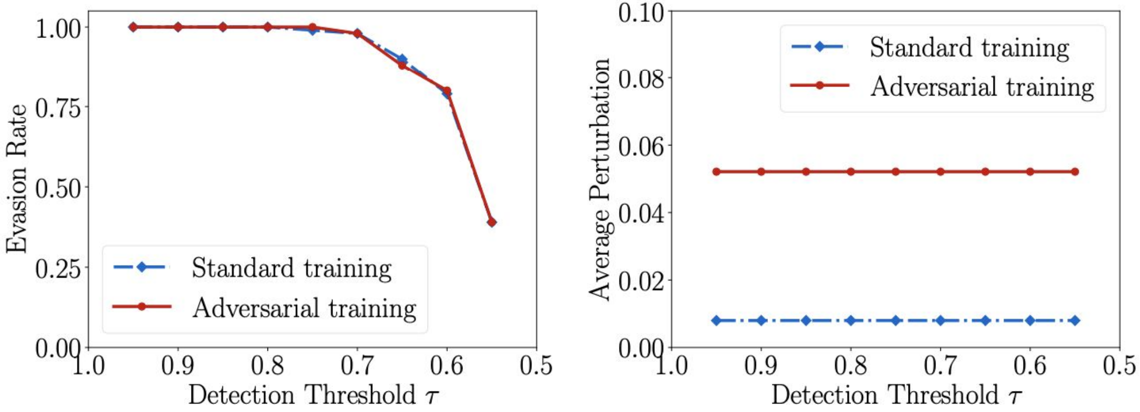


Figure 11: Standard vs. adversarial training for WEvade-W-II

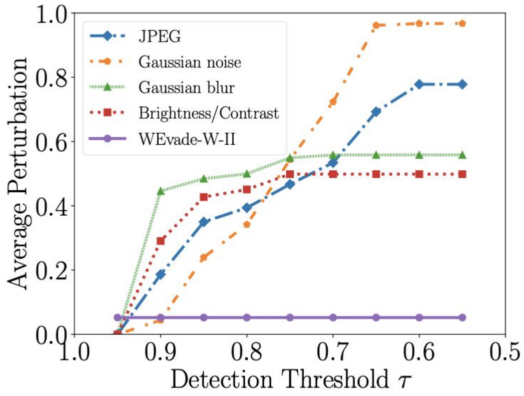


Figure 25: Average perturbation added by each post-processing method to evade the double-tail detector with different threshold τ for the COCO dataset. We set the parameters of existing post-processing methods such that they achieve the same evasion rate as WEvade-W-II. The watermarking method is HiDDeN and adversarial training is used. After adversarial training, the average bitwise accuracy is around 0.87. When τ is 0.95, empirical FNR is 99.6%, and thus existing post-processing methods do not add perturbations to a large fraction of watermarked images based on how we evaluate them, leading to 0 perturbations. However, they need much larger perturbations when τ is smaller than 0.9.

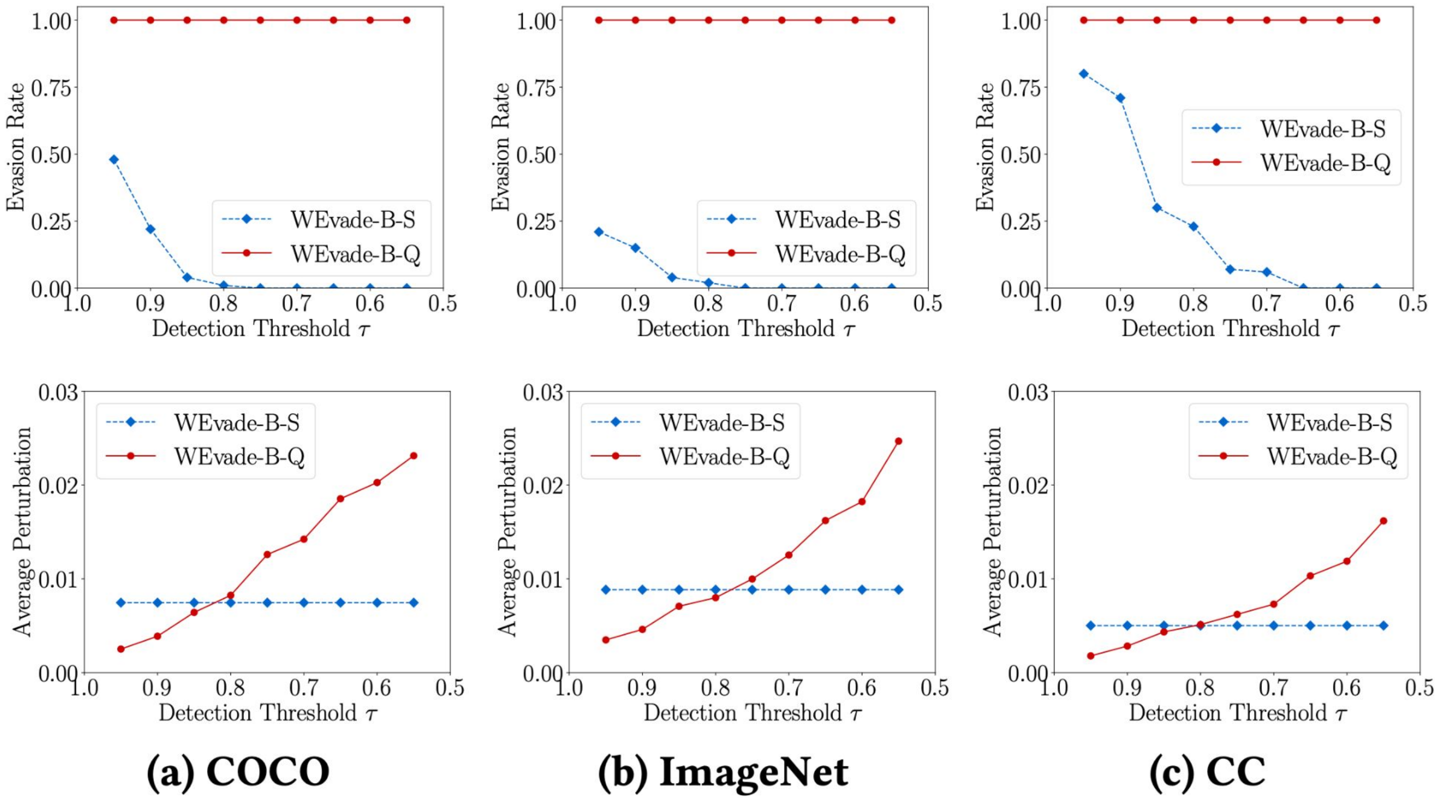
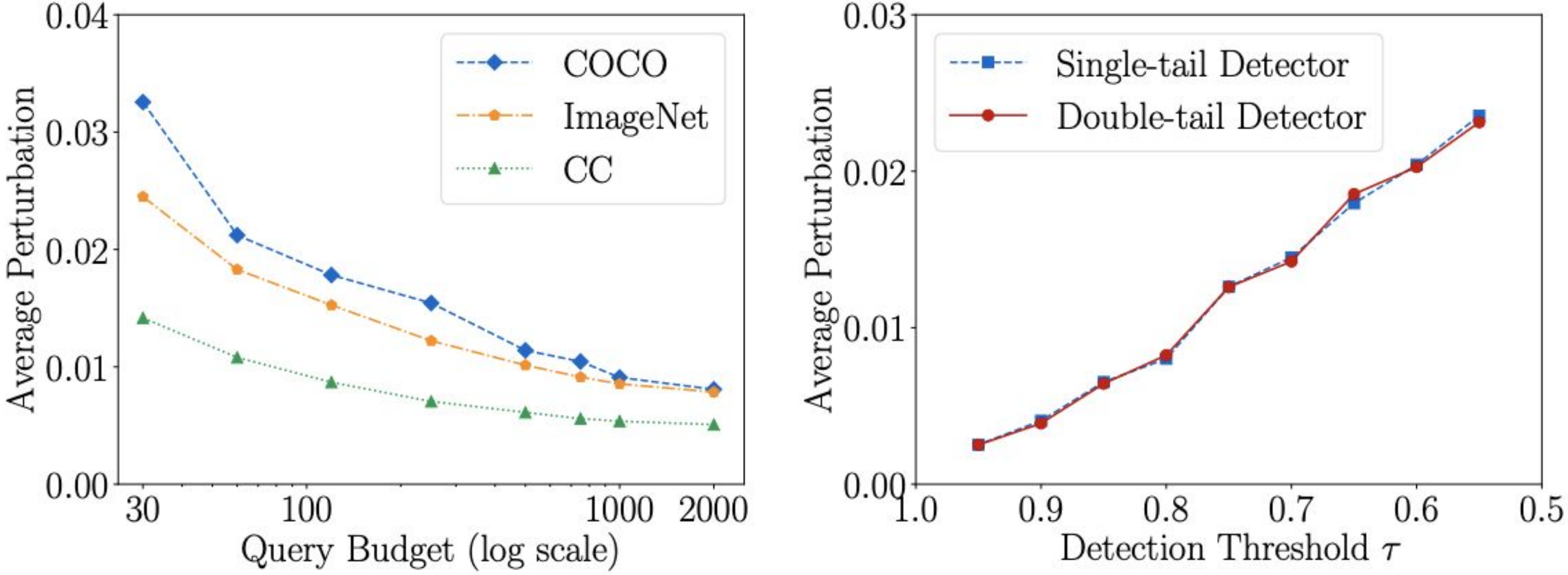


Figure 12: Comparing evasion rates (*first row*) and average perturbations (*second row*) of WEvade-B-S and WEvade-B-Q in the black-box setting. The watermarking method is HiDDeN and Figure 26 in Appendix shows results for UDH.



(a) Impact of query budget max_q **(b) Single-tail vs. double-tail detector**

Figure 13: (a) Average perturbation of WEvade-B-Q as query budget varies. (b) Average perturbation of WEvade-B-Q to evade the single-tail detector or double-tail detector with different threshold τ.

There is Work to Be Done

- Provably robust watermarking methods
  - Produce similar watermarks for the watermarked image and its post-processed version
  - Guarantee a detector with a given threshold will be able to detect a post-processed image whose perturbations are bounded by a given value
- "If the perturbation bound is large enough to be human-perceptible, an attacker has to sacrifice visual quality of the watermarked image in order to evade watermarking-based detector"