A. Proofs for Section 2

Proof of Theorem 2.1. Consider the function $f^{(j)}(t) = f(x + U^{(j)}t)$. This function has gradient $g^{(j)}$ and Hessian $\nabla^2_{jj}f$. By Assumption 5 $\|g^{(j)}\|_0 \leq \|g\|_0 \leq s$ while $\|\nabla^2_{jj}f\|_1 \leq H$ from Assumption 6. Thus, $f^{(j)}(t)$ satisfies the assumptions of Corollary 2.7 of (Cai et al., 2020b) so Theorem 2.1 follows from this result. □

Proof of Theorem 2.2. For notational convenience, for this proof only we let $s := s_{\text{exact}}$. Let $g_{i_1}, \ldots, g_{i_k}, \ldots, g_{i_s}$ denote the non-zero entries of $g := g(x)$, after the permutation has been applied. Let Y_j be the random variable counting the number of g_{i_k} within block j:

$$Y_j = \#\{i_k : \ i_k \text{ in block } j\}.$$

Thus, the random vector $\mathbf{Y} = (Y_1, \ldots, Y_J) \in \mathbb{R}^J$ obeys the multinomial distribution. Observe $\sum_{j=1}^J Y_j = s$ and, because the blocks are equally sized, $\mathbb{E}(Y_j) = \frac{s}{J}$.

By Chernoff's bound, for $\Delta > 0$ and each Y_j individually:

$$\mathbb{P}\left[\left|Y_j - \frac{s}{J}\right| \geq \Delta\frac{s}{J}\right] \leq 2e^{-\frac{\Delta^2\mathbb{E}(Y_j)}{3}} = 2e^{-\frac{\Delta^2 s}{3J}}.$$

Applying the union bound:

$$\begin{aligned}&\mathbb{P}\left[\exists\, j \text{ s.t. } \left|Y_j - \frac{s}{J}\right| \geq \Delta\frac{s}{J}\right]\\ &\leq \sum_{j=1}^J \mathbb{P}\left[\left|Y_j - \frac{s}{J}\right| \geq \Delta\frac{s}{J}\right] \leq 2Je^{-\frac{\Delta^2 s}{3J}},\end{aligned}$$

and so:

$$\begin{aligned}&\mathbb{P}\left[|Y_j - \frac{s}{J}| \leq \Delta\frac{s}{J},\ \forall j \in [1, \cdots, J]\right]\\ &= 1 - \mathbb{P}\left[\exists\, j \text{ s.t. } \left|Y_j - \frac{s}{J}\right| \geq \Delta\frac{s}{J}\right]\\ &\geq 1 - 2Je^{-\frac{\Delta^2 s}{3J}}.\end{aligned}$$

This finishes the proof. □

Proof of Corollary 2.3. From Theorem 2.2 we have that, with probability at least $1 - 2J\exp(\frac{-0.01s_{\text{exact}}}{3J})$, $\|g^{(j)}\|_0 \leq 1.1s_{\text{exact}}/J \leq s$. Assuming this is true, (5) holds with probability $1 - (s/d)^{b_2 s}$. These events are independent, thus the probability that they both occur is:

$$\left(1 - 2J\exp(\frac{-0.01s_{\text{exact}}}{3J})\right)\left(1 - (s/d)^{b_2 s}\right)$$

Because $s \ll d$ the term exponential in s_{exact}/J is significantly larger than $(s/d)^{b_2 s}$. Expanding, and keeping only dominant terms we see that this probability is equal to $1 - \mathcal{O}\left(J\exp(\frac{-0.01s_{\text{exact}}}{3J})\right)$. □

We emphasize that Corollary 2.3 holds for all j. Before proceeding, we remind the reader that for fixed s the *restricted isometry constant* of $Z \in \mathbb{R}^{m\times n}$ is defined as the smallest $\delta > 0$ such that:

$$(1-\delta)\|v\|_2^2 \leq \|Zv\|_2^2 \leq (1+\delta)\|v\|_2^2$$

for all $v \in \mathbb{R}^n$ satisfying $\|v\|_0 \leq s$ .The key ingredient to the proof of Theorem 2.4 is the following result:

Theorem A.1 ((Krahmer et al., 2014, Theorem 1.1))**.** *Let $z \in \mathbb{R}^n$ be a Rademacher random vector and choose a random subset $\Omega = \{j_1, \ldots, j_m\} \subset \{1, \ldots, n\}$ of cardinality $m = c\delta^{-2}s\log^2(s)\log^2(n)$. Let $Z \in \mathbb{R}^{m\times n}$ denote the matrix with rows $\frac{1}{\sqrt{m}}\mathcal{C}_{j_i}(z)$. Then $\delta_s(Z) < \delta$ with probability $1 - n^{-\log(n)\log^2(s)}$.*

c is a universal constant, independent of s, n and δ. Similar results may be found in (Mendelson et al., 2018) and (Huang et al., 2018). In (Krahmer et al., 2014) a more general version of this theorem is provided, which allows the entries of z to be drawn from any sub-Gaussian distribution.

Proof of Theorem 2.4. Let Z be the sensing matrix with rows $\frac{1}{\sqrt{m}}\mathcal{C}_{j_i}(z)$. By appealing to Theorem A.1 with $s = 4s$, $n = d/J$ and $\delta = 0.3843$ we get $\delta_{4s}(Z) \leq 0.3843$, with probability $1 - (d/J)^{\log(d/J)\log^2(4s)}$ From Theorem 2.2 we have that, with probability at least $1 - 2J\exp(\frac{-0.01s_{\text{exact}}}{3J})$, $\|g^{(j)}\|_0 \leq 1.1s_{\text{exact}}/J \leq s$. Assuming $\delta_{4s}(Z) \leq 0.3843$, Theorem 2.4 follows by the same proof as in (Cai et al., 2020b, Corollary 2.7). The events $\|g^{(j)}\|_0 \leq s/J$ and $\delta_{4s}(Z) \leq 0.3843$ are independent, hence they both occur with probability:

$$\begin{aligned}&\left(1 - 2J\exp\left(\frac{-0.01s_{\text{exact}}}{3J}\right)\right)\left(1 - (d/J)^{\log(d/J)\log^2(4s)}\right)\\ &\geq 1 - 2J\exp\left(\frac{-0.01s_{\text{exact}}}{3J}\right) - (d/J)^{\log(d/J)\log^2(4s)}.\end{aligned} \tag{8}$$

Note that the third term in the right side of (8) is *universal*, *i.e.* it holds for all $x \in \mathbb{R}^d$, as it depends only on the choice of Z, not the choice of x. □

B. ZO-BCD for unequally-sized blocks

Using randomly assigned, equally-sized blocks is an important part of the ZO-BCD framework as it allows one to consider a block sparsity $\approx s/J$, instead of the worst case sparsity of s. Nevertheless, there may be situations where it is preferable to use user-defined, unequally-sized blocks. For such cases we recommend the following (we discuss the modifications here for ZO-BCD-R, but with obvious changes it also applies to ZO-BCD-RC). Let $s^{(j)} \leq s$ be an upper estimate of the sparsity of the j-th block gradient: $\|g^{(j)}(x)\|_0 \leq s^{(j)}$. Let

$m^{(j)} = b_1 s^{(j)} \log(D/J)$ (and use the analogous formula for ZO-BCD-RC) and define $m^{\max} = \max_j m^{(j)}$. When initializing ZO-BCD-R, generate $m^{\max}$ Rademacher random variables: $z_1, \ldots, z_{m^{\max}} \in \mathbb{R}^{d^{\max}}$. At each iteration, if block j is selected, randomly select $i_1, \ldots, i_{m^{(j)}}$ from $1, \ldots, m^{\max}$ and for $k = 1, \ldots, m^{(j)}$ let $\tilde{z}_{i_k} \in \mathbb{R}^{d^{(j)}}$ denote the vector formed by taking the first $d^{(j)}$ components of z_{i_k}. Use $\{\tilde{z}_{i_k}\}_{k=1}^{m^{(j)}}$ as the input to Algorithm 1. Note that the $\{\tilde{z}_{i_k}\}_{k=1}^{m^{(j)}}$ possess the same statistical properties as the $\{z_{i_k}\}_{k=1}^{m^{(j)}}$ (*i.e.* they are i.i.d. Rademacher vectors) so using them as sampling directions will result in the same bound on $\|g_k^{(j)} - \hat{g}_k^{(j)}\|_2$ as Corollary 2.3.

C. Proofs for Section 3

Our proof utilizes the main result of (Tappenden et al., 2016). This paper requires $\alpha_k = \frac{1}{L_{j_k}}$, *i.e.* the step size at the k-th iteration is inversely proportional to the block Lipschitz constant. This is certainly ideal, but impractical. In particular, if the blocks are randomly selected it seems implausible that one would have good estimates of the L_j. Of course, since $L_j \leq L_{\max}$ we observe that $L_{\max}$ is *a* Lipschitz constant for every block, and thus we may indeed take $\alpha_k = \alpha = \frac{1}{L_{\max}}$. This results in a slightly slower convergence resulted, reflected in a factor of $L_{\max}$ in Theorem 3.1. Throughout this section we shall assume f satisfies Assumptions 1–6. For all $j = 1, \ldots, J$ define:

$$V_j(x, t) = \langle g^{(j)}(x), t \rangle + \frac{L_{\max}}{2}\|t\|_2^2,$$

so that, by Lipschitz differentiablity:

$$f(x_k + U^{(j)}t) \leq f(x_k) + V_j(x_k, t).$$

Define $t_{k,j}^* := \arg\min V_j(x_k, t) := -\frac{1}{L_{\max}} g_k^{(j)}$ while let $t_{k,j}^{'}$ be the update step taken by ZO-BCD, *i.e.* $t_{k,j}^{'} = \frac{1}{L_{\max}} \hat{g}_k^{(j)}$.

Lemma C.1. *Suppose f satisfies Assumptions 1–3. Then $f(x) - f^* \geq \frac{1}{2L_{\max}}\|g^{(j)}(x)\|_2^2$ for any $x \in \mathbb{R}^d$ and any $j = 1, \ldots, J$.*

Proof. Define $h_x : \mathbb{R}^{d^{(j)}} \to \mathbb{R}$ as $h_x(t) := f(x + U^{(j)}t)$ where $U^{(j)}$ is as described in Section 1. Since $U^{(j)}$ is a linear transformation and f is convex, h_x is also convex. By Assumption 1 and $L_j \leq L_{\max}$, h_x is $L_{\max}$-Lipschitz differentiable. From Assumption 3 it follows $\mathcal{Y}^* = \arg\min_t h_x(t)$ is non-empty, and $h_x^* := \min_t h_x(t) \geq f^*$. Thus, from (Bertsekas, 1997, Proposition B.3, part (c.ii)) , we have:

$$h_x(t) - h_x^* \geq \frac{1}{2L_{\max}}\|\nabla h_x(t)\|_2^2 = \frac{1}{2L_{\max}}\|g^{(j)}(x + U^{(j)}t)\|_2^2$$

for all t. Choose $t = 0$, and use $h_x(0) = f(x)$ and $f^* \leq h_x^*$ to obtain:

$$f(x) - f^* \geq h_x(0) - h_x^* \geq \frac{1}{2L_{\max}}\|g^{(j)}(x)\|_2^2.$$

This finishes the proof. □

Lemma C.2. *Let $\eta = 2\rho^{2n}$ and $\theta = \frac{4\tau^2\sigma H}{L_{\max}}$. For each iteration of ZO-BCD*

$$V_j(x_k, t') - V_j(x_k, t^*) \leq \eta(f(x_k) - f^*) + \theta. \quad (9)$$

with probability $1 - \mathcal{O}\left(J\exp(-\frac{0.01 s_{\text{exact}}}{3J}\right)$ for ZO-BCD-R and with probability greater than

$$1 - 2J\exp\left(\frac{-0.01 s_{\text{exact}}}{3J}\right) - (d/J)^{\log(d/J)\log^2(4.4s/J)}$$

for ZO-BCD-RC.

Proof. For notational convenience, we define $t^* := t_{k,j}^*$, $t^{'} := t_{k,j}^{'}$ and $e_k^{(j)} := \hat{g}_k^{(j)} - g_k^{(j)}$. By definition:

$$t_{k,j}' = -\frac{1}{L_{\max}}\hat{g}_k^{(j)} = -\frac{1}{L_{\max}}(g_k^{(j)} + e_k^{(j)}).$$

Moreover:

$$V_j(x_k, t^*) = -\frac{1}{2L_{\max}}\|g_k^{(j)}\|_2^2$$
$$V_j(x_k, t') = -\frac{1}{2L_{\max}}\|g_k^{(j)}\|_2^2 + \frac{1}{2L_{\max}}\|e_k^{(j)}\|_2^2,$$

and hence:

$$V_j(x_k, t') - V_j(x_k, t^*) \leq \frac{1}{2L_{\max}}\|e_k^{(j)}\|_2^2. \quad (10)$$

Recall $\hat{g}_k^{(j)}$ is the output of GradientEstimation with gradient sparsity level $s_{\text{block}} = 1.1s/J$. Using the error bound (5) and replacing s with $1.1s/J$:

$$\|e_k^{(j)}\|_2^2 \leq \left(\rho^n\|g_k^{(j)}\|_2 + 2\tau\sqrt{\sigma H}\right)^2$$
$$\leq 2\rho^{2n}\|g_k^{(j)}\|_2^2 + 4\tau^2\sigma H. \quad (11)$$

with the stated probabilities, as justified by Corollary 2.3 (for ZO-BCD-R) and Theorem 2.4 (for ZO-BCD-RC) Finally, from Lemma C.1 $\|g_k^{(j)}\|_2^2 \leq 2L_{\max}(f(x_k) - f^*)$ for any $j = 1, \cdots, J$. Connecting this estimate with (10) and (11) we obtain:

$$V_j(x_k, t') - V_j(x_k, t^*) \leq \underbrace{2\rho^{2n}}_{=\eta}(f(x_k) - f^*) + \underbrace{\frac{4\tau^2\sigma H}{L_{\max}}}_{=\theta}.$$

This finishes the proof. □

Proof of Theorem 3.1. Let p_j denote the probability that the j-th block is chosen for updating at the k-th iteration. Because ZO-BCD chooses blocks uniformly at random, $p_j = 1/J$ for all j. If (9) holds for all k then by (Tappenden et al., 2016, Theorem 6.1) if:

$$\eta^2 + \frac{4\theta}{c_1} < 1 \text{ where } c_1 = 2JL_{\max}\mathcal{R}^2(x_0),$$
$$\frac{c_1}{2}(\eta + \sqrt{\eta^2 + \frac{4\theta}{c_1\zeta}}) < \varepsilon < f(x_0) - f^*,$$
$$u := \frac{c_1}{2}\left(\eta + \sqrt{\eta^2 + \frac{4\theta}{c_1}}\right),$$
$$K := \frac{c_1}{\varepsilon - u} + \frac{c_1}{\varepsilon - \eta c_1}\log\left(\frac{\varepsilon - \frac{\theta c_1}{\varepsilon - \eta c_1}}{\varepsilon\zeta - \frac{\theta c_1}{\varepsilon - \eta c_1}}\right) + 2,$$

then $\mathbb{P}(f(x_K) - f^* \leq \varepsilon) \geq 1 - \zeta$. Note that:

- Our η and θ are α and β in their notation.
- In (Tappenden et al., 2016) $c_1 = 2\mathcal{R}^2_{\ell p^{-1}}(x_0)$ where $\mathcal{R}^2_{\ell p^{-1}}(x_0)$is defined as in (2) but using a norm $\|\cdot\|_{\ell p^{-1}}$ instead of $\|\cdot\|_2$. These norms are related as:

$$\|x\|^2_{\ell p^{-1}} = \sum_{j=1}^{J} \frac{L_j}{p_j}\|x^{(j)}\|_2^2 \overset{(a)}{=} \sum_{j=1}^{J} JL_{\max}\|x^{(j)}\|_2^2 = JL_{\max}\|x\|_2^2$$

  where (a) follows as $p_j = 1/J$ for all j and we are taking $L_j = L_{\max}$. Hence, $c_1 = 2JL_{\max}\mathcal{R}^2(x_0)$ as stated.
- $K = \tilde{\mathcal{O}}\left(J/\varepsilon\right)$.

Replace η and θ with the expressions given by Lemma C.2 to obtain the expressions given in the statement of Theorem 3.1. Because (9) holds with high probability at each iteration, by the union bound it holds for all K iterations with probability greater than

$$1 - K\mathcal{O}\left(J\exp\left(-\frac{0.01 s_{\text{exact}}}{3J}\right)\right)$$
$$= 1 - \tilde{\mathcal{O}}\left(\frac{J^2}{\varepsilon}\exp\left(-\frac{0.01 s_{\text{exact}}}{3J}\right)\right)$$

for ZO-BCD-R and with probability greater than

$$1 - 2JK\exp\left(\frac{-0.01 s_{\text{exact}}}{3J}\right) - (d/J)^{\log(d/J)\log^2(4.4s/J)}$$
$$= 1 - \tilde{\mathcal{O}}\left(\frac{J^2}{\varepsilon}\exp\left(-\frac{0.01 s_{\text{exact}}}{3J}\right)\right) - (d/J)^{\log(d/J)\log^2(4.4s/J)} \tag{12}$$

Note the third term in (12) is universal, *i.e.* holds for all iterations, and hence is not multiplied by K. In both cases we use $K = \tilde{\mathcal{O}}(J/\varepsilon)$. As stated, if (9) holds for all k then $\mathbb{P}(f(x_K) - f^* \leq \varepsilon) \geq 1 - \zeta$. Apply the union bound again to obtain the probabilities given in the statement of Theorem 3.1. ZO-BCD-R uses $m = 1.1b_1(s/J)\log(d/J)$ queries per iteration, all made by the GradientEstimation subroutine, and hence makes:

$$mK = (1.1b_1(s/J)\log(d/J))\,K = \tilde{\mathcal{O}}\left(s/\varepsilon\right)$$

queries in total, using $K = \tilde{\mathcal{O}}\left(J/\varepsilon\right)$. On the other hand, ZO-BCD-RC makes $m = 1.1b_3(s/J)\log^2(1.1s/J)\log^2(d/J)$ queries per iteration. A similar calculation reveals that ZO-BCD-RC also makes $mK = \tilde{\mathcal{O}}\left(s/\varepsilon\right)$ queries in total.

The computational cost of each iteration is dominated by solving the sparse recovery problem using CoSaMP. It is known (Needell & Tropp, 2009) that CoSaMP requires $\mathcal{O}(n\mathcal{T})$ FLOPS, where $\mathcal{T}$ is the cost of a matrix-vector multiply by Z. For ZO-BCD-R $Z \in \mathbb{R}^{m\times(d/J)}$ is dense and unstructured hence:

$$\mathcal{T} = \mathcal{O}\left(m\frac{d}{J}\right) = \mathcal{O}\left(\frac{s}{J}\log(d/J)\frac{d}{J}\right) = \tilde{\mathcal{O}}\left(\frac{sd}{J^2}\right).$$

As noted in (Needell & Tropp, 2009), n may be taken to be $\mathcal{O}(1)$ (In all our experiments we take $n \leq 10$). For ZO-BCD-RC, as Z is a partial ciculant matrix, we may exploit a fast matrix-vector multiplication via fast Fourier transform to reduce the complexity to $\mathcal{O}(d/J\log(d/J)) = \tilde{\mathcal{O}}(d/J)$.

Finally, we note that for both variants the memory complexity of ZO-BCD is dominated by the cost of storing Z. Again, as Z is dense and unstructured in ZO-BCD-R there are no tricks that one can exploit here, so the storage cost is $m(d/J) = \tilde{\mathcal{O}}(sd/J^2)$. For ZO-BCD-RC, instead of storing the entire partial circulant matrix Z, one just needs to store the generating vector $z \in \mathbb{R}^{d/J}$ and the index set Ω. Assuming we are allocating 32 bits per integer, this requires:

$$\frac{d}{J} + 32\cdot b_3\frac{s}{J}\log^2\left(\frac{s}{J}\right)\log^2\left(\frac{d}{J}\right) = \mathcal{O}\left(\frac{d}{J}\right).$$

This finishes the proof. □

D. Experimental setup details

In this section, we provide the detailed experimental settings for the numerical results provided in Section 5.

D.1. Settings for synthetic experiments

For both synthetic examples, we use problem dimension $d = 20,000$ and gradient sparsity $s = 200$. Moreover, we use additive Gaussian noise with variance $\sigma = 10^{-3}$ in the oracles. The sampling radius is chosen to be 10^{-2} for all tested algorithms. For ZO-BCD, we use 5 blocks with

uniform step size[8] $\alpha = 0.9$, and the per block sparsity is set to be $s_{\text{block}} = 1.05s/J = 42$. Furthermore, the block gradient estimation step runs at most $n = 10$ iterations of CoSaMP. For the other tested algorithms, we hand tune the parameters for their best performance, and same step sizes are used when applicable. Note that SPSA must use a very small step size ($\alpha = 0.01$) in max-s-squared-sum problem, or it will diverge.

Re-shuffling the blocks. Note that the max-s-squared-sum function does not satisfy the Lipschitz differentiability condition (*i.e.* Assumption 1). Moreover the gradient support changes, making this an extremely difficult zeroth-order problem. To overcome the difficulty of discontinuous gradients, we apply an additional step that re-shuffles the blocks every J iterations. This re-shuffling trick is not required for the problems that satisfies our assumptions; nevertheless, we observe very similar convergence behavior with slightly more queries when the re-shuffling step was applied on the problems that satisfy the aforementioned assumptions.

D.2. Settings for sparse DWT attacks on images

We allows a total query budget of $10,000$ for all tested algorithms in each image attack, i.e. an attack is considered a failure if it cannot change the label within $10,000$ queries. We use a 3-level 'db45' wavelet transform. All the results present in Section 5.2 use the half-point symmetric boundary extension, hence the wavelet domain has a dimension of $676,353$; slightly larger than the original pixel domain dimension. For the interested reader, a discussion and more results about other boundary extensions can be found in Appendix E.

For all variations of ZO-BCD, we choose the block size to be 170 with per block sparsity $s_{\text{block}} = 10$, thus $m = 52$ queries are used per iteration. Sampling radius is set to be 10^{-2}. The block gradient estimation step runs at most $n = 30$ CoSaMP iterations, and step size $\alpha = 10$.

D.3. Settings for sparse CWT attacks on audio signals

For both targeted and untargeted CWT attack, we use Morse wavelets with 111 frequencies, which significantly enlarges the problem dimension from $16,000$ to $1,776,000$ per clip. In the attack, we choose the block size to be 295 with per block sparsity $s_{\text{block}} = 9$, thus $m = 52$ queries are used per iteration. The sampling radius is $\delta = 10^{-3}$. The block gradient estimation step runs at most $n = 30$ CoSaMP iterations. In targeted attacks, the step size is $\alpha = 0.05$, and Table 4 specifies the step size used in untargeted attacks.

[8]We note that using a line search for each block would maximize the advantage of block coordinate descent algorithms such as ZO-BCD, but we did not do so here for fairness

In Table 4, we also include a result of voice domain attack for the comparison. The parameter settings are same with aforementioned untargeted CWT attack settings, but we have to reduce step size to 0.01 for stability. Also, note that the problem domain is much smaller in the original voice domain, so the number of blocks is much less while we keep the same block size.

E. More experimental results and discussion

E.1. Sparse DWT attacks on images

Periodic extension. As mentioned, when we use boundary extensions other than periodic extension, the dimension of the wavelet coefficients will increase, depending on the size of wavelet filters and the level of the wavelet transform. More precisely, wavelets with larger support and/or deeper levels of DWT result in a larger increase in dimensionality. On the other hand, using periodic extension will not increase the dimension of wavelet domain. We provide test results for both boundary extensions in this section for the interested reader.

Compressed attack. As discussed in Section 4, DWT is widely used in data compression, such as JPEG-2000 for image. In reality, the compressed data are often saved in the form of sparse (and larger) DWT coefficients after vanishing the smaller ones. While we have already tested an attack on the larger DWT coefficients only (see Section 5.2), it is also of interest to test an attack after compression. That is, we zero out the smaller wavelet coefficients ($\text{abs} \leq 0.05$) first, and then attack on only the remaining, larger coefficients.

We use the aforementioned parameter settings for these two new attacks. We present the results in Table 6, and for completeness we also include the results already presented in Table 2. One can see that ZO-BCD-R(compressed) has higher attack success rate and lower ℓ_2 distortion, exceeding the prior state-of-the-art as presented in Table 2. We caution that this is not exactly a fair comparison with prior works however, as the compression step modifies the image before the attack begins.

Defense tests. Finally, we also tested some simple mechanisms for defending against our attacks; specifically harmonic denoising. We test DWT with the famous Haar wavelets, DWT with db45 wavelets which is also used for attack, and the essential discrete cosine transform (DCT). The defense mechanism is to apply the transform to the attacked images and then denoise by zeroing out small wavelet coefficients before transforming back to the pixel domain. We only test the defense on images that were successfully attacked. Tables 7 and 8 show the results of defending against the ZO-BCD-R and ZO-BCD-R(large coeff.) attacks respectively. Interestingly, using the attack wavelet (*i.e.* db45) in

Table 6. Results of untargeted image adversarial attack using various algorithms. Attack success rate (ASR), average final ℓ_2 distortion, and number of queries till first successful attack. ZO-BCD-R(periodic ext.) stands for ZO-BCD-R applying periodic extension for implementing the wavelet transform. ZO-BCD-R(compressed) stands for applying ZO-BCD-R to attack only large wavelet coefficients (abs ≥ 0.05) and vanishing the smaller values. The other methods are the same in Table 2.

METHODS	ASR	ℓ_2 DIST	QUERIES
ZO-SCD	78%	57.5	2400
ZO-SGD	78%	37.9	1590
ZO-AdaMM	81%	28.2	1720
ZORO	90%	21.1	2950
ZO-BCD-R	92%	14.1	2131
ZO-BCD-RC	92%	14.2	2090
ZO-BCD-R(periodic ext.)	95%	21.0	1677
ZO-BCD-R(compressed)	**96**%	**13.1**	**1546**
ZO-BCD-R(large coeff.)	**96**%	13.7	1662

Table 7. Defense of image adversarial wavelet attack by ZO-BCD-R. Defense recovery success rate under haar and db45 wavelet filters, and discrete cosine transform (DCT) filter. Thresholding values of 0.05, 0.10, 0.15, and 0.20 were considered.

DEFENCE METHODS	0.05	0.10	0.15	0.20
Haar	74%	75%	76%	75%
db45	74%	72%	71%	63%
DCT	72%	**79**%	75%	67%

defence is not a good strategy. We obtain the best defense results by using a mismatched transform (*i.e.* DCT or Haar defense for a db45 attack) and a small thresholding value.

More adversarial examples. In Figure 5, we presented some adversarial images generated by the ZO-BCD-R and ZO-BCD-R(large coeff.) attacks. For the interested reader, we present more visual examples in Figure 7, and include the adversarial attack results generated by all versions of ZO-BCD.

E.2. Sparse CWT attacks on audio signals

Adversarial attacks on speech recognition is a more nebulous concept than that of adversarial attacks on image classifiers, with researchers considering a wide variety of threat models. In (Cisse et al., 2017), an attack on the speech-to-text engine `DeepSpeech` (Hannun et al., 2014) is successfully conducted, although the proposed algorithm, Houdini, is only able to force minor mis-transcriptions ("A great saint, saint Francis Xavier" becomes "a green thanked saint fredstus savia"). In (Carlini & Wagner, 2018), this problem is revisited, and they are able to achieve 100% success in targeted attacks, with any initial and target transcription

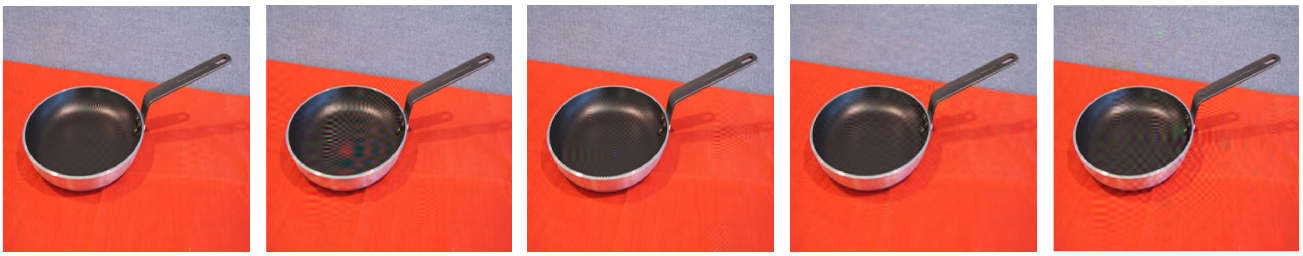
Table 8. Defense of image adversarial wavelet attack by ZO-BCD-R (large coeff.). Defense recovery success rate under Haar and db45 wavelet filters, and discrete cosine transform (DCT) filter. Thresholding values 0.05, 0.10, 0.15, and 0.20 were considered.

DEFENCE METHODS	0.05	0.10	0.15	0.20
Haar	75%	72%	**78**%	76%
db45	71%	70%	69%	63%
DCT	62%	74%	68%	58%

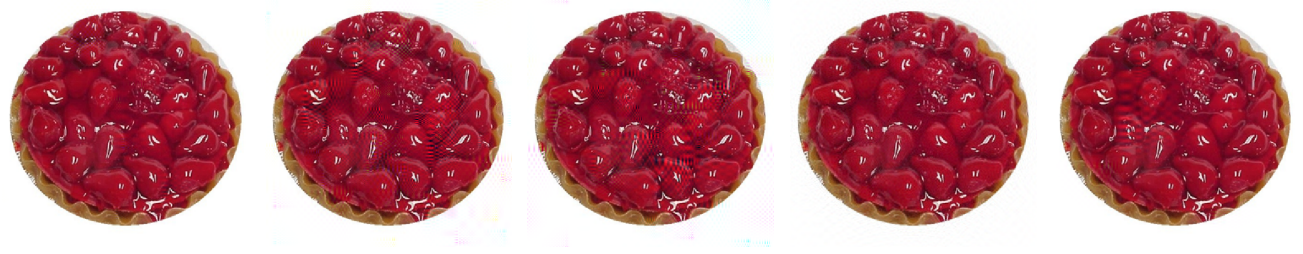
from the Mozilla Common Voices dataset (for example, "it was the best of times, it was the worst of times" becomes "it is a truth universally acknowledged that a single"). We emphasize that both of these attacks are *whitebox*, meaning that they require full knowledge of the victim model. We also note that speech-to-text transcription is not a classification problem, thus the classic Carlini-Wagner loss function so frequently used in generating adversarial examples for image classifiers cannot be straightforwardly applied. The difficulty of designing an appropriate attack loss function is discussed at length in (Carlini & Wagner, 2018).

A line of research more related to the current work is that of attacking keyword classifiers. Here, the victim model is a classifier; designed to take as input a short audio clip and to output probabilities of this clip containing one of a small, predetermined list of keywords ("stop", "left" and so on). Most such works consider the `SpeechCommands` dataset (Warden, 2018). To the best of the author's knowledge, the first paper to consider targeted attacks on a keyword classification model was (Alzantot et al., 2018), and they do so in a black-box setting. They achieve an 87% attack success rate (ASR) using a genetic algorithm, whose query complexity is unclear. They do not report the relative loudness of their attacks; instead they report the results of a human study in which they asked volunteers to listen to and label the attacked audio clips. They report that for 89% of the successfully attacked clips human volunteers still assigned the clips the correct (*i.e.* source) label, indicating that these clips were not corrupted beyond comprehension. Their attacks are per-clip, *i.e.* not universal.

In (Vadillo & Santana, 2019), universal and untargeted attacks on a `SpeechCommands` classifier are considered. Specifically, they seek to construct a distortion δ such that for *any* clip x from a specified source class (*e.g.* "left"), the attacked clip $x + \delta$ is misclassified to a source class (*e.g.* "yes") by the model. They consider several variations on this theme; allowing for multiple source classes. The results we recorded in Section 5.3 (ASR of 70.3% at a remarkably low mean relative loudness of -41.63 dB) were the best reported in the paper, and were for the single-source-class setting. This attack was in the white-box setting.



(a) ZO-BCD-R: "frying pan" → "strainer" (b) ZO-BCD-RC: "frying pan" → "strainer" (c) ZO-BCD-R (periodic ext.): "frying pan" → "strainer" (d) ZO-BCD-R (compressed): "frying pan" → "strainer" (e) ZO-BCD-R (large coeff.): "frying pan" → "strainer"



(f) ZO-BCD-R: "strawberry" → "pomegranate" (g) ZO-BCD-RC: "strawberry" → "pomegranate" (h) ZO-BCD-R (periodic ext.): "strawberry" → "pomegranate" (i) ZO-BCD-R (compressed): "strawberry" → "pomegranate" (j) ZO-BCD-R (large coeff.): "strawberry" → "pomegranate"

Figure 7. More examples of wavelet attacked images by ZO-BCD-R, ZO-BCD-RC, ZO-BCD-R(periodic ext.), ZO-BCD-R(compressed), and ZO-BCD-R(large coeff.), with true labels and mis-classified labels.

Finally, we mention two recent works which consider very interesting, but different threat models. (Li et al., 2020) considers the situation where a malicious attacker wishes to craft a short (say 0.5 second long) that can be added *to any part* of a clean audio clip to force a misclassification. Their attacks are targeted and universal, and conducted in the white-box setting. They do not report the relative loudness of their attacks. In (Xie et al., 2020), a generative model is trained that takes as input a benign audio clip x, and returns an attacked clip $x+\delta$. The primary advantage of this approach is that attacks can be constructed on the fly. In the targeted, per-clip white-box setting they achieve the success rate of 93.6% advertised in Section 5.3, at an approximate relative loudness of -30 dB. They also consider universal attacks, and a transfer attack whereby the generative model is trained on a surrogate classification model.

In all the aforementioned keyword attacks, the victim model is some variant of the model proposed in (Sainath & Parada, 2015). Specifically, the audio input is first transformed into a 2D spectrogram using Mel frequency coefficients, bark coefficients or similar. Then, a 3- to 5- layer convolutional neural network is applied.