A Appendix

A.1 A General Auditor for finite hypothesis classes

Algorithm 4 A General Auditor for finite hypothesis classes

```
$\mathcal{S}_0 := \mathcal{H}$ where $|\mathcal{H}|$ is finite;   stop_flag := False;   $t := 1$; $\epsilon := eps$
while !stop_flag do
    $x_t :=$ picking_next_query($\mathcal{S}_{t-1}$)
    Auditor receives label $y_t$ and explanation $e_t$ from the DS
    $\mathcal{S}_{t+1} :=$ update_search_space($\mathcal{S}_t, x_t, y_t, e_t$)
    Y_a, stop_flag = check_stopping_condition($\mathcal{S}_t$)
end while
return Y_a
```

Algorithm 5 check_stopping_condition()

```
Input: $\mathcal{S}_t$
if $\forall h \in \mathcal{S}_t, s(h) > eps$ then
    decision = Yes, stop_flag = True
else if $\forall h \in \mathcal{S}_t, s(h) \leq eps$ then
    decision = No, stop_flag = True
else
    decision = None, stop_flag = False
end if
return decision, !stop_flag
```

Algorithm 6 picking_next_query()

```
Input: $\mathcal{S}_t$
for $x \in \mathcal{X}$ do
    for $(e, y) \in \mathcal{E} \times \mathcal{Y}$ do
        $\mathcal{S}_{t+1} =$ update_search_space($\mathcal{S}_t, x, y, e$)
    end for
    value$_x$ = $\min_{\mathcal{E} \times \mathcal{Y}} |\mathcal{S}_t| / |\mathcal{S}_{t+1}|$
end for
return argmax$_{x \in \mathcal{X}}$ value$_x$
```

Algorithm 7 update_search_space()

```
Input: $\mathcal{S}, x, y, e$
$\mathcal{S}_{new} := \{h \in \mathcal{S} | h(x) = y$, check_consistent_explanation$(x, e_h(x), e)\}$ {//Explanation method dependent consistency check}
return $\mathcal{S}_{new}$
```

A.2 Auditing Linear Classifiers with Counterfactual Explanations

In this section, we will prove that auditing linear classifiers using counterfactual explanations requires only one query. We denote our auditor by $\texttt{AlgLC}_c$, as outlined in Alg. 1. The proof goes by noting that the counterfactual explanation x' returned by the DS is very close to the projection of input x and that $x - x'$ is parallel to w. We consider the d-th feature to be our feature of interest without loss of generality.

Lemma 3. *Given hyperplane $w^T x + b = 0$, point x and its projection on the hyperplane x'', $x - x'' = \lambda w$ where $\lambda = \frac{w^T x + b}{\|w\|_2^2}$.*

Algorithm 8 $\texttt{AlgLC}_c$: Auditing Linear Classifiers using Counterfactuals

```
Query any point $x$ from the DS
Auditor receives label $y$ and explanation $x'$ from the DS
$\hat{w} \leftarrow x - x'$
if $\hat{w}_i = 0$ ($i^{th}$ feature is the FoI) then
    return No
else
    return Yes
end if
```

Proof. The projection, x'', of x on the hyperplane $w^T x + b = 0$, is found by solving the following optimization problem.

$$\begin{aligned} \min_{x''} \quad & \|x'' - x\|^2 \\ \text{s.t.} \quad & w^T x'' + b = 0 \end{aligned} \tag{1}$$

Let $L(x'', \lambda)$ be the lagrangian for the above optimization problem.

$$\begin{aligned} L(x'', \lambda) &= \|x'' - x\|^2 + 2\lambda(w^T x'' + b) \\ &= \|x''\|^2 + \|x\|^2 - 2x^\top x'' + 2\lambda w^\top x'' + 2\lambda b \\ &= \|x''\|^2 + \|x\|^2 - 2(x - \lambda w)^\top x'' + 2\lambda b \end{aligned} \tag{2}$$

Taking derivative of the lagrangian with respect to x'' and equating with zero we get,

$$\begin{aligned} \frac{\partial L}{\partial x''} = 2x'' - 2(x - \lambda w) &= 0 \\ x - x'' &= \lambda w \end{aligned} \tag{3}$$

By substituting above equation in the constraint for the optimization problem $w^T x'' + b = 0$, we get $\lambda = \frac{w^\top x + b}{\|w\|_2^2}$.

□

Theorem 3.2. *For any $\epsilon \in [0, 1]$, auditor $\texttt{AlgLC}_c$ is an (ϵ,0)-auditor for feature sensitivity and hypothesis class $\mathcal{H}_{LC}$ with $T = 1$ query.*

Proof. Recall that the counterfactual explanation returned by the DS for input x is given as $x' = \text{argmin}_{x':h(x') \neq h(x)} d(x, x')$ where $d(x, x') = \|x - x'\|_2$.

The projection of x on the hyperplane, x'' is the closest point to x on the hyperplane. Therefore $x' = x'' + \Delta$ where Δ is a vector in the direction of w, $\Delta = \gamma w$, γ is a very small non-zero constant.

Therefore, $x - x' = x - (x'' + \Delta) = (x - x'') + \Delta$.
Using lemma 3,

$$\hat{w} := x - x' = \lambda w + \gamma w = c_0 w, \tag{4}$$

where c_0 is a non-zero constant. ($x - x'$ is non-zero due to the definition of counterfactuals, specifically that they have different labels.)

If $w_d = 0$, then it implies that $\hat{w_d} = 0$ and the feature has no effect on the prediction. Thus the score function is zero. Since $\texttt{AlgLC}_c$ returns a $\texttt{No}$ when $\hat{w_d} = 0$, it is always correct in this case. For all the other cases when $w_d \neq 0$, it implies that $\hat{w_d} \neq 0$ and therefore, the feature has an effect on the prediction. Since $\texttt{AlgLC}_c$ returns a $\texttt{Yes}$ when $\hat{w_d} \neq 0$, it is always correct.

Also $\delta = 0$ since our auditor and DS are deterministic. □

Note that this is partial learning since 1) we do not need to learn w exactly and 2) we do not need to learn the bias term b.

A.3 Connection between Model Parameters and Score Function

Notation For vector v, the i^{th} feature is denoted v_i.
Let hypothesis $h_{w,b} \in \mathcal{H}_{LC}$. When w, b are clear from the context, we simply write h. Let w' be the $(d-1)$-dimensional vector $[w_1 \dots w_{d-1}]^T$. Hence w is a concatenation of w' and w_d, denoted by the shorthand $w = [w', w_d]$. We assume that $\|w^{'}\|_2 = 1$.
Let x be a d-dimensional input to this hypothesis. Let x' be the $(d-1)$-dimensional vector $[x_1 \dots x_{d-1}]^T$. Let $\bar{x} = [x_1 \dots x_{d-1}, 1]^\top$. Without loss of generality, let the d^{th} feature be the feature of interest and $x_d \in \{0, 1\}$.
The score function for a hypothesis h is given as, $s(h) = \Pr\left(\left(x^i, x^j\right) \text{ forms a responsive pair}\right)$ where $\left(x^i, x^j\right)$ is sampled uniformly from the set of all pairs and labeled by h. Henceforth we use this score function.

In the following theorem, we bound the fraction of responsive pairs, also our score function, using the weight of our feature of interest, w_d. The score function is bounded by $c \cdot |w_d|$ where c depends on the dimension of the input. This implies that if $|w_d|$ is small, there are not a lot of responsive pairs (low score function value).

Theorem 1. *Assume $\forall x \in \mathcal{X}, \|\bar{x}\|_2 \leq 1$. Let $h_{[w', w_d], b} \in \mathcal{H}_{\mathrm{LC}}$. Then*

$$s(h) \leq \mathrm{C} \cdot |w_d| \tag{5}$$

where $\mathrm{C} = \frac{2^{d-2}}{\pi^{\left(\frac{d-1}{2}\right)} \cdot \Gamma\left(\frac{d+1}{2}\right)}$ is a constant for finite dimension d and Γ is Euler's Gamma function.

Proof. Let P be the set of all pairs of points. Let x^i and x^j denote two inputs forming a pair. Let the pair (x^i, x^j) drawn uniformly from P form a responsive pair.

From the definition of a pair, x^i and x^j only differ in the d^{th} feature. Hence,

$$\begin{aligned} w^\top x^j + b &= w'^\top x'^j + b + w_d x^j_d \\ &= w'^\top x'^i + b + w_d \left(1 - x^i_d\right) \\ &= w'^\top x'^i + b + w_d - w_d x^i_d. \end{aligned} \tag{6}$$

Without loss of generality, let $x^i_d = 0$, hence $x^j_d = 1$.
Therefore,

$$w^\top x^j + b = w'^\top x'^i + b + w_d \tag{7}$$

Next, writing the definition of a responsive pair for $\mathcal{H}_{LC}$ we get,

$$\text{sign}\left(w^\top x^i + b\right) \neq \text{sign}\left(w^\top x^j + b\right) \tag{8}$$

Substituting eq. 7 into the RHS of eq. 8, we get,

$$\text{sign}\left(w^\top x^i + b\right) \neq \text{sign}\left(w'^\top x'^i + b + w_d\right) \tag{9}$$

Expanding the LHS of eq. 9 and substituting $x^i_d = 0$, we get,

$$\text{sign}\left(w'^\top x'^i + b\right) \neq \text{sign}\left(w'^\top x'^i + b + w_d\right) \tag{10}$$

Eq. 10 implies the following,

$$\begin{aligned} 0 \leq w'^{\top} x'^{i} + b \Rightarrow w'^{\top} x'^{i} + b + w_d < 0 \\ 0 > w'^{\top} x'^{i} + b \Rightarrow w'^{\top} x'^{i} + b + w_d \geq 0 \end{aligned} \tag{11}$$

Combining the two equations in eq. 11 we get,

$$\begin{aligned} 0 \leq w'^{\top} x'^{i} + b < -w_d \\ 0 < -\left(w'^{\top} x'^{i} + b\right) \leq w_d \end{aligned} \tag{12}$$

Note that the model (defined by w, b) is fixed. Hence the variables in the above conditions are the inputs x. Note that only one of the conditions in eq. 12 can be satisfied at any time, based on whether $w_d \geq 0$ or $w_d < 0$. The fraction of the inputs which satisfy one of the above conditions correspond to the fraction of responsive pairs and hence is the value of the score function.

Conditions in eq. 12 correspond to intersecting halfspaces formed by parallel hyperplanes. If $\|\bar{x}\|_2 \leq r$, the region of intersection can be upper bounded by a hypercuboid of length $2r$ in $d-2$ dimensions and perpendicular length between the two hyperplanes $l = \frac{|w_d|}{\|w'\|_2}$ in the $(d-1)$-th dimension.

Hence, we can upper bound score function $s(h)$ as,

$$s(h) \leq \frac{(2r)^{d-2} \cdot l}{V_{d-1}(r)} \tag{13}$$

where $l = \frac{|w_d|}{\|w'\|_2}$ and $V_{d-1}(r)$ is the volume of the $(d-1)$-dimensional ball given by $\frac{\pi^{(d-1)/2}}{\Gamma(\frac{d-1}{2}+1)} r^{d-1}$ and Γ is Euler's Gamma function.

Upon simplification we get,

$$s(h) \leq \frac{2^{d-2}}{\pi^{\left(\frac{d-1}{2}\right)}} \frac{l}{r\Gamma\left(\frac{d+1}{2}\right)} \tag{14}$$

Assuming $r = 1$ and $\|w'\|_2 = 1$, we can write eq. 14 as,

$$s(h) \leq \mathrm{C} \cdot |w_d| \tag{15}$$

where $\mathrm{C} = \frac{2^{d-2}}{\pi^{\left(\frac{d-1}{2}\right)} \cdot \Gamma\left(\frac{d+1}{2}\right)}$ is a constant for small dimensions.

□

A.4 Auditing Linear Classifiers with Anchor Explanations

Imperfect Precision. When the precision is not perfect, the anchor augmentation scheme (lines 11-13 in Alg. 2) is not as straightforward. Essentially we cannot assign the same label to all randomly sampled points in the hyperrectangle. To overcome this problem we can use some heuristics and tricks. (a) Note that we wish to exploit the information given by anchors to automatically label samples, but due to imperfect precision there will be errors in this labeling if we use our old augmentation scheme – this is analogous to active learning with noisy labels and we can explore existing literature in this field to deal with this problem. (b) We can internally consider a smaller hyperrectangle than that supplied by the data scientist and only sample from that – this can reduce the error arising from potential wrong labeling (c) We can also ask for labels for some points within each anchor - this might help due to the structure of linear classifiers (only points closer to the decision boundary can have imperfect precision) and the fact that the problem only arises when the precision is somewhere in between 0 and 1, it does not arise at the ends or closer to 0 or 1. Imperfect precision can potentially increase the number of queries or the computations required for anchor augmentation.

Alabdulmohsin et al. (2015) proposed a query synthesis spectral algorithm to learn homogeneous linear classifiers in $O(d \log \frac{1}{\Delta})$ steps where Δ corresponds to a bound on the error between estimated and true classifier. They maintain a version space of consistent hypotheses approximated using the largest ellipsoid $\varepsilon^\star = (\mu^\star, \Sigma^\star)$ where $\mu^\star$ is the center and $\Sigma^\star$ is the covariance matrix of the ellipsoid. They prove that the optimal query which halves the version space is orthogonal to $\mu^\star$ and maximizes the projection in the direction of the eigenvectors of $\Sigma^\star$.

We propose an auditor $\texttt{AlgLC}_a$ as depicted in alg. 2 using their algorithm. The anchor explanations are incorporated through anchor augmentation. But, in the worst-case anchors are not helpful and hence the algorithm reduces essentially to that of Alabdulmohsin et al. (2015) (without anchors). In this section we find the query complexity of this auditor.

Notation In $\texttt{AlgLC}_a$, $\varepsilon^{t\star} = (\mu^{t\star}, \Sigma^{t\star})$ denotes the largest ellipsoid that approximates the version space (corresponds to search space in our case) at time t where $\mu^{t\star}$ is the center and $\Sigma^{t\star}$ is the covariance matrix of the ellipsoid at time t. N^t is the orthonormal basis of the orthogonal complement of $\mu^{t\star}$ and $N^{t'}$ is its transpose. $\alpha^{t\star}$ is the top eigenvector of the matrix $N^{t'}\Sigma^{t\star}N^t$. In the implementation by Alabdulmohsin et al. (2015), some warm-up labeled points are supplied by the user, we denote this set as W. Let $W(t)$ denote the t-th element of this set. Let the d^{th} feature be the feature of interest without loss of generality.

Algorithm 9 $\texttt{AlgLC}_a$: Auditing Linear Classifiers using Anchors

```
 1: Input: $T$, augmentation size $s$, set of warm-up labeled points $W$
 2: set of queried points $Q := \emptyset$, $l :=$size$(W)$
 3: for $t = 1, 2, 3, \ldots, T + l$ do
 4:     if $t <= l$ then
 5:         $(x, y) := W(t)$
 6:         $Q \leftarrow Q \cup (x, y)$
 7:         goto step 15
 8:     else
 9:         Query point $x^t := N^t\alpha^{t\star}$ from the DS
10:         Auditor receives label $y$ and explanation $A_x$ from the DS
11:         $Q \leftarrow Q \cup (x^t, y)$                              }
12:         Sample randomly $q$ points $x^{t1} \ldots x^{tq}$ from $A_x$   }  Anchor Augmentation
13:         $Q \leftarrow Q \cup \{(x^{t1}, y) \ldots (x^{tq}, y)\}$           }
14:     end if
15:     $\varepsilon^{(t+1)*} = \left(\mu^{(t+1)\star}, \Sigma^{(t+1)\star}\right) :=$ estimate_ellipsoid$(Q)$
16:     $N^{t+1} :=$ update_N$(\mu^{(t+1)\star})$
17:     $\alpha^{(t+1)\star} :=$ update_alpha$(N^{t+1}, \Sigma^{(t+1)\star})$
18:     {//Exact formulae for steps 15, 16 and 17 can be found in Alabdulmohsin et al. (2015)}
19: end for
20: $\hat{w} = \mu^{T+1}$
21: if $|\hat{w}_i| \leq \Delta$ ($i^{th}$ feature is the FoI) then
22:     return No
23: else
24:     return Yes
25: end if
```

Next we give a bound on the number of queries required to audit using $\texttt{AlgLC}_a$. With worst-case anchors, it means that we are just using the algorithm of Alabdulmohsin et al. (2015), essentially without explanations and anchor augmentation. The auditor has a fixed ϵ that it decides beforehand. $\texttt{AlgLC}_a$ decides how many times it must run the algorithm of Alabdulmohsin et al. (2015) such that for the fixed ϵ, it satisfies def. 2.

Theorem 3.3. *For every dimension d, there exists $c > 0$ such that for any $\epsilon \in (0, 1)$, auditor $\texttt{AlgLC}_a$ is an $(\epsilon, 0)$-auditor for feature sensitivity and $\mathcal{H}_{LC}$ with $T = O\left(d \log \frac{2c}{\epsilon}\right)$ queries.*

Proof. Let w be the true classifier and $\hat{w}$ be the estimated classifier learnt by $\texttt{AlgLC}_a$.

Let the difference between w and $\hat{w}$ be bounded by Δ as follows,

$$\|w - \hat{w}\|_2 \leq \Delta. \tag{16}$$

The value of Δ will be set later on. Since $\texttt{AlgLC}_a$ uses $|\hat{w}_d|$ to make its decision, the worst case is when the entire error in estimation is on the d^{th} dimension. Hence, we consider $|w_d - \hat{w}_d| \leq \Delta$.

To guarantee that the auditor is an $(\epsilon, 0)$-auditor we need to verify for every hypothesis in the class that if $s(\cdot) = 0$, then the answer is `No` and if $s(\cdot) > \epsilon$, then the answer is `Yes`, see def. 2.

Importantly, $s(w)$ is zero only if $w_d = 0$ from theorem 1. If $w_d = 0 \Rightarrow |\hat{w}_d| \leq \Delta$, by eq. 16. Since $\texttt{AlgLC}_a$ returns a `No` for $|\hat{w}_d| \leq \Delta$, $\texttt{AlgLC}_a$ satisfies def. 2 when $s(w) = 0$ with $\delta = 0$.

Next, we have the case $s(w) > \epsilon$ when auditor should return a `Yes` with high probability. $\texttt{AlgLC}_a$ returns a `Yes` when $|\hat{w}_d| > \Delta$. Hence for $\texttt{AlgLC}_a$ to be correct, we need that $s(w) > \epsilon$ imply that $|\hat{w}_d| > \Delta$.

We can upper bound $s(w)$ using eq. 14 as,

$$s(w) \leq \mathrm{C} \cdot |w_d| \tag{17}$$

Since $\epsilon < s(w)$,

$$\epsilon < \mathrm{C} \cdot |w_d| \tag{18}$$

Since $|w_d - \hat{w}_d| \leq \Delta$,

$$\epsilon \leq \mathrm{C}\left(|\hat{w}_d| + \Delta\right) \tag{19}$$

On rearranging,

$$\frac{\epsilon}{\mathrm{C}} - \Delta \leq |\hat{w}_d| \tag{20}$$

Eq. 20 connects ϵ with Δ and $\hat{w}_d$. For $s(w) > \epsilon$, $|\hat{w}_d|$ should be greater than Δ for $\texttt{AlgLC}_a$ to be correct. Hence, lower bounding the LHS of eq. 20 we get,

$$\begin{aligned} \Delta &\leq \frac{\epsilon}{\mathrm{C}} - \Delta \\ \Delta &\leq \frac{\epsilon}{\mathrm{C}} - \Delta \\ \Delta &\leq \frac{\epsilon}{2\mathrm{C}} \end{aligned} \tag{21}$$

We set $\Delta = \frac{\epsilon}{2\mathrm{C}}$.

From Lemma 1, $\texttt{AlgLC}_a$ reduces to Alabdulmohsin et al. (2015)'s spectral algorithm in the worst-case. This algorithm has a bound of $O(d \log(\frac{1}{\Delta}))$. Hence, the query complexity of $\texttt{AlgLC}_a$ is $O(d \log(\frac{2\mathrm{C}}{\epsilon}))$.

□

A.5 Auditing Decision Trees

Algorithm 10 findpath(x, P)

```
1:  Given : Query $x$ and explanation paths set $P$
2:  for path $p \in P$ do
3:    $not\_p := False$
4:    for $(f_i, v_i, dir_i) \in p$ do
5:      if $(dir_i == \leq \ \& \ x_{f_i} > v_i) || (dir_i == \geq \ \& \ x_{f_i} < v_i)$ then
6:        $not\_p := True$
7:        goto line 3
8:      end if
9:    end for
10:   if $not\_p == False$ then
11:     return True {Query satisfies a pre-existing path}
12:   end if
13: end for
14: return False
```

Algorithm 11 perturb(x, p)

```
Given : Query $x$ and corresponding explanation path $p = [(f, v, dir = \{\leq / \geq\})]$ {Two directions of inequality are used for ease of illustration. The code is generalizable to include other conditions easily.}
Randomly pick a tuple $(f, v, d)$ from $p$
if $dir == \leq$ then
  Perturb $x_f$ such that $x_f > v$
else
  Perturb $x_f$ such that $x_f < v$
end if
$x := x_{\setminus x_f} \cup x_f$
return $x$
```

Theorem A.1. *For any $\epsilon \in [0, 1]$, auditor* `AlgDT` *is an $(\epsilon, 0)$-auditor for feature sensitivity and hypothesis class $\mathcal{H}_{DT}$ with $T = O(V)$ queries where V is the number of nodes in the decision tree.*

Proof. Let L be the number of leaves and V be the number of nodes in the decision tree. The total number of queries asked by 3 equals the number of paths in the tree. The number of paths in a binary decision tree equals the number of leaves L and $L = (V + 1)/2$. Once the auditor makes V queries, it has explored all the paths in the tree and hence knows the tree exactly. Therefore with $O(V)$ queries, it can give the correct auditing decision precisely. □

A.6 Manipulation-Proofness

Yan and Zhang (2022) define manipulation-proofness as follows. Given a set of classifiers V, a classifier h, and a unlabeled dataset S, define the version space induced by S to be $V(h, S) := \{h' \in V : h'(S) = h(S)\}$. An auditing algorithm is ϵ-manipulation-proof if, for any h^*, it outputs a set of queries S and estimate $\hat{s}$ that guarantees that $\max_{h \in \mathcal{V}(h^*, S)} |s(h) - \hat{s}| \leq \epsilon$.

Our anchor and decision tree auditors are manipulation-proof trivially since the version space size at the end of auditing is 1. Our counterfactual auditor is manipulation-proof since only those classifiers which have $w_i = 0$ will be in the version space.

A.7 Experiments

For Adult dataset, the output variable is whether Income exceeds $50K/yr. For Covertype, the output variable is whether forest covertype is category 1 or not. For Credit Default, the output is default payment (0/1).