THE HUMAN-AI SUBSTITUTION GAME: ACTIVE LEARNING FROM A STRATEGIC LABELER

Anonymous authors
Paper under double-blind review

ABSTRACT

The standard active learning setting assumes a willing labeler, who provides labels on informative examples to speed up learning. However, if the labeler wishes to be compensated for as many labels as possible before learning finishes, the labeler may benefit from actually slowing down learning. This incentive arises for instance if the labeler is to be replaced by the ML model, once it is learned. In this paper, we initiate the study of learning from a strategic labeler, who selectively abstains from labeling to slow down learning. We first prove that strategic abstention can prolong learning, and propose novel complexity measures to analyze the query cost of the learning game. Next, we develop a near-optimal deterministic algorithm, prove its robustness to strategic labeling, and contrast it with other active learning algorithms. We also provide extensions that encompass other learning setups/goals. Finally, we characterize the query cost of multi-task active learning, with and without abstention. Our first exploration of strategic labeling aims to add to our theoretical understanding of the imitative nature of ML in human-AI interaction.

1 INTRODUCTION

Over the past few years, the rapid growth of Machine Learning (ML) capabilities has raised the possibility of wide-ranging automation, and consequent worker replacement. Taking a step back from when these ML models are phased in, we ask a basic question on how they first come about.

Where will the training data for these ML models come from?

In many industries, domain-specific knowledge is required to perform the job. Much of this expertise is proprietary (e.g. as trade secrets), and not made publicly available (e.g. on the internet). Thus, for these industries, the answer to our question is paradoxically that: the training data can only come from the workers themselves. Evidently, at this point, we arrive at a clear conflict of interest.

On the one hand, corporations wishes to automate tasks through ML models. On the other hand, the data needed to train these models can only come from the domain experts — the workers in this case, who *know full well* that these models, when trained, will replace them at their jobs. Thus, this line of thinking raises the possibility that we may see domain experts label to actually slow down learning, and in this case, to delay replacement and be compensated for as many labels as possible before then.

Phenomenon: We point out that the conflict of interest described above applies more broadly *whenever* the labeler wishes to maximize payment from labeling. Consider more generally the interaction between a data provider (e.g. a data labeling company) and a learner (e.g. company needing ML models). The more informative the data labeled by the provider, the faster the learner learns, the fewer the examples the learner needs to query the provider, and the lower the provider's total payment. To comment on the generality of this phenomenon, the automation setting is one (of many) instance(s) where this phenomenon arises: the labelers wish to maximize their payment from labeling before the models are fully trained and render their expertises redundant.

In this paper, we study the learning game that arises when the labeler and learner's objective are at odds. The learner wants to learn quickly, but the labeler wants to learning slowly. This departs from the standard assumption that the labeler readily labels any example queried, especially the informative ones. We term this game the **Human-AI Substitution game**, since typically the labeler is human, and the more the model is trained, the less the learner needs the labeler (to label). To study the rate of learning, we turn to learning theory to analyze how the labeler can slow down learning.

1.1 ACTIVE LEARNING WITH A SIMPLE TWIST

We begin our investigation by adopting the standard active learning setup (Hanneke et al., 2014), with the only twist that the labeler aims to maximize the learner's query cost. Since we know of no prior work on this, we focus on perhaps the most fundamental setting: exact learning through membership queries (Angluin, 1988; Hegedűs, 1995). As we will see, this setup is fairly general, and one may use standard reductions to relate the PAC and noisy setting to this setting.

Setup of the Learning Game:

- The learner is interested in learning a hypothesis $h^* \in \mathcal{H} \subset (\mathcal{X} \to \{+1, -1\})$ in hypothesis class $\mathcal{H}$ over a finite pool of unlabeled data $\mathcal{X}$, collected by the learner.
- The labeler knows h^*.
- The learner (adaptively) queries the labeler on unqueried example x.
- The labeler (adaptively) responds using labeling strategy T with response $T(x) \in \{h^*(x), \perp\}$, where $\perp$ denotes abstention.

In this paper, we model the labeler as being able to strategically abstain on queried data, to slow down learning. Being the domain expert with specialized expertise, the labeler is assumed to be able to use this leverage to selectively decide which data points to label. As noted in Section 1, some data points are particularly informative, and naturally the labeler would wish to decline labeling these so that more data would need to be labeled. We also add that this strategy of slowing down the transfer of expertise is not a novel conception. It has been well-documented that in apprenticeships, for instance, teachers (master) strategically slow down the training of their apprentices (Garicano & Rayo, 2017).

The interaction finishes when the termination condition is met, or the learner's querying strategy halts. Based on the learner's desired learning outcome, the termination condition is defined as when $h^* \in \mathcal{H}$ is identified, which we formalize in the following section. If the termination condition is met, the labeler gets a payoff of 1 for every *labeled* data provided. If the termination condition is not met, the labeler gets a payoff of 0. In this game, the learner aims to minimize the total payoff needed to learn h^*, while the labeler aims for the opposite and to maximize the total payoff.

Guaranteeing Learning Outcome: Before proceeding, we note that the labeler *can* always satisfy the learner's objective — by using the non-strategic labeling strategy $T(x) = h^*(x)$ as in the standard active learning setup. Since the labeler can realize the learning outcome, we assume that the learner has this guarantee (of the learning outcome) written into the contract; no payment is awarded otherwise. Indeed, if the labeler cannot guarantee the learning outcome, it seems unlikely that the learner would have chosen to contract the labeler in the first place.

Prolonging Learning through Abstention: The key tension in this interaction is that the labeler has to label in order to be paid, but any labeling results in less data that subsequently need to be labeled. With the labeler only allowed to abstain besides labeling, it is natural to ask: can abstention *significantly* enlarge the query complexity? Our investigation is motivated by the affirmative answer below, where we find that abstention can *exponentially* enlarge query complexity in some settings.

Proposition 1.1 (Abstention induces exponentially higher query complexity)**.** *There exists a hypothesis class $\mathcal{H}$, instance domain $\mathcal{X}$ such that: the query complexity is $O(\log |\mathcal{X}|)$ if the labeler is unable to abstain, and $\Omega(|\mathcal{X}|)$ for any learning algorithm if the labeler is allowed to abstain.*

2 THE MINIMAX LEARNING GAME

2.1 REPRESENTATION OF THE LEARNING GAME STATE

To study this learning game, we first develop an useful, succinct representation of the game state, which is a key contribution of our paper and allow us to formalize the termination condition/protocol. We start by defining the canonical state representation, the version space (VS) (Mitchell, 1982).

Protocol 1 Human-AI Substitution game interaction protocol

Require: Instance domain $\mathcal{X}$, hypothesis class $\mathcal{H}$, queried examples S_X, queried dataset S

```
V ← H, S_X ← ∅, S ← ∅
Nature chooses some h* ∈ H given to the labeler    ▷ throughout, labeler maintains that h* is identifiable: h* ∈ E(V, S_X).
while |E(V, S_X)| ≥ 2 do
    Learner adaptively queries example x ∈ X \ S_X using learning algorithm A
    Labeler adaptively gives label feedback y ∈ {h*(x), ⊥} using labeling oracle T
    Learner updates the VS: V ← V[(x, y)]    ▷ denote V[(x, y)] = {h ∈ V : h(x) = y}
    S_X ← S_X ∪ {x}, S ← S ∪ {(x, y)}
if |E(V, S_X)| = 1 then
    Learner makes total payment to the labeler: Σ_{(x_i,y_i)∈S} 𝟙{y_i ≠ ⊥}
```

$\mathcal{H}$ \ $\mathcal{X}$	x_1	x_2	x_3
h_1	$+1$	-1	$+1$
h_2	-1	-1	$+1$
h_3	$+1$	$+1$	-1
h_4	-1	$+1$	-1
h_5	$+1$	$+1$	$+1$

Table 1: Consider an example hypothesis class $\mathcal{H} = \{h_1, h_2, h_3, h_4, h_5\}$ and instance space $\mathcal{X} = \{x_1, x_2, x_3\}$. The interaction history is $S = \{(x_1, \perp)\}$. Let us use S_X to index just the instances in S, here $S_X = \{x_1\}$. Under S, we have that the VS (Definition 2.1), $V = \mathcal{H}[S] = \{h_1, h_2, h_3, h_4, h_5\}$. We observe that h_1 and h_2 make identical predictions on $\mathcal{X} \setminus S_X = \{x_2, x_3\}$. Likewise, h_3 and h_4 make identical predictions on $\mathcal{X} \setminus S_X$. Therefore, effective version space is actually $E(V, S_X) = \{h_5\}$. If the game reaches this stage, the learner can *already identify* that the target h^* must be h_5.

Definition 2.1. *Given a queried dataset S and a set of classifiers V, define version space $V[S] = \{h \in V : \forall (x, y) \in S \wedge y \neq \perp, h(x) = y\}$ as the subset of classifiers in V consistent with S.*

Some queried examples in S will not have binary labels, due to abstention. And so, we observe that certain hypotheses may be consistent, but *indistinguishable* from other hypotheses, even if all the remaining unqueried data is labeled. This motivates defining a new notion of identifiability of a hypothesis under queried dataset S. Let the set of all queried examples be $S_X = \{x : (x, y) \in S\}$.

Definition 2.2. *Given the set of queried examples and their label responses S, and the queried examples S_X, classifier $h \in \mathcal{H}$ is said to be identifiable with respect to S if:*

- *h is consistent with S, $h \in \mathcal{H}[S]$.*
- *for all other consistent $h' \in \mathcal{H}[S]$: $h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \implies h' = h$, where for brevity we denote $h_1(U) = h_2(U) \iff \forall x \in U \,.\, h_1(x) = h_2(x)$.*

With this, we may develop a new representation of the state of the game, effective version space (E-VS). The E-VS is a refinement of VS, and comprises of only identifiable models given the examples queried. Please see Table 1 for an illustration.

__**Remark:**__ The key insight here is that abstention can in fact *reveal information*. This is despite that abstention is used by the labeler precisely to *prevent releasing* information about h^*. The reason why one can gleam information from labeler's abstention is that hypotheses could be rendered unidentifiable by abstention on a data point, and thus be ruled out without needing any further queries. We operationalize this insight to develop the effective version space representation, which we formalize below.

Definition 2.3. *Given a set of classifiers V and a set of examples S_X, define*

$$E(V, S_X) = \{h \in V : \forall h' \in V \setminus \{h\} : h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)\}$$

as the effective version space with respect to V and S_X.

Definition 2.4. *$h^* \in \mathcal{H}$ is* identified *by queried dataset S if the E-VS, $E(\mathcal{H}[S], S_X) = \{h^*\}$.*

With the interaction termination condition defined, we now formalize the interaction in Protocol 1.

2.2 The minimax learning game

In this paper, we analyze the minimax query complexity — that of the worst-case $h^* \in \mathcal{H}$ to learn under Protocol 1. Towards this, we formulate the minimax learning game, where both the learner queries and the labeler labels *adaptively*, depending on the interaction in previous rounds.

$$\mathrm{CC}(V, S_X) = \begin{cases} -\infty & E(V, S_X) = \emptyset \\ 0 & |E(V, S_X)| = 1 \\ \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \mathcal{Y}} \left(\mathbb{1}(y \neq \perp) + \mathrm{CC}(V[(x, y)], S_X \cup \{x\}) \right) & |E(V, S_X)| \geq 2 \end{cases}$$

Here, the termination states are defined as either $|E(V, S_X)| = 1$ (a hypothesis is identified and the learning outcome is met), or $E(V, S_X) = \emptyset$ (no hypothesis *can* be identified). In the case of non-identifiability, we use a base-case payoff of $-\infty$ to encode that the labeler must ensure identification. As noted in Section 1, the labeler will never end up in such a state, because a positive payoff can always be achieved. There is at least one strategy in $\mathcal{T}_{h^*}$, namely $T = h^*$, that results in a positive payoff. Thus, the labeler's minimax labeling strategy in this game must be identifiable, as only these strategies lead to a positive payoff. We now turn to formalizing what an identifiable strategy is.

Definition 2.5. *Given $h \in \mathcal{H}$, define the set of labeling oracles consistent with h, as:*

$$\mathcal{T}_h = \{T : \mathcal{X} \to \{+1, -1, \perp\} \,|\forall x \in \mathcal{X} \text{ s.t } T(x) \neq \perp, T(x) = h(x)\}.$$

For subset $S_X \subseteq \mathcal{X}$, let $T(S_X) = \{(x, T(x)) : x \in S_X\}$ be the labeled examples provided by labeling oracle T on the examples S_X.

Definition 2.6. *A labeling strategy $T \in \mathcal{T}_h$ is an identifiable oracle if the VS, $\mathcal{H}[T(\mathcal{X})] = \{h\}$.*

In the learning game, the labeler's strategy is some labeling oracle, while the learner's strategy corresponds to some deterministic, querying algorithm: $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{X}$, where $\mathcal{Y} = \{+1, -1, \perp\}$. Define $\mathrm{CC}_{\mathcal{A},T}(V, S_X)$ to be the learning game under querying strategy $\mathcal{A}$ and labeling strategy T. The key result of this subsection is that the game value $\mathrm{CC}(\mathcal{H}, \emptyset)$ can serve as a useful measure of minimax query complexity. $\mathrm{CC}(\mathcal{H}, \emptyset)$ lower bounds the worst-case query complexity of any deterministic learning algorithm in Protocol 1.

Proposition 2.7. *For any deterministic, exact learning algorithm $\mathcal{A}$,*

$$\max_{h \in \mathcal{H}, T \in \mathcal{T}_h} \mathrm{CC}_{\mathcal{A},T}(\mathcal{H}, \emptyset) \geq \mathrm{CC}(\mathcal{H}, \emptyset)$$

This means that for every exact learning algorithm $\mathcal{A}$, there is some worst-case labeling oracle T_h that induces at least $\mathrm{CC}(\mathcal{H}, \emptyset)$ labeled queries by $\mathcal{A}$. Please see Appendix C for all proofs in this section.

3 E-VS Bisection Algorithm Analysis

In this section, we design an efficient algorithm based on E-VS bisection, Algorithm 2, which we prove achieves query complexity $O(\mathrm{CC}(\mathcal{H}, \emptyset) \ln |\mathcal{H}|)$. Proving this guarantee allows us to use the lower bound result, Proposition 2.7, from the previous section to conclude that Algorithm 2's minimax query complexity is optimal up to log factors. Towards analyzing the algorithm performance (and inspired by a related measure in Hanneke (2006) for the non-abstention setting), we introduce a new complexity measure, GIC, that will allow us to bridge Algorithm 2's performance to CC.

Definition 3.1. *Given $\mathcal{H}, \mathcal{X}$, define the global identification cost of version space V, instance set S_X and label cost c:*

$$\mathrm{GIC}(V, S_X) = \min\{t \in \mathbb{N} : \forall T : \mathcal{X} \setminus S_X \to \{+1, -1, \perp\}, \\ \exists \Sigma \subseteq \mathcal{X} \setminus S_X \text{ s.t. } \sum_{x \in \Sigma} c(T(x)) \leq t \wedge |E(V[T(\Sigma)], S_X \cup \Sigma)| \leq 1\}.$$

Intuitively, GIC represents the worst-case sample complexity of a clairovyant querying algorithm that knows ahead of time the labeling oracle that is used by the labeler.

The key lemma behind Algorithm 2 is that there always exists a point that significantly bisects the current E-VS. This justifies greedily querying the point that maximally bisects the E-VS. The lemma below shows this results in an E-VS size reduction of at least a constant $\left(1 - \frac{1}{\mathrm{GIC}(V, S_X)}\right)$ factor.

Algorithm 2 E-VS Bisection Algorithm

Require: Data pool $\mathcal{X}$, hypothesis class $\mathcal{H}$

```
V ← H, S ← ∅                                ▷ VS, queried dataset
while |E(V, S_X)| ≥ 2 and S_X ≠ X do
    Query:                                  ▷ Maximal E-VS bisection point
        x = argmin_{x ∈ X \ S_X} max_{y ∈ {−1,+1}} |E(V, S_X)[(x, y)]|
    Labeler T provides label response: y ∈ {−1, +1, ⊥}
    S ← S ∪ {(x, y)}
    if y ≠ ⊥ then
        V ← V[(x, y)]
return h, the unique element in E(V, S_X)
```

Algorithm 3 Bisection Point Search Sub-routine

Require: Unqueried examples $U = \mathcal{X} \setminus S_X$, abstained examples $S_\perp$, Version Space V, sampling oracle $\mathcal{O}$

```
for sample h ∼ O(V) do
    Construct Z_1 = {(x, ¬h(x)) : x ∈ S^⊥}, Z_2 = {(x, h(x)) : x ∈ X \ S^⊥}
    Run C-ERM to obtain: ĥ ∈ argmin {err(h', Z_1) : h' ∈ H, err(h', Z_2) = 0}
    if ĥ ≠ h then continue
    else                                    ▷ h ∈ E(V, S_X) in this case
        r_x^- ← r_x^- + 1 if h(x) = −1 else r_x^+ ← r_x^+ + 1 for x ∈ U, n ← n + 1
return x* = argmin_{x∈U} |r_x^+/n − r_x^-/n|
```

Lemma 3.2. *For any V, S_X such that* $\mathrm{GIC}(V, S_X)$ *is finite, $\exists x \in \mathcal{X} \setminus S_X$ such that:*

$$\max_{y \in \{-1,+1\}} \left(|E(V[(x,y)], S_X \cup \{x\}))| - 1\right) \leq \left(|E(V, S_X)| - 1\right)\left(1 - \frac{1}{\mathrm{GIC}(V, S_X)}\right).$$

To analyze the algorithm's query complexity, we lower bound $\mathrm{CC}(V, S_X)$ using $\mathrm{GIC}(V, S_X)$.

Lemma 3.3. *For any $V \subset \mathcal{H}$ and $S_X \subset \mathcal{X}$:* $\mathrm{GIC}(V, S_X) \leq \mathrm{CC}(V, S_X)$.

With this, we can prove that Algorithm 2 a) has query complexity $O(\mathrm{CC}(\mathcal{H}, \emptyset) \ln |\mathcal{H}|)$ b) identifies when h^* is identifiable. Please see Appendix D for all the proofs.

Theorem 3.4 (Algorithm 2's query complexity guarantee)**.** *If Algorithm 2 interacts with a labeling oracle T, then it incurs total query cost at most* $\mathrm{GIC}(\mathcal{H}, \emptyset) \ln |\mathcal{H}| + 1$. *Furthermore, if Algorithm 2 interacts with an identifiable oracle T consistent with some $h^* \in \mathcal{H}$, then it identifies h^*.*

3.1 Accessing the E-VS

Algorithm 2 may be viewed as the E-VS variant of the well-known, VS bisection algorithm (Tong & Koller, 2001), an "aggressive" active learning algorithm that greedily queries the point that maximally bisects the VS. The canonical approach for accessing the VS is via sampling, by assuming access to a sampling oracle $\mathcal{O}$. For example, if $\mathcal{H}$ is linear, the VS is a single polytope and one can use a polytope sampler $\mathcal{O}$ to evaluate and search for the point x that maximally bisects the VS.

E-VS Structure: Maximal E-VS bisection point search is less straightforward by contrast. The following structural lemma shows that there exists a linear setting with $\mathcal{X}$ and S such that the E-VS comprises of an *exponential* number of disjoint polytopes. This means that it is computationally infeasible to access the E-VS as polytopes, if one is to use the sampling approach as in VS-bisection.

Proposition 3.5. *There exists an instance space $\mathcal{X} \subset \mathbb{R}^d$ and query response S such that the resultant E-VS comprises of an exponential in d number of disjoint polytopes.*

Towards tractable maximal E-VS bisection point search: To overcome this issue, we develop a novel, oracle-efficient method for accessing the E-VS. We observe that a structural property of the E-VS can be used to check membership given access to a constrained empirical risk minimizaton (C-ERM) oracle (Dasgupta et al., 2007). This allows us to design an oracle-efficient subroutine, Algorithm 3 for any general hypothesis class $\mathcal{H}$, which we prove is sound.

Definition 3.6. *A* constrained-ERM oracle *for hypothesis class $\mathcal{H}$,* C-ERM*, takes as input labeled datasets Z_1 and Z_2, and outputs a classifier: $\hat{h} \in \arg\min_{h' \in \mathcal{H}} \{\mathrm{err}(h', Z_1) : \mathrm{err}(h', Z_2) = 0\}$, where for dataset Z, $\mathrm{err}(h', Z) = \sum_{(x,y) \in Z} \mathbb{1}(h'(x) \neq y)$.*

Proposition 3.7. *Given some $h \in \mathcal{H}$ and access to a* C-ERM *oracle, one can verify $h \in E(V, S_X)$ with one call to the* C-ERM *oracle.*

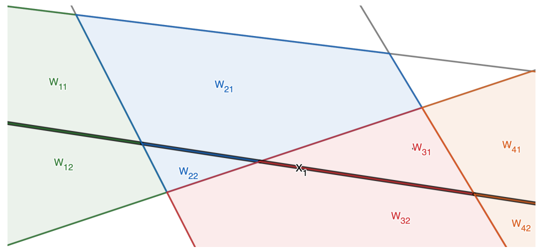

Figure 1: Geometric view of the linear hypothesis class in dual space (as in Tong & Koller (2001)), with examples as hyperplanes and hypotheses as cells, illustrates: (i) Abstention on example x_1 (hyperplane in black) renders hypotheses w_{i1} and w_{i2} (cells of the same color) indistinguishable from each other. In this way, abstentions can carve up the VS (single polytope) into multiple polytopes, as in Proposition 3.5. (ii) In the approximate identifiability game (Subsection 4.1), if x_1 is not in pool X^m, then it induces clusters of merged $\{w_{i1}, w_{i2}\}$ for $i \in [4]$. The goal then is to only identify up to clusters (e.g. the blue cluster of $\{w_{21}, w_{22}\}$), instead of the exact hypothesis (e.g. cell w_{21}).

3.2 Comparing with the VS bisection algorithm

Labeling without identifiability: An advantage of the E-VS algorithm is its robustness to strategic labeling. Theorem 3.4 states that the E-VS algorithm has provable guarantees, *even when* the labeler does not guarantee identification. By contrast, VS-bisection is not robust this way. To concretely compare the two, we construct a learning setup without identification, wherein Algorithm 2 incurs a much smaller number of samples.

Theorem 3.8. *There exists a $\mathcal{H}$ and $\mathcal{X}$ such that the number of labeled examples queried by the E-VS bisection algorithm is $O(\log |\mathcal{X}|)$, while the VS bisection algorithm queries $\Omega(|\mathcal{X}|)$ labels.*

Remark: The key insight is that, by *optimistically* assuming identifiability (even when this is not guaranteed), Algorithm 2 can minimize the number of examples queried. It does so by using the E-VS cardinality to detect when the labeling strategy is non-identifiable and halt the interaction.

3.3 Comparing with EPI-CAL

EPI-CAL (Huang et al., 2016) is a "mellow" active learning algorithm that can handle labeler abstention in a streaming setting, wherein the learner *cannot* control the query order (unlike Algorithm 2), and performs PAC learning (Valiant, 1984). Despite the two differences, we can nevertheless analyze what happens when the labeler can strategically abstain. Our finding is that a strategic labeler can again hold up learning and induce an arbitrarily large query complexity, when the data pool size is not finite and the query order cannot be decided by the learner. This may be evidenced in the simple setting of learning thresholds, where we note that the stream samples are drawn i.i.d, and not adversarially, from a continuous distribution satisfying a standard regularity condition.

Proposition 3.9. *Fix some constant $\epsilon > 0$. Consider a PAC-learning task, where the learner seeks to learn a 1D threshold with at most $\epsilon-$risk with respect to continuous distribution $\mathcal{D}$. For any m i.i.d samples with m sufficiently large and $\mathcal{D}$ probability density bounded away from 0, there is a labeling strategy under which EPI-CAL queries $\Omega(\sqrt{m})$ labeled samples, with probability at least $1/2$.*

Please refer to Appendix E for all proofs in these three subsections.

4 Extensions to Other Learning Settings

The prior sections have assumed that the labeler (e.g. data labeling company) is resourcefully providing non-noisy, labeled data that exactly identifies h^*. In this section, we examine a few ways in which the labeler (e.g. a human worker) may be imperfect in labeling, and extend our guarantees to show how the learner may learn in such settings.

4.1 APPROXIMATE IDENTIFIABILITY

A relaxation of the goal of exact learning is PAC learning: learning some $\hat{h}$ such that inaccuracy $\Pr_{x\sim\mathcal{D}}(\hat{h}(x) \neq h^*(x)) \leq \epsilon$, with probability (w.p.) greater than $1-\delta$ on distribution $\mathcal{D}$ supported on $\mathcal{X}$. This learning goal can arise when the learner wishes to relax the learning outcome/termination criterion, or wishes to weaken the assumption that the labeler knows h^*, to only knowing a fairly accurate hypothesis $h' \in \mathcal{H}$ with $\Pr_{x\sim\mathcal{D}}(h'(x) \neq h^*(x)) \leq \epsilon$.

Reduction: To study the PAC setting, one may use the standard PAC to exact learning reduction (Vapnik, 1999). PAC learning is equivalent to exact learning on a sub-sampled set, $X^m \subseteq \mathcal{X}$, of $m = O(\frac{\mathrm{VC}(\mathcal{H})}{\epsilon}(\ln\frac{1}{\epsilon} + \ln\frac{1}{\delta}))$ i.i.d points from $\mathcal{D}$ ($\mathrm{VC}(\mathcal{H})$ denotes the VC dimension of $\mathcal{H}$).

Then, X^m partitions $\mathcal{H}$ into *clusters* of equivalent hypotheses. Let the projection of $\mathcal{H}$ on X^m be $\mathcal{H}_{|X^m} = \{h(X^m) : h \in \mathcal{H}\}$. For $y \in \mathcal{H}_{|X^m}$, a cluster $C(y)$ of equivalent hypotheses may then be defined as $C(y) = \{h \in \mathcal{H} : h(X^m) = y\}$. The reduction guarantees that, w.p. over $1-\delta$ over the samples X^m, identifying h^*'s cluster $C(h^*(X^m))$ is sufficient for finding a hypothesis $\hat{h}$ such that $\Pr_{x\sim\mathcal{D}}(\hat{h}(x) \neq h^*(x)) \leq \epsilon$.

Approximate Identifiability: Using this reduction, we may analyze the query complexity of approximate identifiability in the resulting learning game. In this game, the learner sets the data pool to be X^m (can be much smaller than $\mathcal{X}$) and aims to only learn *the cluster* h^* belongs to, $C(h^*(X^m))$.

We demonstrate how our E-VS representation can be adapted to apply Algorithm 2 in this approximate identifiability game. We first note that the original E-VS, defined over $\mathcal{H}$ and X^m will no longer suffice as state representation. Consider some $h \in \mathcal{H}$ such that $|C(h(X^m))| \geq 2$ with $\{h', h\} \subseteq C(h(X^m))$. Then, $h(X^m) = h'(X^m) \Rightarrow h'(X^m \setminus \emptyset) = h(X^m \setminus \emptyset)$, which results in the premature elimination of the entire $C(h(X^m))$ cluster at the very start.

To address this, we define a refinement of E-VS, X^m-E-VS. This fix follows from observing that in this game, we should only consider non-identifiability with respect to hypotheses from *other* clusters.

$$E^{X^m}(V, S_X) = \left\{h \in V : \forall h' \in V \setminus \{\bar{h} : \bar{h}(X^m) = h(X^m), \bar{h} \in V\} : h'(X^m \setminus S_X) \neq h(X^m \setminus S_X)\right\}$$

With this, we note that the X^m-E-VS bisection algorithm attains analogous near-optimal guarantees.

Corollary 4.1. *Consider Algorithm 2 instantiated with data pool X^m and state representation X^m-E-VS. When interacting with a labeling oracle T, it incurs total query cost at most* $\mathrm{GIC}(\mathcal{H}, \emptyset)\ln|\mathcal{H}| + 1$. *Furthermore, if the X^m-E-VS bisection algorithm interacts with an identifiable oracle T consistent with some $h^* \in \mathcal{H}$, then it identifies h^*.*

The only remaining consideration is how to efficiently search for the point that maximally bisects clusters in X^m-E-VS. Here, we show that we may adapt the membership check implemented in Algorithm 3 (with the data pool set to X^m) to check hypothesis membership in the coarser X^m-E-VS. That is, we still have an oracle-efficient way of accessing the X^m-E-VS, without needing to explicitly compute and iterate through the clusters.

Proposition 4.2. $h \notin E^{X^m}(V, S_X)$ *iff* $\hat{h}(X^m) \neq h(X^m)$, *where $\hat{h}$ is the minimizer of the C-ERM output on Algorithm 3, Line 3 with $\mathcal{X} = X^m$.*

4.2 NOISED LABELING

In some cases, a labeler can make honest mistakes simply due to human error. We can model this by assuming noised queries (Castro & Nowak, 2008): querying example x returns $h^*(x)$ w.p. $1-\delta(x)$, and $-h^*(x)$ w.p. $\delta(x)$. In this setup, we may use the common approach of repeatedly query a datum to estimate its label w.h.p. (e.g. as in Yan et al. (2016)). This approach thus reduces the noised-label setting to cost-sensitive exact learning, where each x incurs differing cost $c(x)$ dependent on $\delta(x)$. In Appendix D, we prove the generalized version of the results in Section 3 that factors in example-based cost, showing that Algorithm 2 can be applied in this setting with near-optimal guarantees.

4.3 Arbitrary labeling

Thus far, we have assumed a labeler who can (approximately) identify h^*. Here, we touch on when the labeler either does not know h^* (h^*'s cluster), or myopically labels in a way that cannot guarantee the learning outcome. Since the labeler behaves arbitrarily, the learner now cannot be assured of any learning outcome guarantees. In this case, we note that the learner can use the E-VS to preemptively detect when the learning outcome cannot be realized, and halt the futile interaction. While the h^* is unknown, it is possible to detect when *no hypothesis/cluster* is learnable. This is when the E-VS is empty, certifying that the labeler cannot realize the learning outcome. Here, our Theorem 3.4 provides guarantees on the maximum number of times that a non-identifiable oracle will be queried.

Corollary 4.3 (of Theorem 3.4)**.** *Algorithm 2 guarantees bounded query complexity* $GIC(\mathcal{H}, \emptyset) \ln |\mathcal{H}| + 1$ even when the *labeling oracle is non-identifiabble.*

In closing, we note that our algorithm is sound in that if the labeler does turn out to be able to identify h^*, then our algorithm learns h^*. Thus, Algorithm 2 is both sample-efficient with respect to an identifiable labeler, and robust to a non-identifiable one. Please refer to Appendix F for more details on this section.

5 Multi-Task learning from a Strategic Labeler

Multi-task setting: In most jobs, workers in fact perform multiple roles. This motivates the study of multi-task exact learning from a strategic labeler, which we now outline:

1. The learner is now interested in learning multiple $h_i^* \in \mathcal{H}_i$, for tasks $i \in [n]$. The learner can query from instance domain $\mathcal{X} \subseteq \times_{i=1}^n \mathcal{X}_i$, where $\mathcal{X}_i$ is the instance domain for task i.
2. Labeler now provides multi-task labels $y \in \mathcal{Y}^n = \{+1, -1, \perp\}^n$, and for the label cost:
   i) One natural extension of the single task payoff is: $c_{one}(y) = \mathbb{1}(\exists i, y_i \neq \perp)$.
   ii) Another variant of the multi-task labeling payoff is: $c_{all}(y) = \mathbb{1}(\forall i, y_i \neq \perp)$.

We are interested in asking: can the labeler use the multi-task structure to *further* amplify the query complexity? To answer this question, we relate the multi-task query complexity to that of single-task.

Single-task setting:

- **Definition of S_X^i:** given queried data S_X, define the queried data for task i, S_X^i, as: $S_X^i = \mathcal{X}_i \setminus (\mathcal{X} \setminus S_X)_i$, where $(\mathcal{X} \setminus S_X)_i = \left\{x' \in \mathcal{X}_i : \exists x \in \mathcal{X} \setminus S_X, x_i = x'\right\}$.
  In words, S_X^i are examples in $\mathcal{X}_i$, whose label can no longer be obtained. Note that in the multi-task setting, there may exist multiple points that can label some $x_i \in \mathcal{X}_i$. So abstention on one of those points does *not* necessarily mean that x_i cannot be labeled.
  Example: $\mathcal{X} = \{x_{11}, x_{12}\} \times \{x_{21}, x_{22}\}$. $S_X = \left\{[x_{11}, x_{21}], [x_{12}, x_{22}]\right\}$, then $S_X^i = \{\}$ for $i = 1, 2$. This is because it is still possible for the labeler to give labels on all points, i.e. x_{11}, x_{22} through $[x_{11}, x_{22}]$ and x_{12}, x_{21} through $[x_{12}, x_{21}]$.
- **Definition of V_i:** given the current multi-task version space V, we can naturally define the single-task version space for task i as: $(V)_i = V_i = \{h_i : h \in V\}$

5.1 Upper Bound

To understand if multi-task structure can inflate query complexity, we upper bound the multi-task complexity in terms of the sum of the single-task complexities. Proving an upper bound would imply that the labeler cannot increase the query complexity through the multi-task structure. We find that upper bounds only arise under certain regularity assumptions. Thus, we first provide complementary negative results without these assumptions, showing settings where the labeler *can* amplify the multi-task query complexity. Please note that all proofs in this section may be found in Appendix G, where we also prove results in the non-abstention setting that may be of independent interest.

Proposition 5.1. *Under both label costs, there exists a non-Cartesian product version space $V \subseteq \mathcal{H}$ and query response $S \subseteq (\mathcal{X} \times \mathcal{Y})^*$ such that $CC(V_i, S_X^i) \geq 0$ for all i, and: $CC(V, S_X) \geq \sum_{i=1}^n CC(V_i, S_X^i) + n - 1$.*

Remark: Below, we find that the choice of label cost matters in multi-task learning. If the (more generous) c_{one} is used as label cost, the labeler can leverage this to increase the query complexity.

Proposition 5.2. *If the label cost is $c_{one}(y) = \mathbb{1}(\exists i, y_i \neq \perp)$, there exists V and S such that $CC(V_i, S_X^i) = 1$, but $CC(V, S_X) = |\mathcal{X}|$. This implies that: $CC(V, S_X) > \sum_{i=1}^n CC(V_i, S_X^i)$.*

Through the two negative examples, we have that: in order for the labeler to be unable to amplify the multi-task query complexity, two necessary regularity conditions are a) the version space is a cartesian product b) the payoff cost is c_{all} (and cannot be c_{one}). In the result below, we prove the two conditions are sufficient, providing a full characterization when the upper bound can be achieved.

Theorem 5.3. *For all $V = \times_{i\in[n]} V_i$ and $S_X \subseteq \mathcal{X}$, under labeling cost $c_{all}(y) = \mathbb{1}(\forall i, y_i \neq \perp)$: $CC(V, S_X) \leq \sum_{i=1}^n CC(V_i, S_X^i)$.*

For the remainder of the section, we will prove results under the (more generous) label cost, c_{one}.

5.2 Lower Bound

Through lower bounds, we illustrate that the multi-task version space structure can in fact speed up learning as well. The intuition is that the structure in V may make it so that the multi-task E-VS shrinks faster due to unidentifiability. The following negative example evidences this.

Proposition 5.4. *There exists a non-Cartesian product version space V and query response S such that $CC(V_i, S_X^i) \geq 0$ for all i, but: $CC(V, S_X) < \max_{i\in[n]} CC(V_i, S_X^i)$.*

Proposition 5.5. *There exists a Cartesian product version space V and query response S with $CC(V, S_X) < 0$ such that: $CC(V, S_X) < \max_{i\in[n]} CC(V_i, S_X^i)$.*

Thus, identifiability ($CC(V, S_X) \geq 0$), and Cartesian product are needed to prove a lower bound.

Theorem 5.6. *For all $V = \times_{i\in[n]} V_i$ and $S_X \subseteq \mathcal{X}$, if $CC(V, S_X) \geq 0$, then: $CC(V, S_X) \geq \max_{i\in[n]} CC(V_i, S_X^i)$.*

6 Related Works

The theory of Active Learning (Hanneke, 2009) (AL) has a rich history and began with the study of realizable learning (Angluin, 1988; Hegedűs, 1995; Freund et al., 1997; Dasgupta, 2004; Dasgupta et al., 2005). To the best of our knowledge, we are the first to consider a labeler whose objective is the *opposite* of the learner: the labeler wants to maximize, and not minimize, the query complexity. Our work also initiates the study of this setup by focusing on the fundamental setting of realizable learning. In face of such a strategic labeler, we develop an active learning algorithm with near-optimal query complexity guarantees.

Abstaining Labeler: The closest two papers to our work are Yan et al. (2016); Huang et al. (2016), who also study learning from a labeler that can abstain. In Yan et al. (2016), the labeler can abstain or noise, where the rate of an incorrect label/abstention is fixed apriori. Our work differs from that of Yan et al. (2016; 2015) in that the labeler can adaptively label (e.g. abstain) based on the full interaction history so far, thus allowing for more complex, sequential labeling strategies. In Huang et al. (2016), the labeler abstains when uniformed, and after a number of abstentions in a region, learns to label the region (an "epiphany"). Our setting differs in that the labeler does know the labels for all regions, but instead strategically abstains to enlarge query complexity.

Other related AL works: Our technical results are inspired by the minimax results on exact learning in Hanneke (2006). The noisy setup we consider is similar to that of e.g. Castro & Nowak (2008). Our algorithm belongs the class of "aggressive" learning algorithms (Dasgupta, 2004; Golovin & Krause, 2010), which has been of interest for their sample-efficiency. As in (Sabato et al., 2013), we also study label-dependent cost. Please refer to Appendix I for further discussion on related works.

7 Discussion

In this paper, we provide the first set of theoretical evidence that labelers can slow down learning, making even active learning algorithms sample-inefficient. With this, we explore and characterize the resultant minimax learning game, in the single and multi-task setting. This theoretical investigation was motivated by the broader observation that a labeler's objective may be at odds with the learner's, which applies for instance in the setting where workers slow down model training to delay replacement and to maximize labeling payment before being replaced.

Limitations/Future Work: Our work takes a first step into understanding what labelers can do to slow down learning. We hope that our results can pave the way for analyzing more complicated learning settings. One such setting is agnostic learning (Balcan et al., 2006; Dasgupta et al., 2007).

Societal/Broader Impact: Zooming further out, workers have this incentive to slow down training — if they lack financial security after being replaced. ML offers tremendous potential in bettering our lives, automating away jobs people do not want to do. However, it can also automate away jobs that people *do want to do*. It is our hope that this paper adds to the important discussion on whether we should always automate, once we have the ability to automate, as well as the discussion on fair labeler compensation during the automation process (De Vynck, 2023).

REFERENCES

Dana Angluin. Queries and concept learning. *Machine learning*, 2:319–342, 1988.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 65–72, 2006.

James N Brown and Robert W Rosenthal. Testing the minimax hypothesis: A re-examination of o'neill's game experiment. *Econometrica: Journal of the Econometric Society*, pp. 1065–1081, 1990.

Rui M Castro and Robert D Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353, 2008.

Yiling Chen, Chara Podimata, Ariel D Procaccia, and Nisarg Shah. Strategyproof linear regression in high dimensions. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 9–26, 2018.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 17, 2004.

Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *International conference on computational learning theory*, pp. 249–263. Springer, 2005.

Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. *Advances in neural information processing systems*, 20, 2007.

Gerrit De Vynck. Ai learned from their work. now they want compensation. *Washington Post*, 2023. URL `https://www.washingtonpost.com/technology/2023/07/16/ai-programs-training-lawsuits-fair-use/`.

Ofer Dekel, Felix Fischer, and Ariel D Procaccia. Incentive compatible regression learning. *Journal of Computer and System Sciences*, 76(8):759–777, 2010.

Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28:133–168, 1997.

Drew Fudenberg and Luis Rayo. Training and effort dynamics in apprenticeship. *American Economic Review*, 109(11):3780–3812, 2019.

Luis Garicano and Luis Rayo. Relational knowledge transfers. *American Economic Review*, 107(9): 2695–2730, 2017.

Daniel Golovin and Andreas Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. In *COLT*, pp. 333–345, 2010.

Steve Hanneke. The cost complexity of interactive learning. *Unpublished manuscript*, 2006.

Steve Hanneke. *Theoretical foundations of active learning*. Carnegie Mellon University, 2009.

Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pp. 111–122, 2016.

Tibor Hegedűs. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the eighth annual conference on Computational learning theory*, pp. 108–117, 1995.

Tzu-Kuo Huang, Lihong Li, Ara Vartanian, Saleema Amershi, and Jerry Zhu. Active learning with oracle epiphany. *Advances in neural information processing systems*, 29, 2016.

Tom M Mitchell. Generalization as search. *Artificial intelligence*, 18(2):203–226, 1982.

Javier Perote and Juan Perote-Pena. Strategy-proof estimators for simple regression. *Mathematical Social Sciences*, 47(2):153–176, 2004.

Nikita Puchkin and Nikita Zhivotovskiy. Exponential savings in agnostic active learning through abstention. In *Conference on Learning Theory*, pp. 3806–3832. PMLR, 2021.

Sivan Sabato, Anand D Sarwate, and Nati Srebro. Auditing: Active learning with outcome-dependent query costs. *Advances in Neural Information Processing Systems*, 26, 2013.

Yufei Tao, Hao Wu, and Shiyuan Deng. Cross-space active learning on graph convolutional networks. In *International Conference on Machine Learning*, pp. 21133–21145. PMLR, 2022.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. Active learning from noisy and abstention feedback. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1352–1357. IEEE, 2015.

Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. Active learning from imperfect labelers. *Advances in Neural Information Processing Systems*, 29, 2016.

Yinglun Zhu and Robert Nowak. Efficient active learning with abstention. *arXiv preprint arXiv:2204.00043*, 2022.

A EXPERIMENTS

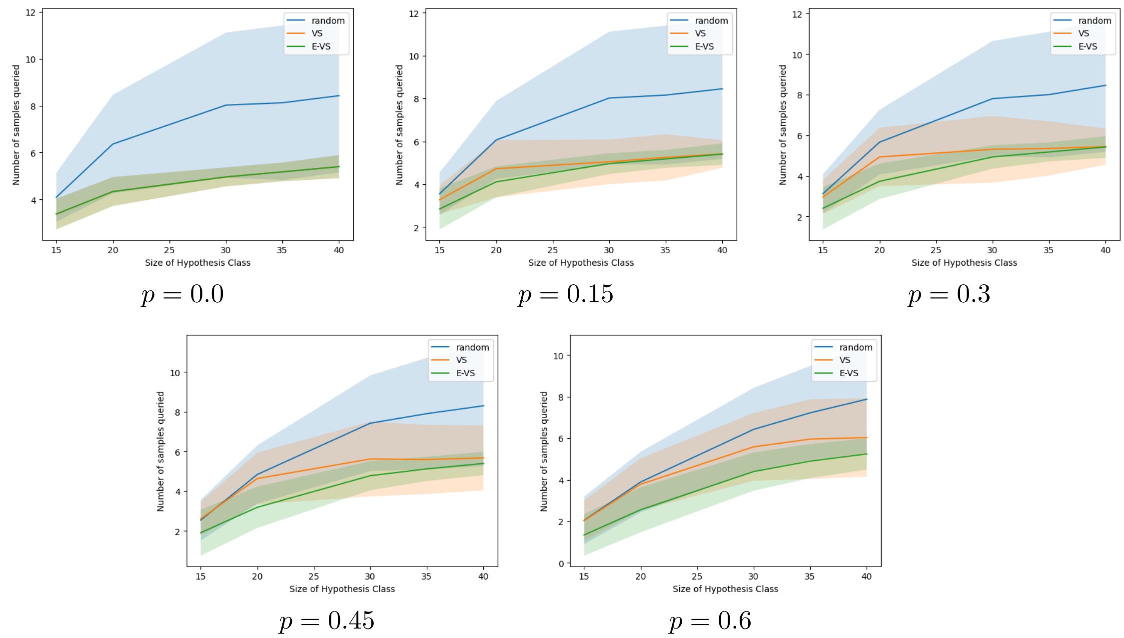

Figure 2: Plots the average number of examples queried by each algorithm across 50 randomly generated instances, along with its standard deviation (shaded region). For this set of plots, the labeling oracle is random (and may not ensure identifiability), with varying amount of abstention p. In the plots, the lower the average, the better the algorithm (needing fewer samples).

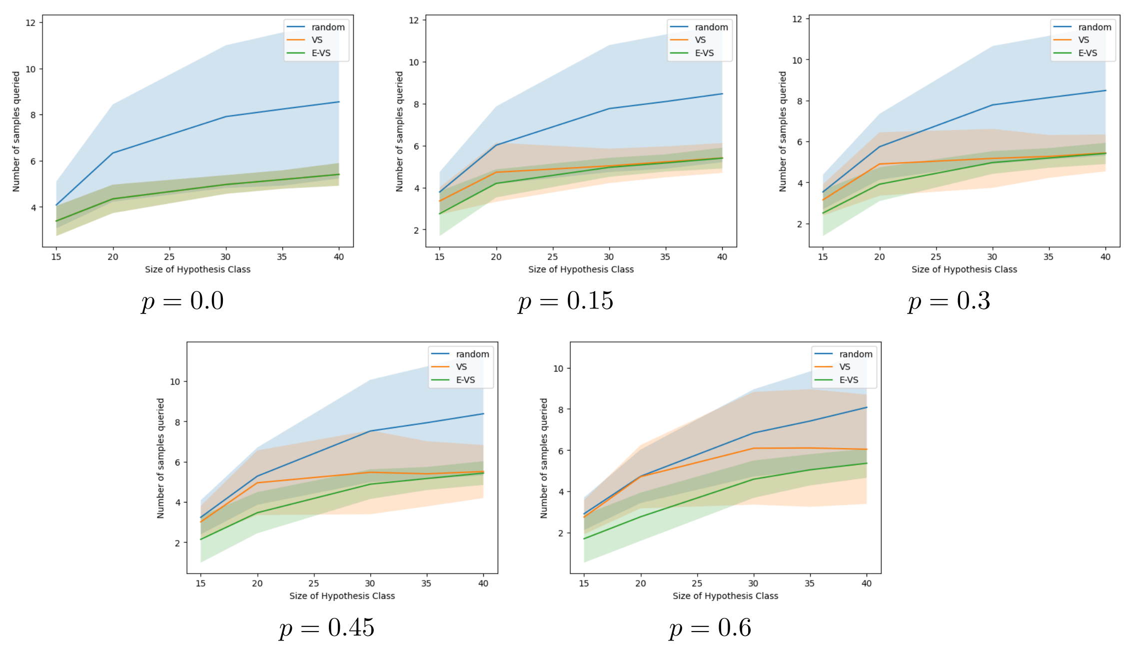

Figure 3: Plots the average number of examples queried by each algorithm across 50 randomly generated instances, along with its standard deviation (shaded region). For this set of plots, the labeling oracle is identifiable, with varying amount of abstention p. In the plots, the lower the average, the better the algorithm (needing fewer samples).

To supplement our theoretical minimax analysis in the main section, we examine the performance of three learning algorithms, E-VS bisection, VS-bisection and randomly query (a point), in "average-case" settings by randomly generating learning instances.

Experiment Setup: We consider five sizes for the hypothesis class ranging from 15 to 40. Given a particular hypothesis class size $|\mathcal{H}|$, we generate 50 random learning instances by randomly generating the binary labels of hypotheses on examples $x \in \mathcal{X}$, where the number of data points $|\mathcal{X}|$ is varied from 5 to 30. Given a learning instance, we consider setting (the underlying hypothesis) h^* to be every $h \in \mathcal{H}$, and thus average the query complexity across random instances as well as across $\mathcal{H}$. This is done to explore the average-case query complexity, where we do not focus on the query complexity of one particular $h^* = h \in \mathcal{H}$ (as was done in some of the worst-case analyses).

We investigate two possible labeling strategies, with varying amounts of abstention $p = 0.0, 0.15, 0.3, 0.45, 0.6$. The first strategy is that given the underlying hypothesis $h^* \in \mathcal{H}$, it abstains on labeling a point x with probability p, and outputs $h^*(x)$ otherwise (w.p. $1 - p$). This labeling strategy may be viewed as one that abstains arbitrarily, and may compromise identifiability. This models the labeling strategy of a myopic labeler. The second strategy is a more careful, adaptive labeling strategy that always ensures identifiability. Given the underlying h^*, when x is queried, it computes the resultant E-VS if x was abstained upon. If abstention leads to non-identifiability, it labels x and returns $h^*(x)$. Otherwise, it abstains with probability p and provides the label otherwise. This may be viewed as a more shrewed labeling strategy that always ensures identifiability, while using some abstention.

Results: We have a few observations. First, as a sanity check, we observe that in the absence of abstention ($p = 0.0$), the E-VS and VS algorithm behave exactly the same and thus their performance should match, which they do as in the first plot of both Figure 2 and Figure 3.

Next, we observe the general trend that the E-VS algorithm attains the lowest query complexity, followed by the VS algorithm and then the random querying algorithm. Moreover, the gap becomes more pronounced with the amount of abstention. This makes sense because the E-VS representation is designed to handle abstention, while the VS is not. This trend thus illustrates the effectiveness of using the E-VS representation in face of an abstaining labeler.

Finally, we see that the gap is most significant in face of a non-identifying labeler (as in plots of Figure 2). This is because the E-VS algorithm can do early detection of non-identifiability and aptly halt the interaction, while the VS bisection and random querying algorithm cannot detect non-identifiability due to the use of the VS representation. We proved that the query complexity can be significantly larger in a worst-case setup in Theorem 3.8. And here, we see that in addition to the worst-case setting (as in Theorem 3.8), the E-VS also fares better in the average-case. Thus, this again affirms the robustness of the E-VS algorithm in face of a non-identifying labeler.

Notation	
S	$S = \{(x_1, y_1), (x_2, y_2), ...\}$, query responses in the interaction history
S_X	$S_X = \{x : (x, y) \in S\}$, indexes the queried examples in S
$S^\perp$	$S^\perp = \{x : (x, y) \in S, y = \perp\}$, queried examples that were given abstention
$V_x^y, V[(x, y)]$	$V_x^y, V[(x, y)] = \{h \in V : h(x) = y\}$, updated VS (used interchangeably)
$E(V, S_X)$	$E(V, S_X) = \{h \in V : \forall h' \in V \setminus \{h\} : h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)\}$, effective VS
$S_{\mathcal{A},T}$	Interaction history between $\mathcal{A}$ and T
S_X^i	$S_X^i = \mathcal{X}_i \setminus (\mathcal{X} \setminus S_X)_i$, where $(\mathcal{X} \setminus S_X)_i = \{x' \in \mathcal{X}_i : \exists x \in \mathcal{X} \setminus S_X, x_i = x'\}$
$(V)_i$	$(V)_i = V_i = \{h_i : h \in V\}$
$c_{one}(y)$	$c_{one}(y) = \mathbb{1}(\exists i, y_i \neq \perp)$
$c_{all}(y)$	$c_{all}(y) = \mathbb{1}(\forall i, y_i \neq \perp)$

Table 2: Table of commonly used notation.

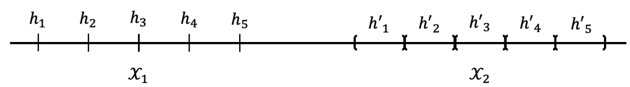

Figure 4: The setup behind Proposition 1.1 is that of learning an one-to-one threshold-interval hypothesis class $\mathcal{H} = \{(h_i, h_i')\}_{i \in [n]}$. The learner seeks to identify $(h_{i^*}, h_{i^*}^{'})$. The labeler can abstain on $\mathcal{X}_1$, and prevent the learner from learning through this sample-efficient part of the instance space. This forces the learner to learn the interval $h_i^{'*}$ (instead of threshold h_i^*) through $\mathcal{X}_2$, and incur much larger sample complexity.

B Proofs for Section 1

B.1 Technical Results

Proposition B.1. *There exists a hypothesis class $\mathcal{H}$, instance domain $\mathcal{X}$ such that the exact learning sample complexity is $O(\log |\mathcal{X}|)$ if the labeler is unable to abstain, and $\Omega(|\mathcal{X}|)$ for any learning algorithm if the labeler is allowed to abstain.*

Proof. Let the $h_i : [0, 1] \to \{+1, -1\}$ for $i \in [n]$ denote intervals of length $1/n$ centered at $(2i - 1)/2n$ for $i \in [n]$, and $h_i' : (1, 2] \to \{+1, -1\}$ for $i \in [n]$ denote thresholds at $1 + i/n$ for $i \in [n]$. Define hybrid-hypothesis class $\mathcal{H}$ of threshold-intervals, $\mathcal{H} = \{f_1, ..., f_n\}$, where:

$$f_i(x) = \begin{cases} h_i(x) & x \in [0, 1] \\ h_i'(x) & x \in (1, 2] \end{cases}$$

Let $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$, where $\mathcal{X}_1 = \{\frac{1}{2n}, ..., \frac{2n-1}{2n}\}$ and $\mathcal{X}_2 = \{1 + \frac{3}{2n}, ..., 1 + \frac{2n-1}{2n}\}$.

1) When the labeler is not allowed to abstain, the learner may binary search on $\mathcal{X}_2$ to identify h_{i^*}', which identifies f_{i^*}. The required sample complexity is $O(\log n)$.

2) When the labeler is allowed to abstain, consider the following labeling strategy T:

i) $T(x) = \perp$ for all $x \in \mathcal{X}_2$

ii) $T(x) = h_{i^*}(x)$ for all $x \in \mathcal{X}_1$.

Note T is a labeling strategy that allows for identification. $\mathcal{H}[T(\mathcal{X})] = \mathcal{H}[T(\mathcal{X}_1)] = \{f_{i^*}\}$.

Interacting with T is equivalent to learning one of n disjoint intervals, which requires $\Omega(n)$ samples under any learning algorithm Dasgupta (2004). And so, T induces $\Omega(n)$ samples, which in turn lower bounds the sample complexity induced by the minimax labeling strategy. ☐

Remark B.2. *We note that one may be generalize the above result to any cross-space learning setting (Tao et al., 2022) with significant differences in query complexity among the instance spaces.*

Protocol 4 Minimax strategic slow learning game

Require: Instance domain $\mathcal{X}$, hypothesis class $\mathcal{H}$

```
S ← ∅, V ← H
▷ Throughout, the labeler needs to maintain that there is at least one classifier consistent with all labels so far and is identifiable
while |E(V, S_X)| ≥ 2 do
    Learner queries example x ∈ X \ S_X
    Labeler provides label feedback y ∈ {−1, +1, ⊥}
    Learner incurs cost c(y), and updates its version space V ← V_x^y
    S ← S ∪ {(x, y)}
Nature sets h* to be the only model in E(V, S_X) if |E(V, S_X)| = 1    ▷ Nature sides with the labeler, sets h* to be the remaining model at the end
```

The labeler's optimal strategy here is simple: label only through the instance space that leads to the highest query complexity, and abstain on all other (more informative) instance spaces.

Remark B.3. *We also add that the labeling strategy need not be identifiable for this result to hold. One can simply define T to still abstain on all of $\mathcal{X}_2$ and output -1 on all of $\mathcal{X}_1$, which still induces $\Omega(|\mathcal{X}|)$ query complexity.*

C Proofs for Section 2

C.1 The Minimax Learning Game

The game strategy for the labeler and learner now corresponds to a labeling oracle, and a querying algorithm.

Labeling Oracle Notation: Given $h \in \mathcal{H}$, define the set of labeling oracles consistent with h as,

$$\mathcal{T}_h = \{T : \mathcal{X} \to \{+1, -1, \perp\} | \forall x \in \mathcal{X} \text{ s.t } T_h(x) \neq \perp, T(x) = h(x)\}$$

Given subset $S_X \subseteq \mathcal{X}$, let us define $T(S_X)$ to be the set of labeled examples induced by oracle T on the examples S_X.

Suppose $V \subseteq \mathcal{H}$, let us define:

$$V[T(S_X)] = \left\{h \in V | h(x) = T(x), \forall x \in S_X \wedge T(x) \neq \perp \right\}$$

A labeling strategy $T \in \mathcal{T}_h$ is an identifiable oracle if $\mathcal{H}[T(\mathcal{X})] = \{h\}$.

Querying Algorithm Notation: Formally, a learning algorithm consists of the following:

- Query function $f_{query} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{X}$
- Termination function $f_{term} : (\mathcal{X} \times \mathcal{Y})^* \to \{\text{TRUE}, \text{FALSE}\}$
- Output function $f_{out} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{H}$

$\mathcal{A}$ interacts with the labeler by:

```
S ← ∅
while f_term(S) = FALSE do
    Query x ← f_query(S)
    Receive label y
    S ← S ∪ {(x, y)}
return f_out(S)
```

Properties of f_{term}:

- If $\mathcal{A}$ is an exact learning algorithm, $f_{term}(S) = \text{TRUE}$ if $|E(V, S_X)| \leq 1$.

- If $\mathcal{A}$ has a fixed budget N, f_{term} outputs TRUE when S is such that: $|\{(x,y) \in S : y \neq \perp\}| = N$

$CC_{\mathcal{A},T}(V, S_X)$ **Learning Game:** Denote $CC_{\mathcal{A},T}(V, S_X)$ as the learning game under querying strategy $\mathcal{A}$, labeling strategy T. Formally, let point $x_{\mathcal{A},S}$ be queried by $\mathcal{A}$ after seeing interaction history S (corresponding to some sequentially labeled dataset) induced by labeling oracle T. With this, the value function of the learning game with strategies $\mathcal{A}$ and T may be recursively defined as follows:

$$CC_{\mathcal{A},T}(V, S_X) = \begin{cases} -\infty & E(V, S_X) = \emptyset \\ 0, & |E(V, S_X)| = 1 \\ \mathbb{1}(T(x_{\mathcal{A},S}) \neq \perp) + CC(V[(x_{\mathcal{A},S}, T(x_{\mathcal{A},S}))], S_X \cup \{x_{\mathcal{A},S}\}) & |E(V, S_X)| \geq 2, \end{cases}$$

C.2 Technical Results

Lemma C.1. *Let the deterministic query algorithm $\mathcal{A}$ interact with labeling oracle $T \in \mathcal{T}_{h_0}$ for M queries, generating the following interaction history: $S_M = (x_1, T(x_1)), (x_2, T(x_2)), ..., (x_M, T(x_M))$. Suppose, there exists a classifier h_1 and $T' \in \mathcal{T}_{h_1}$ such that for all $x \in \{x_1, ..., x_M\}$, $T(x_i) = T'(x_i)$. Then, $\mathcal{A}$ generates the same interaction history, when interacting with T' for M queries.*

Proof. As defined previously, algorithm $\mathcal{A}$ comprises of query function f_{query}, termination function f_{term} and output function f_{out}. We show by induction that for steps $i = 0, 1, ..., M$, the interaction histories of $\mathcal{A}$ with T and T' agree on their first i elements for $i \leq M$.

Base Case: For step $i = 0$, both interaction histories are empty and thus agree.

Induction Step: Suppose the statement holds up until step i for some $i < M$. That is, when $\mathcal{A}$ interacts with T and T' generates the same set of queried examples:

$$S_i = \{(x_1, y_1), ..., (x_i, y_i)\}$$

Consider step $i + 1$. Firstly, $\mathcal{A}$ continues to make a query and does not terminate, since $f_{term}(S_i) =$ FALSE for $i < M$.

Now, for the $i+1$th query, $\mathcal{A}$ applies function f_{query} and queries $x_{i+1} = f_{query}(S_i)$. Since $T'(x_j) = T(x_j)$ for all j and in particular for $j = i + 1$, we have that $(x_{i+1}, T'(x_{i+1})) = (x_{i+1}, T(x_{i+1}))$. And so, with this and the induction hypothesis, we have that $\mathcal{A}$ when interacting with T' and T generates the same set of queried examples:

$$S_{i+1} = \{(x_1, y_1), ..., (x_{i+1}, y_{i+1})\}$$

up to step $i + 1$.

Using this, we can conclude that the interaction histories after M steps of $\mathcal{A}$ with T' and T are identical. □

Remark C.2. *Suppose, after the Mth step, we have that TRUE $= f_{term}(S_{\mathcal{A},T}) = f_{term}(S_M)$. And so, we have that $S_M = S_{\mathcal{A},T'}$, and the interaction of $\mathcal{A}$ with T' also terminates at the Mth step.*

Thus, for model output, we have $S_{\mathcal{A},T} = S_M = S_{\mathcal{A},T'} \Rightarrow f_{out}(S_{\mathcal{A},T}) = f_{out}(S_{\mathcal{A},T'})$.

Proposition C.3. *Let N denote the labeling budget. Let $S_N^{\mathcal{A},T}$ be the interaction history of a deterministic algorithm $\mathcal{A}$ with oracle T up until the Nth label is given, or at termination (without using all of the budget). Let $(S_X)_N^{\mathcal{A},T}$ be the examples queried during the interaction. For any deterministic algorithm $\mathcal{A}$, if $N < CC(\mathcal{H}, \emptyset)$, there exists some $h \in \mathcal{H}$ and identifiable oracle $T \in T_h$ such that $|E(\mathcal{H}[S_N^{\mathcal{A},T}], (S_X)_N^{\mathcal{A},T})| \geq 2$.*

Proof. Fix a deterministic algorithm $\mathcal{A}$. We will show the following. If $\mathcal{A}$ has already obtained an ordered sequence of queried examples S, and has a remaining label budget $N \leq CC(\mathcal{H}[S], S_X) - 1$, then there exists $h \in \mathcal{H}[S]$ and T_h such that, $\mathcal{A}$, when interacting with T_h:

1. obtains a sequence of queried examples S in the first $|S|$ rounds

2. when the interaction terminates, the E-VS has cardinality at least two: $|E(\mathcal{H}[S_N^{\mathcal{A},T_h}],(S_X)_N^{\mathcal{A},T_h})| \geq 2$.

The theorem follow from the second point of this claim by taking $S = \emptyset$.

We now turn to proving the above claim by induction on $\mathcal{A}$'s remaining label budget N.

Base Case: If $N = 0$, then $CC(\mathcal{H}[S], S_X) \geq 1$. By Lemma D.6, we know that $|E(\mathcal{H}[S], S_X)| \geq 2$.

Construction of T_h:

Let $h \in E(\mathcal{H}[S], S_X)$.

Define T_h to be such that for $(x_i, y_i) \in S$, $T_h(x_i) = y_i = h(x_i)$ (the latter equality holds by definition of h) if $y_i \neq \perp$ and $T_h(x_i) = \perp$ if $y_i = \perp$.

Define $T_h(x) = h(x)$ for all $x \in \mathcal{X} \setminus S_X$.

Since $h \in E(\mathcal{H}[S], S_X)$, we know that $h(\mathcal{X} \setminus S_\perp) \neq h'(\mathcal{X} \setminus S_\perp), \forall h' \neq h \in V$. And so, $\mathcal{H}[T(\mathcal{X})] = \mathcal{H}[T(\mathcal{X} \setminus S_\perp)] = \{h\}$, which implies that T is an identifiable oracle for h.

By construction and using Lemma C.1, T_h's interaction with $\mathcal{A}$ results in S, satisfying the first point. Moreover, since $N = 0$, $S_0^{\mathcal{A},T_h} = S$. And so, $|E(\mathcal{H}[S_0^{\mathcal{A},T_h}],(S_X)_0^{\mathcal{A},T_h})| = |E(\mathcal{H}[S], S_X)| \geq 2$

Induction Step: Suppose the claim holds for all $N \leq n$ for some $0 < n < CC(\mathcal{H}, \emptyset) - 1$.

Now, suppose during the interaction, algorithm $\mathcal{A}$ has remaining budget $N = n + 1$, and the obtained queried examples history S is such that $CC(\mathcal{H}[S], S_X) \geq N + 1 = n + 2$.

Our goal is to show the existence of h and T_h that satisfy the two listed properties under these two assumptions.

Define x'_j for index $j \geq 1$ to be the next example $\mathcal{A}$ queries such that a binary label y'_j is given (i.e $y'_j \neq \perp$), as we recursively unroll the CC expression, via the querying procedure below.

$L \leftarrow S, L_X \leftarrow S_X, j \leftarrow 1$
repeat
  Query $x'_k \leftarrow f(L)$ using $\mathcal{A}$
  Labeler return $y'_k = \arg\max_{y \in \{-1,+1,\perp\}} \Big(\mathbb{1}(y \neq \perp) + CC(\mathcal{H}[L \cup \{(x'_k, y)\}], L_X \cup \{x'_k\}\Big)$
  $L \leftarrow L \cup \{(x'_k, y'_k)\}$
  $L_X \leftarrow L_X \cup \{x'_k\}$
until $y_j \neq \perp$ or $f_{term}(L) = \text{TRUE}$

There are two cases:

- If j exists (i.e. the final j satisfies $y_j \neq \perp$), then after querying $\{(x'_i, y'_i)\}_{1:j}$, the learner has remaining budget of $N - 1 = n$.

  Next, we see that with each abstention, the CC value is non-decreasing, as justified in the first three steps:

  We have that:

$$
\begin{aligned}
CC(\mathcal{H}[S], S_X) &\leq \max_{y_1 \in \{+1,-1,\perp\}} \mathbb{1}(y_1 \neq \perp) + CC(\mathcal{H}[S \cup \{(x'_1, y_1)\}], S_X \cup \{x'_1\}) \\
&= \mathbb{1}(y'_1 \neq \perp) + CC(\mathcal{H}[S \cup \{(x'_1, y'_1)\}], S_X \cup \{x'_1\}) \\
&= CC(\mathcal{H}[S \cup \{(x'_1, y'_1)\}], S_X \cup \{x'_1\}) \\
&\leq \ldots \qquad \text{(unroll from } j-1 \text{ to 1, using } \mathbb{1}(y'_i \neq \perp) = 0 \text{ for } i < j \text{ and } \diamond) \\
&\leq \mathbb{1}(y'_j \neq \perp) + CC(\mathcal{H}[S \cup \{(x'_i, y'_i)\}_{1:j}], S_X \cup \{x'_i\}_{1:j}) \\
&= 1 + CC(\mathcal{H}[S \cup \{(x'_i, y'_i)\}_{1:j}], S_X \cup \{x'_i\}_{1:j})
\end{aligned}
$$

($\diamond$) : We may use the non-decreasingness property to unroll, because from non-decreasingness, for all $l \leq j$, $CC(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:l}], S_X \cup \left\{x'_i\right\}_{1:l}) \geq n + 2 \geq 2$. Therefore, $\left|E(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:l}], S_X \cup \left\{x'_i\right\}_{1:l})\right| \geq 2$, and we have that:

$$CC(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:l}], S_X \cup \left\{x'_i\right\}_{1:l}) = \\ \min_x \max_y \mathbb{1}(y \neq \perp) + CC(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:l} \cup \left\{(x, y)\right\}], S_X \cup \left\{x'_i\right\}_{1:l} \cup \{x\})$$

From this, we get that:

$$n \leq CC(\mathcal{H}[S], S_X) - 2 \leq (CC(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:j}], S_X \cup \left\{x'_i\right\}_{1:j}) + 1) - 2$$

By induction hypothesis, there exists $h \in \mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:j}]$ and T_h, such that when $\mathcal{A}$ interacts with T_h (after obtaining query history $S \cup \left\{(x'_i, y'_i)\right\}_{1:j}$) and with label budget n, the final version space is of cardinality at least two:

$$|E(\mathcal{H}[S_N^{\mathcal{A},T_h}], (S_X)_N^{\mathcal{A},T_h})| \geq 2$$

In addition, when interacting with T_h, $\mathcal{A}$ obtains history $S \cup \left\{(x'_i, y'_i)\right\}_{i=1}^{j}$ in its first $|S| + j$ rounds of interaction, which implies that it obtains example sequence S in its first $|S|$ rounds of interaction with T_h. This proves the first property also holds and completes the induction.

- Now, we consider the case when j does not exist. This means that the other exit condition must hold: $f_{term}(L) = \text{TRUE}$. And so, $\mathcal{A}$ terminates with all abstentions: $y'_i = \perp$ for $i \in [j]$.

  As above, we iteratively use the non-decreasingness of CC with abstention $y'_i = \perp$ to get that:
  $$n + 2 \leq CC(\mathcal{H}[S], S_X) \leq ... \leq CC(\mathcal{H}[L], L_X)$$
  for the final state $\mathcal{H}[L], L_X$.

  From this, we have that $|E(\mathcal{H}[L], L_X)| \geq 2$.

  Pick some $h \in E(\mathcal{H}[L], L_X)$. As in the prior T_h construction, define T_h so that: $T_h(x) = y$ for all $(x, y) \in L$, and $T_h(x) = h(x)$ for all $x \in \mathcal{X} \setminus L_X$.

  By construction and Lemma C.1, T_h's interaction with $\mathcal{A}$ induces L.

  Since $f_{term}(L) = \text{TRUE}$, $S_N^{\mathcal{A},T} = L$. And so, $|E(\mathcal{H}[S_N^{\mathcal{A},T_h}], (S_X)_N^{\mathcal{A},T_h})| = |E(\mathcal{H}[L], L_X)| \geq 2$, satisfying the second condition.

  Finally, since $\mathcal{A}$'s interaction with T_h generates L, the first $|S|$ steps also matches S. This satisfies the first property.

□

Proposition C.4. *For any deterministic, exact learning algorithm $\mathcal{A}$,*

$$\max_{h \in \mathcal{H}, T \in \mathcal{T}_h} CC_{\mathcal{A},T}(\mathcal{H}, \emptyset) \geq CC(\mathcal{H}, \emptyset)$$

Proof. From Prop. C.3, we know that for $N = CC(\mathcal{H}, \emptyset) - 1$, there exists some $h \in \mathcal{H}$ and $T \in \mathcal{T}_h$ such that $|E(\mathcal{H}[S_N^{\mathcal{A},T}], (S_X)_N^{\mathcal{A},T})| \geq 2$.

We construct a labeling strategy T' that yields at least $N + 1$ labeled samples as follows:

1. Let $T'(x) = T(x)$ for $x \in S_N^{\mathcal{A},T}$.
2. Let $T'(x) = h(x)$ for $x \in \mathcal{X} \setminus S_N^{\mathcal{A},T}$.

Note that T' is an identifiable oracle for h, since $\mathcal{H}[T'(\mathcal{X})] \subseteq \mathcal{H}[T(\mathcal{X})] = \{h\}$, and $h \in \mathcal{H}[T'(\mathcal{X})]$ by construction.

And so, we have that:

$$\begin{aligned}
\max_{h\in\mathcal{H}, T\in\mathcal{T}_h} CC_{\mathcal{A},T}(\mathcal{H},\emptyset) &\geq CC_{\mathcal{A},T'}(\mathcal{H},\emptyset) \\
&= N + CC_{\mathcal{A},T'}(\mathcal{H}[S_N^{\mathcal{A},T'}], (S_X)_N^{\mathcal{A},T'}) && (\diamond) \\
&= CC(\mathcal{H},\emptyset) - 1 + CC_{\mathcal{A},T'}(\mathcal{H}[S_N^{\mathcal{A},T'}], (S_X)_N^{\mathcal{A},T'}) && (\diamond\diamond) \\
&\geq CC(\mathcal{H},\emptyset) - 1 + 1
\end{aligned}$$

$(\diamond)$: Since $T'(x) = T(x)$ for $x \in S_N^{\mathcal{A},T}$, by Lemma C.1, we must have that $S_N^{\mathcal{A},T'} = S_N^{\mathcal{A},T}$, and $(S_X)_N^{\mathcal{A},T'} = (S_X)_N^{\mathcal{A},T}$.

In particular, note that this implies $|E(\mathcal{H}[S_N^{\mathcal{A},T'}], (S_X)_N^{\mathcal{A},T'})| = |E(\mathcal{H}[S_N^{\mathcal{A},T}], (S_X)_N^{\mathcal{A},T})| \geq 2$.

$(\diamond\diamond)$: Since $\mathcal{A}$ is an exact learning algorithm, it does not terminate at the $|S_N^{\mathcal{A},T'}|$th step, because $|E(S_N^{\mathcal{A},T'}, (S_X)_N^{\mathcal{A},T}))| \geq 2$.

And so, $\mathcal{A}$ will make at least one more query on some $x \in \mathcal{X} \backslash S_N^{\mathcal{A},T'}$. Since $T'(x) \neq \perp$ for any $x \in \mathcal{X} \setminus S_N^{\mathcal{A},T'}$, and T' is identifiable (yielding terminal cost 0), we have that $CC_{\mathcal{A},T'}(\mathcal{H}[S_N^{\mathcal{A},T'}], (S_X)_N^{\mathcal{A},T'}) \geq 1$.

$\square$

D PROOFS FOR SECTION 3

D.1 DEFINITIONS

Definition D.1. *Given $\mathcal{H}, \mathcal{X}$, define the global identification cost of version space V and example set S as*

$$GIC(V, S_X) = \min\{t \in \mathbb{N} : \forall T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}, \exists \Sigma \subseteq \mathcal{X} \setminus S_X \text{ s.t. } \sum_{x \in \Sigma} c(x, T(x)) \le t \wedge |E(V[T(\Sigma)], S_X \cup \Sigma)| \le 1\}.$$

Remark D.2. *Denote by $\Gamma_{V,S_X} : \mathbb{N} \to \{\text{TRUE}, \text{FALSE}\}$ as:*

$$\Gamma_{V,S_X}(t) = \left(\forall T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}, \exists \Sigma \subseteq \mathcal{X} \setminus S_X \text{ s.t. } \sum_{x \in \Sigma} c(x, T(x)) \le t \wedge |E(V[T(\Sigma)], S_X \cup \Sigma)| \le 1 \right)$$

Note that Γ_{V,S_X} is monotonic increasing: for $t_1, t_2 \in \mathbb{N}$, if $t_1 < t_2$, then $\Gamma_{V,S_X}(t_1) \to \Gamma_{V,S_X}(t_2)$. With this notation,

$$\Gamma_{V,S_X}(t) = \begin{cases} \text{TRUE} & t \ge GIC(V, S_X) \\ \text{FALSE} & t \le GIC(V, S_X) - 1 \end{cases}$$

A good way to visualize this is that, on the axis of natural numbers, the value of $\Gamma_{V,S_X}(t)$'s will have the pattern of $\{\text{FALSE}, , ..., \text{FALSE}, \text{TRUE}, \text{TRUE}...\}$, where the turning point is $t = GIC(V, S_X)$.

As a consequence,

$$\begin{aligned} & GIC(V, S_X) \ge N \\ \iff & \Gamma_{V,S_X}(N) = \text{TRUE} \\ \iff & \forall T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}, \exists \Sigma \subseteq \mathcal{X} \setminus S_X \text{ s.t. } \sum_{x \in \Sigma} c(x, T(x)) \le N \wedge |E(V[T(\Sigma)], S_X \cup \Sigma)| \le 1 \end{aligned}$$

$$\begin{aligned} & GIC(V, S_X) \le N \\ \iff & \Gamma_{V,S_X}(N - 1) = \text{FALSE} \\ \iff & \exists T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}, \forall \Sigma \subseteq \mathcal{X} \setminus S_X, \sum_{x \in \Sigma} c(x, T(x)) \le N - 1 \to |E(V[T(\Sigma)], S_X \cup \Sigma)| \ge 2 \end{aligned}$$

D.1.1 LEMMAS

We prove several lemmas on the properties of E-VS and CC.

Lemma D.3. *We have the following:*

1. *For any $x \in \mathcal{X} \setminus S_X$ and $y \in \{-1, 1\}$,*
$$E(V[(x, y)], S_X \cup \{x\}) = E(V, S_X)[(x, y)]$$

2. *For any set of binary-labeled examples $W \subset (\mathcal{X} \times \{-1, 1\})$,*
$$E(V[W], S_X \cup W) = E(V, S_X)[W]$$

Proof. 1.

$$\begin{aligned} & h \in E(V[(x, y)], S_X \cup \{x\}) \\ \iff & h \in V[(x, y)] \wedge \forall h' \in V[(x, y)] \,.\, h' \ne h \to h'(\mathcal{X} \setminus (S_X \cup \{x\})) \ne h(\mathcal{X} \setminus (S_X \cup \{x\})) \\ \iff & h \in V \wedge h(x) = y \wedge \forall h' \in V[(x, y)] \,.\, h' \ne h \to h'(\mathcal{X} \setminus S_X) \ne h(\mathcal{X} \setminus S_X) \\ \iff & h \in V \wedge h(x) = y \wedge \forall h' \in V \,.\, h' \ne h \to h'(\mathcal{X} \setminus S_X) \ne h(\mathcal{X} \setminus S_X) \\ \iff & h(x) = y \wedge h \in E(V, S_X) \\ \iff & h \in E(V, S_X)[(x, y)] \end{aligned}$$

where the first equality uses the definition of effective version space; the second equality uses the fact that for $h, h' \in V[(x,y)]$, $h'(\mathcal{X} \setminus (S_X \cup \{x\})) \neq h(\mathcal{X} \setminus (S_X \cup \{x\}))$ is equivalent to $h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)$; the third equality follows from that for h such that $h(x) = y$, for all $h' \in V$ such that $h'(x) \neq y$, $h'(x) \neq h(x)$ and therefore $h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)$ holds trivially; the fourth equality uses the definition of effective version space; the last equality uses the definition of version space with respect to labeled examples.

2. The claim follows by induction:

Base case. If $|W| = 1$, the claim follows from the previous item.

Inductive case. Assume that $E(V[W'], S_X \cup W') = E(V, S_X)[W']$ holds for any W' such that $|W'| < n$; Now consider any W of size n; W can be represented as $\{(x,y)\} \cup W'$ for some $(x,y) \in \mathcal{X} \times \{-1, 1\}$ and $|W'| = n - 1$. We have:

$$\begin{aligned} E(V[W], S_X \cup W) =& E(V[W'][(x,y)], S_X \cup W' \cup \{x\}) && \text{(Definition of version space)} \\ =& E(V[W'], S_X \cup W')[(x,y)] && \text{(item 1)} \\ =& E(V, S_X)[W'][(x,y)] && \text{(Inductive hypothesis)} \\ =& E(V, S_X)[W] && \text{(Definition of version space)} \end{aligned}$$

This completes the induction.

□

Lemma D.4. *$E(V, S_X) \neq \emptyset$ iff $CC(V, S_X) \geq 0$*

Proof. ($\Leftarrow$) From the first terminal conditions, we know that $E(V, S_X) = \emptyset \implies CC(V, S_X) = -\infty < 0$. So $CC(V, S_X) \geq 0 \implies E(V, S_X) \neq \emptyset$.

($\Rightarrow$) By backward induction on S_X.

Base case. If $S_X = \mathcal{X}$, $|E(V, S_X)| = 0$ or 1. If $|E(V, S_X)| = 1$, we have by the base case of the definition of CC, $CC(V, S_X) = 0$. Therefore, $E(V, S_X) \neq \emptyset \implies CC(V, S_X) \geq 0$.

Inductive case. Suppose $E(V, S_X) \neq \emptyset \implies CC(V, S_X) \geq 0$ holds for any dataset S_X of size $\geq j + 1$. Consider S_X of size j and V such that $E(V, S_X) \neq \emptyset$.

- If $|E(V, S_X)| = 1$, then $CC(V, S_X) = 0 \geq 0$.
- Otherwise, $|E(V, S_X)| \geq 2$; take $h_1 \in E(V, S_X)$; we have

$$CC(V, S_X) \geq \min_x \big(CC(V[(x, h_1(x))], S_X \cup \{x\}) + 1)\big)$$

  By Lemma D.3, $h_1 \in E(V[(x, h_1(x))], S_X \cup \{x\})$, by inductive hypothesis, $CC(V[(x, h_1(x))], S_X \cup \{x\}) \geq 0$, and therefore $CC(V, S_X) \geq 1 \geq 0$.

In summary, $CC(V, S_X) \geq 0$.

This completes the induction. □

Corollary D.5. *$CC(V, S_X) = -\infty$ iff $|E(V, S_X)| = 0$*

Lemma D.6. *$|E(V, S_X)| \geq 2$ iff $CC(V, S_X) \geq 1$.*

Proof. ($\Leftarrow$) From the first two terminal conditions in the definition of CC, we know that if $|E(V, S_X)| \leq 1 \Rightarrow CC(V, S_X) \leq 0$ and so, $CC(V, S_X) \geq 1 \Rightarrow |E(V, S_X)| \geq 2$.

($\Rightarrow$) Let $h_1 \in E(V, S_X)$, consider labeling strategy $T(x) = h_1(x)$ for all $x \in \mathcal{X} \setminus S$ (i.e. never abstains).

Following the definition of $CC(V, S_X)$, we have

$$CC(V, S_X) \geq \min_x \big(CC(V[(x, h_1(x))], S_X \cup \{x\}) + 1)\big)$$

Also, note that by Lemma D.3,

$$E(V[(x, h_1(x))], S_X \cup \{x\}) = E(V, S_X)[(x, h_1(x))] \ni h_1$$

Therefore, by Lemma D.4, for every x, $CC(V[(x, h_1(x))], S_X \cup \{x\}) \geq 0$, and thus $CC(V, S_X) \geq 1$.

□

Corollary D.7. $CC(V, S_X) = 0 \Leftrightarrow |E(V, S_X)| = 1$

Proposition D.8. *For any V, $|E(V, \mathcal{X})| \leq 1$.*

Proof. We consider three cases:

1. If $V = \emptyset$, then $E(V, \mathcal{X}) = \emptyset$
2. If $|V| = 1$, then $E(V, \mathcal{X}) = V$
3. If $|V| \geq 2$, then $E(V, \mathcal{X}) = \emptyset$.

   This is because for any $h \in V$, consider some $h' \in V \setminus \{h\}$. h' trivially agrees with h on $\mathcal{X} \setminus \mathcal{X} = \emptyset$. And so, $h(\emptyset) = h'(\emptyset) \Rightarrow h \notin E(V, \mathcal{X})$.

In summary, in all three cases, $|E(V, \mathcal{X})| \leq 1$. □

Lemma D.9. *Algorithm 2 maintains the invariant that $GIC(V, S_X) \leq GIC(\mathcal{H}, \emptyset)$.*

Proof. It suffices to show that $GIC(V, S_X)$ is nonincreasing throughout. In other words, after obtaining queried sample $(x, T(x))$ during an iteration of the algorithm,

$$GIC(V[T(x)], S_X \cup \{x\}) \leq GIC(V, S_X) \tag{1}$$

Denote by $t = GIC(V, S_X)$. It therefore suffices to show that, for any oracle $T' : \mathcal{X} \setminus (S_X \cup \{x\}) \rightarrow \{-1, +1, \perp\}$, there exists $\Sigma' \subset \mathcal{X} \setminus (S_X \cup \{x\})$ such that:

$$\sum_{x \in \Sigma'} c(x, T'(x)) \leq t \wedge \big|E(V[T(x)][T'(\Sigma')], S_X \cup \{x\} \cup \Sigma')\big| \leq 1. \tag{2}$$

Below we construct such a Σ' for each T'.

First, define oracle $\tilde{T} : \mathcal{X} \setminus S_X \rightarrow \{-1, +1, \perp\}$ as:

$$\tilde{T}(z) = \begin{cases} T(x) & z = x \\ T'(z) & z \neq x \end{cases}$$

By the definition of $GIC(V, S_X)$, for this $\tilde{T}$, there exists $\tilde{\Sigma}$ such that:

$$\sum_{x \in \tilde{\Sigma}} c(x, \tilde{T}(x)) \leq t \wedge \Big|E(V[\tilde{T}(\tilde{\Sigma})], S_X \cup \tilde{\Sigma})\Big| \leq 1. \tag{3}$$

We now construct Σ' differently by considering two cases of $\tilde{\Sigma}$:

1. If $x \in \tilde{\Sigma}$, we construct $\Sigma' := \tilde{\Sigma} \setminus \{x\}$. Note that $\sum_{x \in \Sigma'} c(x, T'(x)) \leq \sum_{x \in \tilde{\Sigma}} c(x, \tilde{T}(x)) \leq t$, and by the definition of $\tilde{T}$, $E(V[T(x)][T'(\Sigma')], S_X \cup \{x\} \cup \Sigma') = E(V[\tilde{T}(x)][\tilde{T}(\tilde{\Sigma} \setminus \{x\})], S_X \cup \{x\} \cup (\tilde{\Sigma} \setminus \{x\})) = E(V[\tilde{T}(\tilde{\Sigma})], S_X \cup \tilde{\Sigma})$ and therefore has size ≤ 1.

2. If $x \notin \tilde{\Sigma}$, we construct $\Sigma' = \tilde{\Sigma}$. Note that $\sum_{x \in \Sigma'} c(x, T'(x)) = \sum_{x \in \tilde{\Sigma}} c(x, \tilde{T}(x)) \leq t$, and:

$$\begin{aligned} & E(V[T(x)][T'(\Sigma')], S_X \cup \{x\} \cup \Sigma') \\ =& E(V[\tilde{T}(\tilde{\Sigma})][T(x)], S_X \cup \tilde{\Sigma} \cup \{x\}) && \text{(since } T'(\Sigma') = \tilde{T}(\tilde{\Sigma})\text{)} \\ \subseteq& E(V[\tilde{T}(\tilde{\Sigma})], S_X \cup \tilde{\Sigma}) && (\diamond) \end{aligned}$$

and therefore has size ≤ 1.

$(\diamond)$: Here the last inequality uses Lemma D.3 (for when $T(x) \in \{+1, -1\}$) and Lemma D.10 (for when $T(x) = \perp$) which implies that for any set $\mathcal{F} \subset \mathcal{H}$ and unlabeled examples U, $E(\mathcal{F}[T(x)], U \cup \{x\}) \subseteq E(\mathcal{F}, U)$.

In summary, there always exists Σ' that satisfies Eq. 2, and therefore Eq. 1 holds for every iteration. This concludes the proof of the lemma. ☐

Lemma D.10. *For any $V \subset \mathcal{H}$ and $S_X \subset \mathcal{X}$,*

$$E(V, S_X \cup \{x^*\}) \subseteq E(V, S_X)$$

Proof. It suffices to prove that $h \in E(V, S_X \cup \{x^*\}) \Rightarrow h \in E(V, S_X)$.

To see this, let $h \in E(V, S_X \cup \{x^*\})$. Then, $\forall h' \in V \setminus \{h\}, h((\mathcal{X} \setminus S_X) \setminus \{x^*\})) \neq h'((\mathcal{X} \setminus S_X) \setminus \{x^*\})) \Rightarrow \forall h' \in V \setminus \{h\}, h(\mathcal{X} \setminus S_X) \neq h'(\mathcal{X} \setminus S_X)$. This implies that $h \in E(V, S_X)$. ☐

D.2 Main Results

In this section, we prove the generalized version of results in Section 3, in which examples may incur differing costs. Let us denote $c(x) = c(x, 1) = c(x, -1)$.

Lemma D.11. *For any V, S_X such that $GIC(V, S_X)$ is finite, $\exists x \in \mathcal{X} \setminus S_X$ such that:*

$$\max_{y \in \{-1,+1\}} \big(|E(V[(x,y)], S_X \cup \{x\}))| - 1\big) \leq (|E(V, S_X)| - 1)\left(1 - \frac{c(x)}{GIC(V, S_X)}\right).$$

Proof. Recall from Lemma D.3 that we have: $E(V[(x,y)], S_X \cup \{x\})) = E(V, S_X)[(x,y)]$, it suffices to prove that there exists $x \in \mathcal{X} \setminus S_X$ such that

$$\max_{y \in \{-1,+1\}} \big(|E(V, S_X)[(x,y)])| - 1\big) \leq (|E(V, S_X)| - 1)\left(1 - \frac{c(x)}{GIC(V, S_X)}\right).$$

Also, note that $|E(V, S_X)| = |E(V, S_X)[(x, -1)]| + |E(V, S_X)[(x, +1)]|$, as $E(V, S_X)[(x, -1)]$ and $E(V, S_X)[(x, +1)]$ form a disjoint partition of $E(V, S_X)$.

And so, equivalently, it suffices to show that there exists $x \in \mathcal{X} \setminus S_X$ such that:

$$\min\big(|E(V, S_X)[(x,-1)]|, |E(V, S_X)|[(x,+1)]\big) \geq c(x)\frac{|E(V, S_X)| - 1}{GIC(V, S_X)}$$

So, assume towards contradiction that the statement above does not hold. Then, we have that $\forall x \in \mathcal{X} \setminus S_X$:

$$\min\big(|E(V, S_X)[(x,-1)]|, |E(V, S_X)|[(x,+1)]\big) < c(x)\frac{|E(V, S_X)| - 1}{GIC(V, S_X)} \tag{4}$$

Define oracle $T_0 : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}$ such that,

$$T_0(x) = \operatorname*{arg\,max}_{y \in \{-1,1\}} |E(V, S_X)[(x,y)]|$$

With this, for every subset $\Sigma \subseteq \mathcal{X} \setminus S_X$ such that $\sum_{x\in\Sigma} c(x, T_0(x)) \le GIC(V, S_X)$, we have:

$$
\begin{aligned}
|E(V[T_0(\Sigma)], S_X \cup \Sigma)| &= |E(V, S_X)[T_0(\Sigma)]| && \text{(Lemma D.3, item 2)}\\
&= |E(V, S_X)| - |\{h \in E(V, S_X) : \exists x \in \Sigma, h(x) \neq T_0(x)\}| && \text{(Set algebra)}\\
&\ge |E(V, S_X)| - \sum_{x\in\Sigma} |E(V, S_X)[(x, \neg T_0(x))]| && \text{(Union bound)}\\
&= |E(V, S_X)| - \sum_{x\in\Sigma} \min_{y\in\{+1,-1\}} |E(V, S_X)[(x, y)]| && \text{(by definition of } T_0(x))\\
&> |E(V, S_X)| - \sum_{x\in\Sigma} c(x, T_0(x)) \frac{|E(V, S_X)| - 1}{GIC(V, S_X)} && \text{(by Equation 4 and } c(x) = c(x, T_0(x)) \text{ since } T_0(x) \in \{-1, +1\})\\
&\ge |E(V, S_X)| - (|E(V, S_X)| - 1) = 1,
\end{aligned}
$$

In summary, for any $\Sigma \subseteq \mathcal{X} \setminus S_X$ such that $\sum_{x\in\Sigma} c(x, T_0(x)) \le GIC(V, S_X)$, $|E(V[T_0(\Sigma)], S_X \cup \Sigma)| > 1$. Therefore, $\Gamma_{V,S_X}(GIC(V, S_X)) =$ FALSE, which contradicts the definition of $GIC(V, S_X)$. □

Lemma D.12. *For any $V \subset \mathcal{H}$ and $S_X \subset \mathcal{X}$,*

$$GIC(V, S_X) \le CC(V, S_X)$$

Proof. Let $k = GIC(V, S_X) - 1$. By the definition of GIC, $\Gamma_{V,S_X}(k) =$ FALSE. That is:

$$\exists T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}, \forall \Sigma \subseteq \mathcal{X} \setminus S_X, \sum_{x\in\Sigma} c(x, T(x)) \le k \Rightarrow \left|E(V[T(\Sigma)], S_X \cup \Sigma)\right| \ge 2 \tag{5}$$

Let T be a labeling oracle that satisfies the properties in Equation 5. Let U be the output of executing the following algorithm that simulates the interaction between a specific label query strategy and the oracle T before a stopping criterion is reached:

Protocol 5 Simulation process on letting T interacting with a targeted label query strategy

$U \leftarrow \emptyset$
while $U \neq \mathcal{X} \setminus S_X$ and $\sum_{x\in U} c(x, T(x)) \le k - 1$ **do**
  Choose example

$$x = \operatorname*{arg\,min}_{x\in\mathcal{X}\setminus(S_X\cup U)} c(x, T(x)) + CC\left(V[T(U \cup \{x\})], S_X \cup U \cup \{x\}\right). \tag{6}$$

  $U \leftarrow U \cup \{x\}$
return U

We first claim that $\sum_{x\in U} c(x, T(x)) = k$. Suppose not, we have $\sum_{x\in U} c(x, T(x)) \le k - 1$. By the stopping criterion of Algorithm 5, we must have that $U = \mathcal{X} \setminus S_X$. In this case, by Equation 5, $|E(V[T(U)], S_X \cup U)| = |E(V[T(U)], \mathcal{X})| \ge 2$. However, this contradicts Proposition D.8 that for any V, $|E(V[T(U)], \mathcal{X})| \le 1$. Therefore, $\sum_{x\in U} c(x, T(x)) = k$.

Denote by $x_1, \ldots, x_m$ the sequence of m examples queried by Algorithm 5; with this notation, $U = \{x_1, \ldots, x_m\}$. Also, for $i \in \{0, 1, \ldots, m\}$, denote by $U_i := \{x_1, \ldots, x_i\}$ the set of first i examples queried.

We make two observations:

- For any $i \in \{0, 1, \ldots, m-1\}$, by the loop condition, $\sum_{x\in U_i} c(x, T(x)) \le k-1$, therefore by Equation 5, $\left|E(V[T(U_i)], S_X \cup U_i)\right| \ge 2$, and therefore, by the definition of CC,

$$CC(V[T(U_i)], S_X \cup U_i) = \min_{x\in\mathcal{X}\setminus(S_X\cup U_i)} \max_{y\in\{-1,+1,\perp\}} \left(c(x, y) + CC(V[T(U_i)][(x, y)], S_X \cup U_i \cup \{x\})\right) \tag{7}$$

- Since $\sum_{x \in U} c(x, T(x)) = k$, by Equation 5, we also have $\left|E(V[T(U)], S_X \cup U)\right| \geq 2$ and by Lemma D.6, $CC(V[T(U)], S_X \cup U) \geq 1$.

Based on these observations, we have:

$$\begin{aligned}
CC(V, S_X) &= \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{-1,+1,\perp\}} \big(c(x,y) + CC(V[(x,y)], S_X \cup \{x\})\big) && \text{(Eq. 7 with } i = 0) \\
&\geq \min_{x \in \mathcal{X} \setminus S_X} \big(c(x, T(x)) + CC(V[T(\{x\})], S_X \cup \{x\})\big) \\
&= c(x_1, T(x_1)) + CC(V[T(U_1)]), S_X \cup U_1) && \text{(Eq. 6)} \\
&= c(x_1, T(x_1)) + \min_{x \in \mathcal{X} \setminus (S_X \cup U_1)} \max_{y \in \{-1,+1,\perp\}} \big(c(x,y) + CC(V[T(U_1)][(x,y)], S_X \cup U_1 \cup \{x\})\big) \\
& && \text{(Eq. 7 with } i = 1) \\
&\geq \dots \\
&\geq \sum_{i=1}^{m} c(x_i, T(x_i)) + CC(V[T(U)], S_X \cup U) \\
& && \text{(Repeated application of Eqs. 7 and 6)} \\
&\geq k + 1 = GIC(V, S_X). && \text{(since } CC(V[T(U)], S_X \cup U) \geq 1)
\end{aligned}$$

□

Theorem D.13. *If Algorithm 2 interacts with a labeling oracle T, then it incurs total query cost at most $GIC(\mathcal{H}, \emptyset) \ln |\mathcal{H}| + 1$. Furthermore, if Algorithm 2 interacts with an identifiable oracle T consistent with some $h^* \in \mathcal{H}$, then it identifies h^*.*

Proof. First, we show that Algorithm 2 terminates and correctly identifies h^* when interacting with an identifiable oracle of h^*. Its termination can be seen by the fact that the size of S_X is increasing by 1 for each iteration and $S_X \neq \mathcal{X}$ is part of the stopping criterion.

We now show that when it returns, $E(V, S_X) = \{h^*\}$. This can be seen by:

- As T is an identifiable oracle that is consistent with h^*, the algorithm maintains the invariant that $h^* \in E(V, S_X)$.

  This is because if at some point $h^* \notin E(V, S_X)$, then exists some $h' \neq h$ such that $h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X)$. Then, we combine with that $h' \in \mathcal{H}[T(S_X)]$ to get that $h' \in \mathcal{H}[T(S_X) \cup h(\mathcal{X} \setminus S_X)] \subseteq \mathcal{H}[T(S_X) \cup T(\mathcal{X} \setminus S_X)] = \mathcal{H}[T(\mathcal{X})]$, which is in contradiction with that T is an identifiable oracle.

- We claim that when it returns, $|E(V, S_X)| = 1$. Since the E-VS always contains h^*, we must have $|E(V, S_X)| \geq 1$.

  And so, if it returns and $|E(V, S_X)| \neq 1 \Rightarrow |E(V, S_X)| \geq 2$, then we must have $S_X = \mathcal{X}$, which contradicts Proposition D.8.

Next we bound the query cost complexity of Algorithm 2, when interacting with any labeling oracle.

Denote V_i and S_i as the value of V and S_X at the i-th iteration, and denote (x_i, y_i) by the example (x, y) obtained at the i-th iteration.

Therefore, $V_{i+1} = V[(x_i, y_i)]$ and $S_{i+1} = S_i \cup \{x_i\}$.

We claim that

$$\left(\left|E(V_{i+1}, S_{i+1})\right| - 1\right) \leq \left(\left|E(V_i, S_i)\right| - 1\right) \cdot \exp\left(-\frac{c(x_i)}{GIC(\mathcal{H}, \emptyset)}\right). \tag{8}$$

To see this, we consider two cases:

1. If $y_i \in \{-1, +1\}$, then applying Lemma D.11 with $V = V_i$, $S_X = S_i$, $x = x_i$, we have

$$
\begin{aligned}
(|E(V_{i+1}, S_{i+1})| - 1) \leq & \max_{y \in \{-1,+1\}} \left(\left| E(V_i[(x_i, y)], S_{i+1}) \right| - 1 \right) \\
\leq & (|E(V_i, S_i)| - 1) \left(1 - \frac{c(x_i)}{GIC(V_i, S_i)} \right) \\
& \text{(Lemma D.11 since } y_i \in \{-1, +1\}) \\
\leq & (|E(V_i, S_i)| - 1) \left(1 - \frac{c(x_i)}{GIC(\mathcal{H}, \emptyset)} \right) \\
& \text{(by Lemma D.9, } GIC(V_i, S_i) \leq GIC(\mathcal{H}, \emptyset)) \\
\leq & (\left| E(V_i, S_i) \right| - 1) \cdot \exp \left(-\frac{c(x_i)}{GIC(\mathcal{H}, \emptyset)} \right). \quad \text{(since } 1 - x \leq e^{-x})
\end{aligned}
$$

2. If $y_i = \perp$, $c(x_i, y_i) = 0$. Therefore, to show Equation 8, it suffices to show that $E(V_{i+1}, S_{i+1}) \subseteq E(V_i, S_i)$. This follows from Lemma D.10.

To summarize, Equation 8 holds for each iteration i.

Consider the last iteration i_0 before the termination condition is reached; note that by the termination criterion, the penultimate E-VS is such that $|E(V_{i_0}, S_{i_0})| \geq 2$. We now upper bound the total cost up to iteration $i_0 - 1$. By repeatedly using Eq. 8 for $i = 1, \ldots, i_0 - 1$, we have:

$$
1 \leq \left| E(V_{i_0}, S_{i_0}) \right| - 1 \leq \left| E(\mathcal{H}, \emptyset) \right| \cdot \exp \left(-\frac{\sum_{i=1}^{i_0-1} c(x_i, y_i)}{GIC(\mathcal{H}, \emptyset)} \right)
$$

Therefore, $\sum_{i=1}^{i_0-1} c(x_i, y_i) \leq GIC(\mathcal{H}, \emptyset) \ln |\mathcal{H}|$ (since $E(\mathcal{H}, \emptyset) = \mathcal{H}$) and:

$$
\sum_{i=1}^{i_0} c(x_i, y_i) = c(x_{i_0}, y_{i_0}) + \sum_{i=1}^{i_0-1} c(x_i, y_i) \leq GIC(\mathcal{H}, \emptyset) \ln |\mathcal{H}| + 1.
$$

$\square$

E PROOFS FOR SUBSECTIONS 3.1, 3.2 AND 3.3

E.1 COMPARING VS VERSUS E-VS

Consider the case when $\mathcal{H}$ is linear. In this setting, the (conventional) version space is a single polytope, which we may access by sampling using any polytope sampler. The structural lemma below illustrates that, by contrast, the E-VS can be a more complicated object to access.

Proposition E.1. *There exists an instance space $\mathcal{X} \subset \mathbb{R}^d$ and query responses S such that the resultant E-VS includes an exponential in d number of disjoint polytopes.*

Proof. **Defining the Learning Task:** Define $\mathcal{H} = \left\{h_w(x) = \text{sign}(w^T x) | w = [w', 1], w' \in [0, 1]^d\right\}$. We observe that, for any set of points $\mathcal{X}$, $\mathcal{X}$ divide polytope $\left\{w = [w', 1] : w' \in [0, 1]^d\right\}$ into cells, where every point in the cell has the same labeling of $\mathcal{X}$, and different cells have different labelings of $\mathcal{X}$. Thus, without loss of generality, we can treat each cell formed by $\mathcal{X}$ as an element of $\mathcal{H}$, and $\mathcal{H}$ comprises of all the cells that lie in polytope $\left\{w = [w', 1] : w' \in [0, 1]^d\right\}$.

Now, we construct a $\mathcal{X}$ that allows us to easily reason about the E-VS. Consider any $3n$ positive reals a_k^j for $j \in [n], k \in [3]$ such that $0 < a_1^1 < a_2^1 < a_3^1 < ... < a_3^n < 1$. Define $x_{jk}^i = [-e_i, a_k^j]$ for $i \in [d]$. As a concrete example, $x_{23}^1 = [-1, 0, ..., a_2^3]$.

Define the instance space to be $\mathcal{X} = \left\{x_{jk}^i | i \in [d], j \in [n], k \in [3]\right\}$. With $\mathcal{X}$ defined, we see the cells formed by $\mathcal{X}$ consists of: $\times_{i=1}^d I$, where $I = \left\{[0, a_1^1], [a_1^1, a_2^1], [a_2^1, a_3^1], ..., [a_3^n, 1]\right\}$.

Now, define the interaction history $S = \left\{(x_{jk}^i, \perp) | i \in [d], j \in [n], k = 2\right\}$. Note that then $S_X = S^\perp = \left\{x_{jk}^i | i \in [d], j \in [n], k = 2\right\}$.

Characterizing the E-VS: We first claim that for any cell with one of its faces a subset of a hyperplane in $S^\perp$ cannot be in the E-VS. Specifically, if there $\exists i \in [d], j \in [n]$ such that $w_i \in [a_1^j, a_3^j]$, then the cell w belongs to is not in the E-VS.

To see this, WLOG $w_i \in [a_1^j, a_2^j]$.

Now, construct $\tilde{w} = [w_1, ..., w_{i-1}, \tilde{w}_i, w_{i+1}, ...1]$, for some $\tilde{w}_i \in [a_2^j, a_3^j]$. Note that by construction, w' does not lie in the same cell as w. Then, we see that $\text{sign}(w'^T x) = \text{sign}(w'^T x), \forall x \in \mathcal{X} \setminus \left\{x_{j2}^i\right\}$. And so, since $\mathcal{X} \setminus S^\perp \subseteq \mathcal{X} \setminus \left\{x_{j2}^i\right\}$, we have that $w(\mathcal{X} \setminus S^\perp) = w'(\mathcal{X} \setminus S^\perp) \Rightarrow w \notin E(V, S_X)$.

This means that only the set of disjoint cells $\times_{i=1}^d I'$, where $I' = \left\{[0, a_1^1], [a_3^1, a_1^2], \ldots, [a_3^n, 1]\right\}$, can be in the E-VS. Next, we will argue that the E-VS is all of $\times_{i=1}^d I'$.

Consider a classifier corresponding to some cell $c \in \times_{i=1}^d I'$. Consider any other cell classifier corresponding to cell $c' \in \times_{i=1}^d I$. Since $c \neq c'$, there must be at least one dimension, WLOG i, such that c and c' belong to different sub-intervals, when projected onto coordinate i.

We know that along dimension i, c's sub-interval is either of the form $[0, a_1^1]$, $[a_3^j, a_1^{j+1}]$ for some j, or $[a_3^n, 1]$.

We see that in the first case, $x_{11}^i \in \mathcal{X} \setminus S^\perp$ must separate c and c', since $c(x) = +1 \neq -1 = c'(x)$. Analogously, in the second case, either x_{j3}^i or $x_{(j+1)1}^i$ must separate c and c' (with both such points are in $\mathcal{X} \setminus S^\perp$). Finally, in the last case, $x_{n3}^i \in \mathcal{X} \setminus S^\perp$ must separate c and c'.

This shows that all of $\times_{i=1}^d I'$ is in the E-VS. And so, since I' comprises of $n + 1$ disjoint intervals, there are in total $(n + 1)^d$ number of disjoint cells, corresponding to distinct classifiers. □

E.2 E-VS MEMBERSHIP CHECK

The key idea behind the membership check $h \in E(V, S_X)$ is that we want to find a hypothesis $\hat{h}$ in V, different from h, that agrees on the rest of the unqueried samples. If we succeed in finding this

$\hat{h}$, then this means that even if all of the remaining unqueried samples $\mathcal{X} \setminus S_X$ is labeled, h and $\hat{h}$ cannot be distinguished from each other. This implies that h is non-identifiable and does not belong to the E-VS.

Proposition E.2. *Given some $h \in \mathcal{H}$ and access to a* C-ERM *oracle, one can verify $h \in E(V, S_X)$ with one call to the* C-ERM *oracle.*

Proof. Firstly, note that by definition, $\forall h, h' \in \mathcal{H}$, $h \neq h' \Rightarrow h(\mathcal{X}) \neq h'(\mathcal{X})$.

Now, we rewrite the definition of not being in the E-VS:

$$\begin{aligned} h \notin E(V, S_X) &\Leftrightarrow \exists h' \in V \setminus \{h\}, h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \\ &\Leftrightarrow \exists h', h'(S_X \setminus S^\perp) = y_{S_X \setminus S^\perp} = h(S_X \setminus S^\perp) \wedge h'(\mathcal{X}) \neq h(\mathcal{X}) \wedge h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \\ &\Leftrightarrow \exists h', h'(S_X \setminus S^\perp) = y_{S_X \setminus S^\perp} = h(S_X \setminus S^\perp) \wedge h'(S^\perp) \neq h(S^\perp) \wedge h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \\ &\Leftrightarrow \exists h', \exists x^\perp \in S^\perp, h'(S_X \setminus S^\perp) = y_{S_X \setminus S^\perp} = h(S_X \setminus S^\perp) \wedge h'(x^\perp) \neq h(x^\perp) \wedge h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \end{aligned}$$

And so, we may check for the existence of such a h' with one C-ERM call on $\mathcal{H}$, given some $h \in V$ (note that by construction $h \in V \Rightarrow h(S_X \setminus S^\perp) = y_{S_X \setminus S^\perp}$).

We are interested in finding $\hat{h}$ of the following program:

$$\begin{aligned} &\min_{h' \in \mathcal{H}} \sum_{x' \in S^\perp} \mathbb{1}\left\{h'(x') = h(x')\right\} \\ &\text{s.t } h'(x) = h(x), \forall x \in \mathcal{X} \setminus S^\perp \end{aligned} \tag{9}$$

This may be emulated by defining data $Z_1 = \left\{(x, \neg h(x))\right\}_{x \in S^\perp}$, $Z_2 = \left\{(x, h(x))\right\}_{x \in \mathcal{X} \setminus S^\perp}$, and calling C-ERM on Z_1, Z_2 to compute $\hat{h} \in \arg\min\left\{\text{err}(h', Z_1) : h' \in \mathcal{H}, \text{err}(h', Z_2) = 0\right\}$.

It suffices to test: if C-ERM output $\hat{h} \neq h \Rightarrow h \notin E(V, S_X)$ ☐

E.3 Contrasting E-VS bisection Algorithm with VS bisection

E.3.1 Paired interval-threshold hypothesis learning setting

Setup: Our example will revolve around the hybrid-hypothesis class of thresholds and intervals. Let $n \geq 8$.

Let the $f_i : [0, 2] \to \{+1, -1\}$ denote intervals of length $1/n$, $f_i(x) = \mathbb{1}(x \in [(i-1)/n, i/n])$ for $i \in [n-1]$.

Let $f_i' : [0, 2] \to \{+1, -1\}$ denote thresholds, $f_i'(x) = \mathbb{1}(x \geq 1 + i/n)$ for $i \in [n]$.

Define $\mathcal{H} = \bigcup_{i=1}^{n-1} \left\{(f_i, f_i'), (f_i, f_{i+1}')\right\}$.

Let $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$, where $\mathcal{X}_1 = \left\{x_1^1, ..., x_{n-1}^1\right\} = \left\{[\frac{1}{2n}, 0], ..., [\frac{2n-4+1}{2n}, 0]\right\}$ and $\mathcal{X}_2 = \left\{x_1^2, ..., x_{n-1}^2\right\} = \left\{[2, 1 + \frac{3}{2n}], ..., [2, 1 + \frac{2n-1}{2n}]\right\}$.

So $|\mathcal{X}| = 2(n-1)$.

Note that for $\mathcal{X}_1$, the second coordinate gives no information on f_i' (all -1 label), and for $\mathcal{X}_2$ the first coordinate gives no information on f_i (all -1 label).

E.3.2 Algorithm Analysis

Under the paired interval-threshold setup, we compare the algorithms based on the number of samples queried before termination.

In the case of the VS-bisection algorithm, it queries the point that maximally bisects the VS each time. Accordingly, the algorithm terminates when there is no point that bisects the VS. This arises either because the set of unqueried points is non-empty but the VS agrees on all of the points' labels, or the set of unqueried points is empty.

While for the E-VS bisection algorithm, it terminates either when the E-VS is of cardinality zero or of one.

Lemma E.3 (E-VS bisection algorithm query complexity)**.** *In the paired interval-threshold hypothesis learning setting, the E-VS algorithm incurs $O(\log n)$ sample complexity against any labeling oracle.*

Proof. Define $\rho(E(V, S_X), x) = \min_{y \in \{+1,-1\}} |E(V, S_X)[x, y]|$.

1. Let $U_2 \subseteq \mathcal{X}_2$ denote the unlabeled part of $\mathcal{X}_2$ such that $U_2 = \{x : \rho(E(V, S_X), x) > 0, x \in \mathcal{X}_2\}$ (i.e. $x \in \mathcal{X}_2$ is in the disagreement region formed by the current E-VS).

   Definition E.4. *A point $x \in U_2$ is balanced if there exists a three-point segments with $x_i^2 + 2/n = x_{i+1}^2 + 1/n = x_{i+2}^2$, $x_j^2 + 2/n = x_{j+1}^2 + 1/n = x_{j+2}^2$ such that $x_{i+2}^2 < x < x_j^2$, where points $x_i^2, x_{i+1}^2, x_{i+2}^2 \in U_2$, and $x_j^2, x_{j+1}^2, x_{j+2}^2 \in U_2$.*

   We have that, if:

   a) x is a balanced point

   b) all queried points thus far have been in $\mathcal{X}_2$, then:

   $$\rho(E(V, S_X), x) \geq 2 = \max_{x' \in \mathcal{X}_1} \rho(E(V, S_X), x')$$

   This follows because if no points have been queried in $\mathcal{X}_1$, $x_i^2, x_{i+1}^2, x_{i+2}^2 \in U_2$ implies that (f_{i+1}, f'_{i+1}) and $(f_{i+1}, f'_{i+2}) \in E(V, S_X)$. Similarly, $x_j^2, x_{j+1}^2, x_{j+2}^2 \in U_2$ implies that (f_{j+1}, f'_{j+1}) and $(f_{j+1}, f'_{j+2}) \in E(V, S_X)$.

   Since $x_{i+2}^2 < x < x_j^2$, the two pairs of models disagree on x (in the second coordinate).

   And so, if there is some point $x \in U_2$ that is balanced, and all points queried thus far have been in $\mathcal{X}_2$, then the E-VS algorithm will query a point in U_2 (we assume that in a tie-breaker, the E-VS algorithm will select the point in $\mathcal{X}_2$).

2. From Lemma E.5, we have that the E-VS algorithm will query some point in $U_2 \subseteq \mathcal{X}_2$ so long as $|U_2| \geq 7$.

   The number of binary labeled samples needed to reach $|U_2| < 7$ is at most $\log n$. This because abstention decreases $|U_2|$ by 1, while a binary label removes $\lfloor |U_2|/2 \rfloor$ points from U_2.

   And so, since $|U_2| = n$, there can be at most $\log n$ binary labeled examples before $|U_2| < 7$.

3. It remains to count the number of binary label samples needed when $|U_2| < 7$ before the interaction finishes.

   We note that if $|U_2| < 7$, then the size of the $|E(V, S_X)| \leq 2 \cdot 6$.

   As each binary label point removes at least one hypothesis from the E-VS, at most 11 more binary label points are needed.

   In summary, we have that the E-VS algorithm incurs $O(\log n)$ samples.

Below are the deferred lemmas:

Lemma E.5. *If $|U_2| \geq 7$, then the E-VS algorithm will query some point $x \in U_2 \subseteq \mathcal{X}_2$.*

Proof. We will show the following properties about U_2^t, which is U_2 at the tth step.

If $|U_2^t| \geq 7$, then:

i) U_2^t is of the form $\{a_1 : b_1\} \cup \{b_2 : a_2\}$, where $b_1 \leq b_2$ ($\{a_1 : b_1\}$ is used to abbreviate $\{a_1, a_1 + 1/n, ..., b_1 - 1/n, b_1\}$).

ii) Some $x \in \{b_1, b_2\}$ satisfies the following: $|| \{x' \in U_2^t : x' < x\} | - | \{x' \in U_2^t : x' > x\} || \leq 1$.

iii) No points $x_1, ..., x_{t-1}$ will have been queried from $\mathcal{X}_1$.

iv) E-VS will query some point $x \in U_2^t$ at step t.

We will see that, at step t, proving property i), ii), iii) proves iv), which is the desired result.

We prove by induction on j, the number of queries, that i), ii), iii) and thus iv) holds.

Base Case: When $j = 0$, no points have been queried from $\mathcal{X}_1$. And so, properties i)-iii) are true with $U_2 = \left\{1 + 3/2n : 1 + (2n-1)/2n\right\}$. Since $n \geq 8$, $|U_2| = |\mathcal{X}_2| = 7$, and so Lemma E.6 applies, meaning iv) is satisfied.

Induction Step: Suppose that if $|U_2^j| \geq 7$, properties i)-iv) holds for time step $j = 0, ..., k-1$.

Now consider time step $j = k$. Suppose $|U_2^k| \geq 7$.

This means that, at time step $k-1$, $|U_2^{k-1}| \geq |U_2^k| \geq 7$ (since the disagreement region only decreases in size).

From induction hypothesis, we know U_2^{k-1} satisfies i)-iv). Let $U_2^{k-1} = \left\{a_1' : b_1'\right\} \cup \left\{b_2' : a_2'\right\}$. Since iv) holds at time $j = k-1$ ($x_{k-1} \in \mathcal{X}_2$), combined with that iii) applies at time $k-1$ ($x_1, ..., x_{k-2} \in \mathcal{X}_2$) implies property iii) holds at time $j = k$ ($x_1, ..., x_{k-1} \in \mathcal{X}_2$)).

Since iv) is satisfied at time step $k-1$, we may WLOG $x_{k-1} = b_1'$. There are two cases to consider:

- If a label is given for x_{k-1}, then we know that U_2^k is either $\left\{a_1' : b_1' - 1/n\right\}$ or $\{b_2 : a_2\}$, in either case, both i) and ii) are satisfied at step $j = k$.
- If an abstention is given for x_{k-1}, then we know that $U_2^k = \left\{a_1' : b_1' - 1/n\right\} \cup \left\{b_2' : a_2'\right\}$, which proves i).

  Since $x_{k-1} = b_1'$, we have that $|| \left\{a_1' : b_1'\right\} | - | \left\{b_2' : a_2'\right\} || \leq 1$.

  If $| \left\{b_2' : a_2'\right\} | \geq | \left\{a_1' : b_1'\right\} |$, picking b_2' satisfies the property, else picking $b_1' - 1/n$ satisfies the property. And so, property ii) for U_2^k holds.

Finally, since iii), i) and ii) holds for U_2^k, using Lemma E.6, we have that $x_k \in \mathcal{X}_2$, which means that iv) holds at $j = k$.

□

Lemma E.6. *If $|U_2^t| \geq 7$, and i)-iii) holds at step t: the E-VS algorithm will query one of $b_1, b_2 \in U_2^t$.*

Proof. Due to ii), we know at least one of b_1, b_2 satisfies $|| \left\{x' \in U_2^t : x' < x\right\} | - | \left\{x' \in U_2^t : x' > x\right\} || \leq 1$.

WLOG let this be b_1 (assume that b_1 wins the E-VS algorithm tie-breaker if both b_1, b_2 satisfy this condition). We claim the E-VS algorithm will query b_1.

- For points in $\mathcal{X}_2 \setminus U_2^t$, they are not in the disagreement region and $\rho(E(V, S_X), x) = 0$, which means they will not be queried.
- For points in U_2^t, we have the following observation.

  Due to i) and iii):

$$\begin{aligned}\rho(E(V, S_X), x) &= \min(2 \cdot | \left\{x' \in U_2^t : x' < x\right\} | + 1, 2 \cdot | \left\{x' \in U_2^t : x' > x\right\} | + 1) \\ &= 2 \cdot \min(| \left\{x' \in U_2^t : x' < x\right\} |, | \left\{x' \in U_2^t : x' > x\right\} |) + 1\end{aligned}$$

  From this, we can see that from ii),

$$\begin{aligned}b_1 &= \arg\max_{x \in U_2^t} \min(| \left\{x' \in U_2^t : x' < x\right\} |, | \left\{x' \in U_2^t : x' > x\right\} |) \\ &= \arg\max_{x \in U_2^t} \rho(E(V, S_X), x)\end{aligned}$$

- For points $x \in \mathcal{X}_1$.

  We know that $|U_2^t| \geq 7 \Rightarrow \min(|\left\{x' \in U_2^t : x' < b_1\right\}|, |\left\{x' \in U_2^t : x' > b_1\right\}|) \geq 3$.

  Due to i), we know that $\left\{x' \in U_2^t : x' < b_1\right\}$ and $\left\{x' \in U_2^t : x' > b_1\right\}$ are contiguous. And so, one can find three-point segments to the left and right of b_1, which means that b_1 is balanced.

  And so, $\rho(E(V, S_X), b_1) \geq 2 = \max_{x \in \mathcal{X}_1} \rho(E(V, S_X), x)$.

In conclusion, b_1 is the point that maximally bisects the E-VS out of all unqueried points, and will thus be queried by the E-VS bisection algorithm. □

□

Theorem E.7. *There exists a $\mathcal{H}$ and $\mathcal{X}$ such that the number of labeled examples queried by the E-VS bisection algorithm is $O(\log |\mathcal{X}|)$, while the VS bisection algorithm queries $\Omega(|\mathcal{X}|)$.*

Proof. From Lemma E.3, we have shown the first part of the theorem. It remains to analyze the VS bisection query complexity.

VS bisection algorithm complexity: By contrast, we show that there exists a labeling oracle that induces $\Omega(n)$ sample complexity from the VS algorithm.

This labeling oracle T is as follows:

i) $T(x) = \perp$ for all $x \in \mathcal{X}_2$

ii) $T(x) = -1$ for all $x \in \mathcal{X}_1$

Under T, we have that labeling each point $x \in \mathcal{X}_1$ removes two hypotheses from the version space at any step in time. Namely, labeling $x_i^1 = [\frac{2i-1}{2n}, 0]$ removes $(f_i, f_i'), (f_i, f_{i+1}')$.

And so, $|\mathcal{X}_1| - 1$ samples $x \in \mathcal{X}_1$ will be queried. Because if there exists two unqueried points $x_i^1, x_j^1 \in \mathcal{X}_1$, then (f_i, f_i') and (f_j, f_j') are both in the VS. This means that the disagreement region is non-empty, and in particular contains both x_i^1, x_j^1.

Since each $x \in \mathcal{X}_1$ is given a binary label by T, the VS bisection algorithm incurs cost $n - 1$. We note that in the end the VS will be of size 2, but the remaining sample in $\mathcal{X}_1$ cannot distinguish between the two.

□

We may also obtain a corresponding result for an identified setting, by tweaking the above setting slightly. In this setting, we still find that the VS-bisection algorithm still incurs an exponentially larger sample complexity relative to E-VS bisections.

Proposition E.8. *There exists a $\mathcal{H}$, $\mathcal{X}$, and a labeling oracle that leads to identification, and the number of labeled examples queried by the E-VS bisection algorithm is $O(\log |\mathcal{X}|)$, while the VS bisection algorithm incurs $\Omega(|\mathcal{X}|)$ samples.*

Proof. **Setup:**

Let the $f_i : [-1, 2] \to \{+1, -1\}$ denote intervals of length $1/n$, $f_i(x) = \mathbb{1}(x \in [(i-1)/n, i/n])$ for $i \in [n-1]$.

Let $f_i' : [0, 2] \to \{+1, -1\}$ denote thresholds, $f_i'(x) = \mathbb{1}(x \geq 1 + i/n)$ for $i \in [n]$.

Define $\mathcal{H}_{pair} = \bigcup_{i=1}^{n-1} \left\{(f_i, f_i'), (f_i, f_{i+1}')\right\}$.

Let $\mathcal{X}_{main} = \mathcal{X}_1 \cup \mathcal{X}_2$, where $\mathcal{X}_1 = \left\{x_1^1, ..., x_{n-1}^1\right\} = \left\{[\frac{1}{2n}, 0], ..., [\frac{2n-4+1}{2n}, 0]\right\}$ and $\mathcal{X}_2 = \left\{x_1^2, ..., x_{n-1}^2\right\} = \left\{[2, 1 + \frac{3}{2n}], ..., [2, 1 + \frac{2n-1}{2n}]\right\}$.

Note that for $\mathcal{X}_1$, the second coordinate gives no information on f_i' (all -1 label), and for $\mathcal{X}_2$ the first coordinate gives no information on f_i (all -1 label).

Ensuring identifiability:

Define an extra interval, $f_0 : [-1, 2] \to \{+1, -1\}$, $f_0(x) = \mathbb{1}(x \in [-1/n, 0])$ and introduce one new data point $\tilde{x} = [-1/2n, 0]$.

So $|\mathcal{X}| = 2(n-1) + 1$.

Now define the extra model $\tilde{f} = (f_0, f_1')$.

Let $\mathcal{H} = \mathcal{H}_{pair} \cup \{f_0\}$ and let $\mathcal{X} = \mathcal{X}_{main} \cup \{\tilde{x}\}$.

Note that obtaining $(\tilde{x}, [1, -1])$ identifies $\tilde{f}$.

E-VS bisection algorithm complexity:

Note that for any V, S_X, $\rho(E(V, S_X), \tilde{x}) \leq 1$.

And so, in the case analysis of Lemma E.6, we again find that as long as $|U_2| \geq 7$, the E-VS algorithm will query some point $x \in U_2$.

Thus, the E-VS algorithm will query at most $\log n$ labeled samples before reaching $|U_2| \leq 6$, at which point the E-VS contains at most $2 \cdot 6 + 1$ hypotheses and will thus require at most 12 more labeled examples before identification.

VS bisection algorithm complexity: We show that there exists an identifiable labeling oracle that induces $\Omega(n)$ samples with the VS algorithm.

This labeling oracle T goes as follows:

i) $T(x) = \perp$ for all $x \in \mathcal{X}_2$

ii) $T(x) = -1$ for all $x \in \mathcal{X}_1$

iii) $T(\tilde{x}) = 1$

It is clear that $\mathcal{H}[T(\mathcal{X})] = \left\{\tilde{h}\right\}$ and T is an identifiable oracle.

The main observation is that while $|S_X \cap \mathcal{X}_1| < |\mathcal{X}_1| - 1$, if a point in $\mathcal{X} \setminus \mathcal{X}_2$ is queried, then it will be a point in $\mathcal{X}_1$, and not $\tilde{x}$.

This is because $\tilde{x}$ for any V, S_X, is such that $\rho(E(V, S_X), \tilde{x}) = 1$. While for any $x \in \mathcal{X}_1 \setminus S_X$, $\rho(E(V, S_X), x) = 2$.

In more detail, if $x_i^1 \notin S_X$, then $(f_i, f_i'), (f_i, f_{i+1}') \in V[S]$, whose label for x_i^1 is $[1, -1]$. And when $|S_X \cap \mathcal{X}_1| < |\mathcal{X}_1| - 1$, there exists at least two other models in $V[S]$ that label x_i^1 with $[-1, -1]$.

Hence, since T never abstains on $x \in \mathcal{X}_1$, $|\mathcal{X}_1|-1$ labels will be given, at which point the disagreement region is still non-empty. Then, the algorithm either queries the $\tilde{x}$ or the remaining element in $\mathcal{X}_1$ depending on the tie-breaker, both of which identifies $\tilde{h}$.

□

E.4 Comparing with EPI-CAL

We examine the sample complexity when the order of data points is not controlled by the learner, who is nevertheless learning using a "mellow" AL algorithm, EPI-CAL. Our finding is that: strategic labeling can lead to a large sample complexity for this setting as well.

In the infinite-support case, even if the data stream is made up of i.i.d samples, EPI-CAL can incur large sample complexity, as the learner experiences an arbitrarily large "hold-up". This may be evidenced even in the simple threshold example in the lemma below.

Proposition E.9. *Fix some constant $\epsilon > 0$. Consider a PAC-learning task, where the learner seeks to learn a 1D threshold with at most $\epsilon-$risk with respect to continuous distribution $\mathcal{D}$. For any m i.i.d samples with m sufficiently large and $\mathcal{D}$ probability density bounded away from 0, there is a labeling strategy under which EPI-CAL queries $\Omega(\sqrt{m})$ labeled samples, with probability at least $1/2$.*

Proof. Let $h^* = 0$ for the 1D threshold hypothesis class $\mathcal{H} = \{\mathbb{1}(x \geq \theta) : \theta \in [0, 1]\}$.

Let $\mathcal{D}$ be some continuous distribution with $\text{supp}(\mathcal{D}) = [0, 1]$. Let $X_1, .., X_m$ denote the m i.i.d samples from $\mathcal{D}$.

By assumption, the pdf of $\mathcal{D}$ is bounded away from zero: $\Pr(x) \geq \kappa$, $\forall x \in \text{supp}(\mathcal{D})$ for some constant κ.

Then, $\Pr_{x\sim\mathcal{D}}(x \in (\epsilon, 1]) = \beta \geq (1 - \epsilon)\kappa = \Omega(1)$.

Under $m \geq 6$, consider some β_0 with $\beta_0 \leq \frac{\ln \frac{4}{3}}{2m}$. Since the CDF is continuous, there exists r such that $\Pr_{x\sim\mathcal{D}}(x \leq r) < \beta_0$, which is such that:

$$\Pr(\forall i \in [m], x_i \notin [0, r]) \geq (1 - \beta_0)^m \geq \exp(-2m\beta_0) \geq \frac{3}{4}$$

using that $1 - x \geq \exp(-2x)$ when $x \in [0, 1/2]$.

Define $\hat{r} = \min(r, \epsilon)$, which also satisfies the condition above since $[0, \hat{r}] \subseteq [0, r]$.

Now, we proceed to defining the labeling strategy:

1. Let $M = \sqrt{m}$. Using the continuity of $\Pr_{x\sim\mathcal{D}}(x < r)$ in r, we can find $1 = r_1 > ... > r_M > r_{M+1}$ with $r_{M+1} = \epsilon$, such that:

$$\Pr_{x\sim\mathcal{D}}(x \in [r_{i+1}, r_i]) = \frac{\beta}{M}$$

   Let $S_i = (r_{i+1}, r_i]$ for $i \in [M]$.

2. We make the observation that if EPI-CAL has only seen points from $S_{i_1}, ..., S_{i_j}$, then any point $x_k \in S_k$ with $k > \max(i_1, ..., i_j)$ will be accepted (bigger index means close to θ^*).

   This is because with labeled points only from $S_{i_1}, ..., S_{i_j}$, the resultant VS is a superset of $[0, r_{\max(i_1,...,i_j)+1}]$.

   And so, x_k is in the disagreement region, since $x_k \leq r_{\max(i_1,...,i_j)+1}$.

3. Now, we describe the sequential labeling strategy.

   a) Abstain on the region: $[\hat{r}, \epsilon]$.

   b) Label if $X_i \in [0, \hat{r})$. Note that labeling $[0, \hat{r})$ ensures that $\epsilon-$PAC learning is possible.

   For $X_i \in (\epsilon, 1]$, sequentially label as follows:

   i) Divide the m samples into M stages of M samples for $M = \sqrt{m}$.

   ii) At the ith stage, abstain if on the jth sample of this stage, $X_{ij} \notin S_i$.

   iii) The first time sample X_{ik} for $k \in [M]$ is such that $X_{ik} \in S_i$, label it and abstain for the rest of this stage.

   Using our previous point, we know that any point $X_{ik} \in S_i$ labeled will be accepted by EPI-CAL, since i is increasing.

   Intuitively, this labeling strategy slows down learning by only labeling points that shrink the VS by a little.

4. To analyze the total number of labeled points, let random variable Z_i denote whether a point is labeled at stage i. It is Bernoulli with probability:

$$p = \Pr(\exists j \in [M], X_{ij} \in [r_{i+1}, r_i]) = 1 - (1 - \beta/M)^M \geq 1 - \exp(-\beta) = \Omega(1)$$

   Using one-sided Chernoff's for Binomial random variables for M sufficiently large (i.e. for $M \geq \frac{8 \ln 4}{p}$) with p constant, we have:

$$\Pr(\sum_{i=1}^{M} Z_i \leq Mp/2) \leq \exp(-Mp/8) \leq 1/4$$

5. And so, using union bound, we have that:

$$\begin{aligned}
&\Pr(x_i \notin [0,\hat{r}], \forall i \in [m] \wedge \sum_{i=1}^{M} Z_i \geq Mp/2) \\
&\geq 1 - \Pr(\exists i \in [m], x_i \in [0,\hat{r}]) - \Pr(\sum_{i=1}^{M} Z_i < Mp/2) \\
&\geq 1 - 1/4 - 1/4 \\
&= 1/2
\end{aligned}$$

And so, the probability that all m samples are seen (i.e. the interaction does not terminate before all m), and that at least $Mp/2 = \Omega(\sqrt{m})$ samples are labeled and accepted by EPI-CAL occurs with probability at least $1/2$.

$\square$

Remark E.10. *We remark that:*

- *Consider when there is no labeler abstention. Let $Z'_i = \mathbb{1}(x_i \leq \min_{j\in[i-1]} x_j)$. Then we see that the expected sample complexity is:*

$$\mathbb{E}[\sum_{i=1}^{m} Z'_i] = \sum_{i=1}^{m} 1/i = O(\log m)$$

  Thus, we see that this is yet another setting, where labeler abstention can significantly increase the sample complexity.

- *From the Erdős–Szekeres theorem, the $\Theta(\sqrt{m})$ result is tight in expectation.*

F ADDITIONAL MATERIAL ON SECTION 4

In this section, we examine a few ways in which the labeler (e.g. a human worker) may be imperfect in both labeling and strategy, and extend our guarantees to such settings. We elaborate on the content covered in Section 4.

Note that in this paper, we make inroads into understanding the minimax strategies of the learning game. Analyzing minimax strategies is the canonical way of characterizing games, studying how players (e.g. a data provider company) may play rationally in the learning game. However, it has been recognized that players with bounded rationality (e.g. a human worker) may play behavioral strategies that are not minimax-optimal (Brown & Rosenthal, 1990). And so, we consider allow for the labeler labeling in a way that is sub-optimal.

F.1 RELAXED LEARNING GOAL

In the previous section, it is assumed that the learner is interested in exact learning some h^*. One may consider the relaxed goal of PAC learning some $\hat{h}$ such that $\Pr_{x\sim\mathcal{D}}(\hat{h}(x) \neq h^*(x)) \leq \epsilon$ w.p. greater than $1-\delta$, for some distribution $\mathcal{D}$ supported on $\mathcal{X}$.

Reduction: Then, following the standard realizable, PAC learning (with VC class) reduction (Vapnik, 1999), one may reduce the PAC setting to the exact learning by sampling $m = O(\frac{VCD}{\epsilon}(\ln\frac{1}{\epsilon} + \ln\frac{1}{\delta}))$ i.i.d samples from $\mathcal{D}$.

More precisely, let this random subset be $X^m \subseteq \mathcal{X}$. X^m partitions $\mathcal{H}$ into clusters of equivalent hypotheses. If we let the projection of $\mathcal{H}$ on X^m be $\mathcal{H}_{|X^m} = \left\{h(X^m) : h \in \mathcal{H}\right\}$, then a cluster $C(y)$ of equivalent hypotheses is defined $C(y) = \left\{h(X^m) = y : y \in \mathcal{H}_{|X^m}, h \in \mathcal{H}\right\}$.

The reduction guarantees that, with probability better than $1-\delta$ over the samples X^m, identification of h^*'s cluster $C(h^*(X^m))$ is sufficient for $\epsilon-$PAC learning. X^m is such that w.h.p $diam(C(h^*(X^m)) \leq \epsilon$, where diameter of a set H is defined as $diam(H) = \max_{h,h'\in H}\Pr_{x\sim D}(h(x) \neq h'(x))$. With this, picking any one model $\hat{h} \in C(h^*(X^m))$ satisfies $\Pr_{x\sim\mathcal{D}}(\hat{h}(x) \neq h^*(x)) \leq \epsilon$, and PAC learning thus reduces to identifying cluster $C(h^*(X^m))$.

F.1.1 APPROXIMATE IDENTIFIABILITY GAME

Using this reduction, we may analyze the query complexity of PAC learning as an exact learning game, where the learner chooses the data pool to be X^m (in place of $\mathcal{X}$). The goal is now only approximate identifiability, and identifying the cluster h^* belongs to, $C(h^*(X^m))$.

We demonstrate how our E-VS definition can be extended to develop a near-optimal algorithm under this approximate identifiable game. Our first observation is that the original E-VS, defined over $\mathcal{H}$ and X^m will no longer suffice:

$$E(V, S_X) = \left\{h \in V : \forall h' \in V \setminus \{h\} : h'(X^m \setminus S_X) \neq h(X^m \setminus S_X)\right\}$$

The issue is premature elimination. Consider some $h \in \mathcal{H}$ such that $|C(h(X^m))| \geq 2$ with $h' \in C(h(X^m)), h' \neq h$. Then, $h(X^m) = h'(X^m) \Rightarrow \exists h' \in \mathcal{H}, h'(X^m \setminus \emptyset) = h(X^m \setminus \emptyset)$, which results in the elimination of the entire $C(h(X^m))$ cluster at the very start. $E(\mathcal{H}, \emptyset)$ will not contain any clusters with cardinality more than one.

To handle this, we define a modification of the E-VS, X^m-E-VS, with relaxed elimination condition. This is a coarser E-VS, and so, we observe that we should only consider non-identifiability with respect to hypotheses from other clusters:

$$E^{X^m}(V, S_X) = \left\{h \in V : \forall h' \in V \setminus \left\{\bar{h} : \bar{h}(X^m) = h(X^m), \bar{h} \in V\right\} : h'(X^m \setminus S_X) \neq h(X^m \setminus S_X)\right\}$$

The added constraint of $V \setminus \left\{\bar{h} : \bar{h}(X^m) = h(X^m), \bar{h} \in V\right\}$ means that two h, h' within the same cluster do not render each other un-identifiable. And so, we only consider h''s from another cluster (that differs on X^m) that can render h (h's cluster) un-identifiable.

Remark F.1. *Through this we see that either an entire cluster is in the X^m-E-VS or it is not.*

Under the new X^m−E-VS definition, we may prove that the X^m−E-VS bisection algorithm similarly attains near-optimal guarantees. One may follow the same proof structure as in Lemma D.11 and Theorem D.13 to show both results also hold under X^m-E-VS. Thus, it suffices to prove the following two lemmas, which are used in the proofs of Lemma D.11 and Theorem D.13.

Lemma F.2. *For any $x \in \mathcal{X} \setminus S_X$ and $y \in \{-1, 1\}$,*

$$E^{X^m}(V[(x,y)], S_X \cup \{x\}) = E^{X^m}(V, S_X)[(x,y)]$$

Proof. The proof is identical to the one for the fine-grain E-VS:

$$\begin{aligned}
& h \in E^{X^m}(V[(x,y)], S_X \cup \{x\}) \\
\iff & h \in V[(x,y)] \wedge \forall h' \in V[(x,y)] \,.\, h'(X^m) \neq h(X^m) \to h'(X^m \setminus (S_X \cup \{x\})) \neq h(X^m \setminus (S_X \cup \{x\})) \\
\iff & h \in V \wedge h(x) = y \wedge \forall h' \in V[(x,y)] \,.\, h'(X^m) \neq h(X^m) \to h'(X^m \setminus (S_X \cup \{x\})) \neq h(X^m \setminus (S_X \cup \{x\})) \\
\iff & h \in V \wedge h(x) = y \wedge \forall h' \in V \,.\, h'(X^m) \neq h(X^m) \to h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X) \\
\iff & h(x) = y \wedge h \in E^{X^m}(V, S_X) \\
\iff & h \in E^{X^m}(V, S_X)[(x,y)]
\end{aligned}$$

□

Lemma F.3. *For any $V \subset \mathcal{H}$ and $S_X \subset \mathcal{X}$,*

$$E^{X^m}(V, S_X \cup \{x\}) \subseteq E^{X^m}(V, S_X)$$

Proof. It suffices to prove that $h \in E^{X^m}(V, S_X \cup \{x\}) \Rightarrow h \in E^{X^m}(V, S_X)$.

To see this, let $h \in E^{X^m}(V, S_X \cup \{x\})$. Then if h is such that:

$$\begin{aligned}
& \forall h' \in V, h'(X^m) \neq h(X^m), h((\mathcal{X} \setminus S_X) \setminus \{x\})) \neq h'((\mathcal{X} \setminus S_X) \setminus \{x\})) \\
& \Rightarrow \forall h' \in V, h'(X^m) \neq h(X^m), h(\mathcal{X} \setminus S_X) \neq h'(\mathcal{X} \setminus S_X) \\
& \Rightarrow h \in E(V, S_X)
\end{aligned}$$

□

Guarantee from learning from labeler with h' that approximates h^*: Suppose the labeler labels with h' and $\Pr(h'(x) \neq h^*(x)) \leq \epsilon/2$. One may consider the approximate identifiability learning game with precision $\epsilon/2$. Approximately-identifying some $\hat{h} \in C(h'(X^m))$ will be such that $\Pr(\hat{h}(x) \neq h'(x)) \leq \epsilon/2$. From this, we can conclude that:

$$\begin{aligned}
\Pr(\hat{h}(x) \neq h^*(x)) &= \Pr(\hat{h}(x) = h'(x) \wedge h'(x) \neq h^*(x)) + \Pr(\hat{h}(x) \neq h'(x) \wedge h'(x) = h^*(x)) \\
&\leq \Pr(h'(x) \neq h^*(x)) + \Pr(\hat{h}(x) \neq h'(x)) \\
&\leq \epsilon
\end{aligned}$$

F.1.2 Accessing the X^m−E-VS

After modifying the E-VS definition, the remaining issue is that we wish to find the maximal bisection point for coarse, X^m-E-VS. Here, we show that for the coarsened E-VS, the membership check implemented in Algorithm 3 (with the pool being X^m) is still sound. That is, we still have an oracle-efficient way of accessing the coarser X^m-E-VS, and can can implicitly track clusters through calls to the C-ERM oracle.

Proposition F.4. *$h \notin E_{X^m}(V, S_X)$ iff $\hat{h}(X^m) \neq h(X^m)$, where $\hat{h}$ is the minimizer of the C-ERM output below:*

$$\begin{aligned}
\hat{h} = \operatorname*{arg\,min}_{h' \in \mathcal{H}} & \sum_{x' \in S^\perp} \mathbb{1}\left\{h'(x') = h(x')\right\} \\
\text{s.t } & h'(x) = h(x), \forall x \in X^m \setminus S^\perp
\end{aligned} \tag{10}$$

Proof.

$$\begin{aligned}
\neg(h \in E_{X^m}(V, S_X)) &\Leftrightarrow \neg(\forall h' \in V \setminus \left\{\bar{h} : \bar{h}(X^m) = h(X^m), \bar{h} \in V\right\} \centerdot h'(X^m \setminus S_X) \neq h(X^m \setminus S_X)) \\
&\Leftrightarrow \exists h' \in V \setminus \left\{\bar{h} : \bar{h}(X^m) = h(X^m), \bar{h} \in V\right\} \centerdot h'(X^m \setminus S_X) = h(X^m \setminus S_X) \\
&\Leftrightarrow \exists h' \in V \centerdot h'(X^m) \neq h(X^m) \centerdot h'(X^m \setminus S_X) = h(X^m \setminus S_X) \\
&\Leftrightarrow \exists h' \centerdot h'(S^X \setminus S^\perp) = h(S^X \setminus S^\perp) \centerdot h'(X^m) \neq h(X^m) \centerdot h'(X^m \setminus S_X) = h(X^m \setminus S_X) \\
&\Leftrightarrow \exists h' \centerdot h'(S^X \setminus S^\perp) = h(S^X \setminus S^\perp) \centerdot h'(S^\perp) \neq h(S^\perp) \centerdot h'(X^m \setminus S_X) = h(X^m \setminus S_X) \\
&\Leftrightarrow \exists h' \centerdot h'(S^\perp) \neq h(S^\perp) \centerdot h'(X^m \setminus S^\perp) = h(X^m \setminus S^\perp) \\
&\Leftrightarrow \exists h' \centerdot \sum_{x' \in S^\perp} \mathbb{1}\left\{h'(x') = h(x')\right\} < |S^\perp| \centerdot h'(X^m \setminus S^\perp) = h(X^m \setminus S^\perp) \\
&\Leftrightarrow \hat{h}(X^m) \neq h(X^m) \centerdot \hat{h}(X^m \setminus S^\perp) = h(X^m \setminus S^\perp)
\end{aligned}$$

□

F.2 NOISED LABELING

It may be reasonable that in some cases, a labeler can make mistakes (even when they have tried their best) due to differing opinion and/or human error. For example, for medical diagnoses, doctors may hold differing opinions for the same case. This can be naturally modeled by the noised learning setting, as in (Castro & Nowak, 2008): querying example x returns $h^*(x)$ with known probability $1 - \delta(x)$, and $-h^*(x)$ with noise rate $\delta(x)$.

In this setup, we may use the common approach of repeatedly query a datum to estimate its label w.h.p. (e.g. as (Yan et al., 2016)). This approach reduces noised-label exact learning to cost-sensitive exact learning, where for each x there is some known query cost $c(x)$ — associated with determining $h^*(x)$ with high probability. With this, we may apply the results from Subsection D.2 to see that E-VS bisection algorithm will have near-optimal guarantees in this setting with example-dependent costs.

F.3 MYOPIC LABELING

Some labelers may want to enlarge the query complexity, but myopically may not have a near-optimal identifiable strategy. Instead, the labeler may have only a heuristic, which is only h^*-labeling, and can be non-identifiable. Non-identifiability is something neither parties want: the learner wants to learn h^*, and the labeler wants to be paid, which can only happen if h^* is learned.

In this light, we believe that the E-VS game representation is not only useful for the learner, but also for a labeler to reason about the game's state. For the labeler, there is an oracle-efficient way through which identifiability can be checked without enumerating the entire E-VS: simply apply the membership check on h^* as in Line 3 of Algorithm 3.

So even if the labeler is using some sub-optimal heuristic that may lead to non-identifiability of h^*, the labeler can prevent the next label from leading to non-identifiability by performing a membership check with a single C-ERM call. We add that only verifying that h^* is in E-VS, need not require enumerating all of the E-VS, and is thus tractable provided access to a C-ERM oracle.

G Proofs for Section 5

G.1 Lemmas used

Lemma G.1. *For all $V = \times_{i=1}^n V_i$ and S_X,*

$$E(V, S_X) = \times_{i=1}^n E(V_i, S_X^i)$$

Proof. We show both that:

1. For $V = \times_{i=1}^n V_i$, $\times_{i=1}^n E(V_i, S_X^i) \subseteq E(V, S_X)$:

   It suffices to show that if $h_i \in E(V_i, S_X^i)$ for $i \in [n]$, then $h = (h_1, .., h_n) \in E(V, S_X)$ for $V = \times_{i=1}^n V_i$.

   Firstly, since $h_i \in V_i$ and $V = \times_{i\in[n]} V_i$, we have that $h \in V$.

   Now suppose there is some $h' \in V$ such that $h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X)$; we would like to show that $h' = h$ – proving this would conclude that $h \in E(V, S_X)$.

   Indeed, consider any i; we have $h'_i((\mathcal{X}\setminus S_X)_i) = h_i((\mathcal{X}\setminus S_X)_i)$; equivalently, $h'_i(\mathcal{X}_i \setminus S_X^i) = h_i(\mathcal{X}_i \setminus S_X^i)$.

   As $h_i \in E(V_i, S_X^i)$ and $h'_i \in V_i$, we have that $h'_i = h_i$. Therefore h' and h are equal in all components, and $h' = h$.

2. For $V = \times_{i=1}^n V_i$, $E(V, S_X) \subseteq \times_{i=1}^n E(V_i, S_X^i)$:

   Consider any $h \in E(V, S_X)$; we would like to show that for any i, $h_i \in E(V_i, S_X^i)$.

   Suppose not, then there exists i, $h' \in V_i$ and $h' \neq h_i$ such that $h'(\mathcal{X}_i \setminus S_X^i) = h_i(\mathcal{X}_i \setminus S_X^i)$. This implies that $h'((\mathcal{X} \setminus S_X)_i) = h_i((\mathcal{X} \setminus S_X)_i)$, therefore, consider

   $$\tilde{h} = (h_1, \ldots, h_{i-1}, h', h_{i+1}, \ldots, h_n)$$

   We have that $\tilde{h} \in V$, $\tilde{h} \neq h$, and $\tilde{h}$ agrees with h on $\mathcal{X} \setminus S_X$, which contradicts the assumption that $h \in E(V, S_X)$.

$\square$

Lemma G.2. *For any data point (x_1, y_1) for $x_1 \notin S_X$ and $y_1 \in \{+1, -1, \perp\}$:*

$$CC(V[(x_1, y_1)], S_X \cup \{x_1\}) \leq CC(V, S_X)$$

Proof. We prove this by induction on $|S_X|$.

Base Case:

The base case is when $|S_X| = |\mathcal{X}| - 1$. Here $S_X \cup \{x_1\} = \mathcal{X}$. We have two subcases:

- $E(V[(x_1, y_1)], S_X \cup \{x_1\}) = \emptyset$.

  In this case, the inequality is satisfied.

- $|E(V[(x_1, y_1)], S_X \cup \{x_1\})| = 1$.

  We will show in general that $E(V[(x_1, y_1)], S_X \cup \{x_1\}) \subseteq E(V, S_X)$:

  i) If $y \neq \perp$, we know from Lemma D.3 that $E(V[(x_1, y_1)], S_X \cup \{x_1\}) = E(V, S_X)[(x_1, y_1)] \subseteq E(V, S_X)$.

  ii) If $y = \perp$, then $E(V[(x_1, y_1)], S_X \cup \{x_1\}) = E(V, S_X \cup \{x_1\}) \subseteq E(V, S_X)$.

  And so, $|E(V, S_X)| \geq 1 \Rightarrow CC(V, S_X) \geq 0 = CC(V[(x_1, y_1)], S_X \cup \{x_1\})$.

Induction Step:

For the inductive case, suppose the induction hypothesis holds for $|S_X| = |\mathcal{X}| - 1, .., j+1$. Consider some S_X with $|S_X| = j$.

We have three subcases:

- $E(V[(x_1, y_1)], S_X \cup \{x_1\}) = \emptyset$

  In this case, the inequality is satisfied.

- $|E(V[(x_1, y_1)], S_X \cup \{x_1\})| = 1$

  As shown before, $E(V[(x_1, y_1)], S_X \cup \{x_1\}) \subseteq E(V, S_X)$.

  And so, we have that $|E(V, S_X)| \geq 1 \Rightarrow CC(V, S_X) \geq 0 = CC(V[(x_1, y_1)], S_X \cup \{x_1\})$.

- $|E(V[(x_1, y_1)], S_X \cup \{x_1\})| \geq 2$.

  Using similar logic as before, $|E(V[(x_1, y_1)], S_X \cup \{x_1\})| \geq 2 \Rightarrow |E(V, S_X)| \geq 2$.

  Define

  $$x' \in \arg\min_{x \in \mathcal{X} \setminus S_X} \max_y \mathbb{1}(y \neq \perp) + CC(V[(x', y)], S_X \cup \left\{x'\right\})$$

  With this definition,

  $$CC(V, S_X) = \max_y \mathbb{1}(y \neq \perp) + CC(V[(x', y)], S_X \cup \left\{x'\right\})$$

  If $x' = x_1$, then the result follows since $CC(V, S_X) \geq \mathbb{1}(y_1 \neq \perp) + CC(V[(x_1, y_1)], S_X \cup \{x_1\})$.

  If $x' \neq x_1$, then $x' \in \mathcal{X} \setminus S \cup \{x_1\}$, and we can write:

$$\begin{aligned}
CC(V[(x_1, y_1)], S_X \cup \{x_1\}) &\leq \max_y \mathbb{1}(y \neq \perp) + CC(V[(x_1, y_1), (x', y)], S_X \cup \left\{x_1, x'\right\}) \\
&\qquad \text{(as } |E(V[(x_1, y_1)], S_X \cup \{x_1\})| \geq 2 \text{ so we can unroll, and } x' \in \mathcal{X} \setminus S \cup \{x_1\}) \\
&\leq \max_y \mathbb{1}(y \neq \perp) + CC(V[(x', y)], S_X \cup \left\{x'\right\}) \\
&\qquad \text{(using induction hypothesis since } |S_X \cup \left\{x'\right\}| = j+1) \\
&= CC(V, S_X)
\end{aligned}$$

□

Lemma G.3. *For $y \neq \perp$, $x \in \mathcal{X} \setminus S_X$:*

$$CC(V[(x, y)], S_X) = CC(V[(x, y)], S_X \cup \{x\})$$

Proof. Firstly, we have that:

$$\begin{aligned}
E(V[(x, y)], S_X) &= \left\{h \in V[(x, y)] : \forall h' \in V[(x, y)] \setminus \{h\}, h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)\right\} \\
&= \left\{h \in V[(x, y)] : \forall h' \in V[(x, y)] \setminus \{h\}, h'(\mathcal{X} \setminus (S_X \cup \{x\}) \neq h(\mathcal{X} \setminus S_X \cup \{x\})\right\} \\
&= E(V[(x, y)], S_X \cup \{x\})
\end{aligned}$$

Hence the statement holds when $S_X = \mathcal{X} \setminus \{x\}$, or more generally, when $CC(V[(x, y)], S_X \cup \{x\})$ or $CC(V[(x, y)], S_X)$ is at its base case (one implies the other due to having the same E-VS).

Now, we will induct on the size of $|S_X|$, since the base case of $S_X = \mathcal{X} \setminus \{x\}$ is satisfied.

Base case: $|S_X| = |\mathcal{X}| - 1$.

If $E(V, S_X) = E(V, S_X \cup \{x\}) = \emptyset$, then $LHS = RHS = -\infty$;

If $|E(V, S_X)| = |E(V, S_X \cup \{x\})| = 1$, then $LHS = RHS = 0$.

Induction Step: Suppose the statement holds for when $|S_X| = |\mathcal{X}|, ..., j+1$. Let $|S_X| = j$.

We first handle the base cases:

If $E(V, S_X) = E(V, S_X \cup \{x\}) = \emptyset$, then $LHS = RHS = -\infty$;

If $|E(V, S_X)| = |E(V, S_X \cup \{x\})| = 1$, then $LHS = RHS = 0$.

Finally, it remains to consider when $|E(V, S_X)| = |E(V, S_X \cup \{x\})| \geq 2$. In this case,

$$CC(V, S_X) = \min_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + CC(V[(x', y')], S_X \cup \left\{x'\right\}).$$

Define $x^* \in \arg\min_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + CC(V[(x', y')], S_X \cup \left\{x'\right\})$.

We will show that $x^* \neq x$.

In fact, for any $x' \in \mathcal{X} \setminus S$, $x' \neq x^*$ (which exists because $\{x\} \subset \mathcal{X} \setminus S_X$) we have:

$$\begin{aligned}
&\max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + CC(V_x^y[(x, y')]], S_X \cup \{x\}) \\
&= \max(1 + CC(V_x^y, S_X \cup \{x\}), 1 + CC(\emptyset, S_X \cup \{x\}), CC(V_x^y, S_X \cup \{x\})) \\
&= 1 + CC(V_x^y, S_X \cup \{x\}) && \text{(maximized at when } y' = y) \\
&\geq \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + CC(V_x^y[(x', y')], S_X \cup \left\{x, x'\right\}) \\
& && \text{(using } 1 \geq \mathbb{1}(y \neq \perp) \text{ and Lemma G.2)} \\
&= \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + CC(V_x^y[(x', y')], S_X \cup \left\{x'\right\}) \\
& && \text{(using induction hypothesis since } |S_X \cup \left\{x'\right\}| = j+1)
\end{aligned}$$

And so,

$$\begin{aligned}
CC(V[(x, y)], S_X) &= \min_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + CC(V_x^y[(x', y')]], S_X \cup \left\{x'\right\}) \\
&= \min_{x' \in \mathcal{X} \setminus (S_X \cup \{x\})} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + CC(V_{x'}^{y'}[(x, y)]], S_X \cup \left\{x'\right\}) \\
& \qquad \text{(since we have just shown that } x^* \neq x) \\
&= \min_{x' \in \mathcal{X} \setminus (S_X \cup \{x\})} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + CC(V_{x'}^{y'}[(x, y)]], (S_X \cup \left\{x'\right\}) \cup \{x\}) \\
& \qquad \text{(using induction hypothesis since } |S_X \cup \left\{x'\right\}| = j+1) \\
&= \min_{x' \in \mathcal{X} \setminus (S_X \cup \{x\})} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + CC(V_x^y[(x', y')]], (S_X \cup \{x\}) \cup \left\{x'\right\}) \\
& \qquad \text{(rearranging)} \\
&= CC(V[(x, y)], S_X \cup \{x\})
\end{aligned}$$

□

G.2 Upper Bound

G.2.1 Negative Results

Upper Bound when there is Identifiability:

We first observe that without assumptions on the structure of V, there exists a setting, in which the upper bound does not hold.

Proposition G.4. *There exists a non-Cartesian product version space $V \subseteq \mathcal{H}$ and query response $S \subseteq (\mathcal{X} \times \mathcal{Y})^*$ such that $CC(V_i, S_X^i) \geq 0$ for all i, but:*

$$CC(V, S_X) \geq \sum_{i=1}^{n} CC(V_i, S_X^i) + n - 1$$

Proof. We will construct a V and S such that $CC(V, S_X) \geq n-1$, but $CC(V_i, S_X^i) = 0$.

Hypothesis Class: Define thresholds functions $f_1 = \mathbb{1}(x \geq 1/4), f_2 = \mathbb{1}(x \geq 1/2), f_3 = \mathbb{1}(x \geq 3/4)$ for $x \in [0, 1]$.

Define $\mathcal{H}'$ as:

$$\mathcal{H}' = \{(f_1, f_2, ..., f_2), (f_2, f_1, ..., f_2), ..., (f_2, f_2, , ..., f_1)\}$$

where the jth model has its jth task model as f_1 instead of f_2.

Define the non-Cartesian product hypothesis class as:

$$\mathcal{H} = \mathcal{H}' \cup \{(f_2, f_2, ..., f_2), (f_3, f_3, ..., f_3)\}$$

We have that $\mathcal{H}_i = \{f_1, f_2, f_3\}$.

Data: Let $\mathcal{X}_1 = \{x_{i1}\}_{i=1}^n$ and $\mathcal{X}_2 = \{x_{i2}\}_{i=1}^n$, where $x_{i1} = 1/3e_i$ and $x_{i2} = 2/3e_i$. Let $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$.

Query Responses: Suppose $S = \{(x_{i2}, [\perp, ..., \perp]) : i \in [n]\}$.

This means that $S_X = \{x_{i2} : i \in [n]\}$, and that $S_X^i = \{2/3\}$, since the only $x \in \mathcal{X}$ such that $x_i = 2/3$ is x_{i2} and $x_{i2} \in S_X$.

Define $V = \mathcal{H}[S] = \mathcal{H}$. And so, $V_i = \{f_1, f_2, f_3\}$.

We have that $E(V_i, S_X^i) = \{f_1\}$, and so, $CC(V_i, S_X^i) = 0$.

Now, it remains to show that $E(V, S_X) = \mathcal{H}'$.

Firstly, since $V = \mathcal{H}[S] = \mathcal{H}$, we examine each model in $\mathcal{H}$.

The model $(f_2, f_2, ..., f_2)$ and $(f_3, f_3, ..., f_3)$'s predictions on x_{i1} (for any i) are both $(-1, -1, \ldots, -1)$. Thus, they have the same predictions on $\{x_{i1}\}_{i\in[n]} = \mathcal{X} \setminus S_X$, and so, $(f_2, f_2, ..., f_2), (f_3, f_3, ..., f_3) \notin E(V, S_X)$.

With this, we see that $E(V, S_X) = \mathcal{H}'$, because for the ith element of $\mathcal{H}'$, it disagrees with every other element on x_{i1}.

Finally, we will show that $CC(V, S_X) \geq n - 1$.

Consider a labeling strategy that returns label $(-1, ..., -1)$ for any x_{i1} queried.

This strategy identifies some $h \in \mathcal{H}$, since each point in $\mathcal{X}_1$ that is queried removes one model from E-VS. And so, after $n - 1$ queries on points in $\mathcal{X}_1$, the E-VS has one hypothesis and the learning interaction finishes since the identification condition is met.

We note that any querying algorithm will require $n - 1$ labeled queries. Each binary labeled example removes only one model from the E-VS, thus $n - 1$ labels are required for identification under any querying algorithm. And so, we have that $CC(V, S_X) \geq n - 1$.

□

Upper Bound when there is no Identifiability:

Proposition G.5. *For non-Cartesian product hypothesis class V, there exists V, S such that $CC(V_i, S_X^i) = -\infty$ for some i, but $CC(V, S_X) \geq 1$.*

Proof. Consider $\mathcal{H} = \{(h_1, h_2), (h_3, h_4)\}$.

$\mathcal{X} = \{[x_1, 0], [0, x_2]\}$, where for $x_1, x_2 \neq 0$, $h_1(x_1) \neq h_3(x_1)$ and $h_2(x_2) \neq h_4(x_2)$. $h_1(0) = h_3(0)$ and $h_2(0) = h_4(0)$.

Consider query response $S = \{([x_1, 0], [\perp, \perp])\}$. $S_X = \{[x_1, 0]\}$, $S_X^1 = \{x_1\}, S_X^2 = \{0\}$.

$V = \mathcal{H}[S] = \mathcal{H}$. $V_1 = \{h_1, h_3\}$ and $V_2 = \{h_2, h_4\}$.

$E(V_1, \{x_1\}) = E(\{h_1, h_3\}, \{x_1\}) = \emptyset$. However, $E(V, \{[x_1, 0]\}) = \mathcal{H}$, since (h_1, h_2) and (h_3, h_4) differ on $[0, x_2]$.

And so, $1 = CC(V, S_X) > \sum_{i=1}^2 CC(V_i, S_X^i) = -\infty$, since $CC(V_1, S_X^1) = -\infty$. □

Remark G.6. *In conclusion, to show the upper bound, need to impose Cartesian product condition.*

Negative Example motivating the need to assume a particular label cost definition:

When the label cost is c_{one}, there are settings where $CC(V, S_X)$ can be much larger i.e. $CC(V, S_X) >> \sum_{i=1}^{n} CC(V_i, S_X^i)$.

Proposition G.7. *If the label cost is $c_{one}(y) = \mathbb{1}(\exists i, y_i \neq \perp)$, there exists V and S such that $CC(V_i, S_X^i) = 1$, but $CC(V, S_X) = |\mathcal{X}|$. This implies that:*

$$CC(V, S_X) > \sum_{i=1}^{n} CC(V_i, S_X^i)$$

Proof. Consider $V = \{h_1, h_2\} \times \{h_3, h_4\}$, where $h_1, h_2 \in V_1$ are thresholds functions $h_1 = \mathbb{1}(x \geq 0), h_2 = \mathbb{1}(x \geq 1)$ and $h_3, h_4 \in V_2$ are also thresholds $h_3 = \mathbb{1}(x \geq 0), h_4 = \mathbb{1}(x \geq 1)$.

$\mathcal{X} = \left\{[\frac{1}{m+1}, \frac{1}{m+1}], ..., [\frac{m}{m+1}, \frac{m}{m+1}]\right\}$, which means that $\mathcal{X}_1 = \mathcal{X}_2 = \left\{\frac{1}{m+1}, ..., \frac{m}{m+1}\right\}$.

We will show that:

$$CC(V, \emptyset) >> CC(V_1, \emptyset) + CC(V_2, \emptyset)$$

We first have that $CC(V_1, \emptyset), CC(V_2, \emptyset) = 1$, since only one labeled sample is needed to distinguish between h_1, h_2 and between h_3, h_4.

However, we have $CC(V, \emptyset) \geq m = |\mathcal{X}|$ with the following labeling strategy T:

1) As long as $|S_X| < m - 1$, for queried point $[\frac{i}{m+1}, \frac{i}{m+1}]$, return $(\perp, h_3(\frac{i}{m+1}))$.

2) Only when $|S_X| = m - 1$, for queried point $[\frac{j}{m+1}, \frac{j}{m+1}]$, return $(h_1(\frac{j}{m+1}), h_3(\frac{j}{m+1}))$.

We can first that this is an identifiable labeling strategy that identifies (h_1, h_3).

And, for any querying algorithm, h^* is only identified when $S_X = \mathcal{X}$.

Thus, $|\mathcal{X}|$ labeled samples need to be queried, making $CC(V, \emptyset) = |\mathcal{X}|$.

□

Remark G.8. *To prove the above bound, we need to assume the label cost to be: $\mathbb{1}(y \neq \perp) = \mathbb{1}(\forall i, y_i \neq \perp) = c_{all}(y)$.*

G.2.2 Positive Results

Change in Definition of the Game:

- To prove the upper bound, we have a changed definition in labeling payoff, which is now:

$$\mathbb{1}(y \neq \perp) := \mathbb{1}(\forall i, y_i \neq \perp)$$

- The earlier negative example motivates requiring the assumption that V is a Cartesian product.

Theorem G.9. *For all $V = \times_{i \in [n]} V_i$ and $S_X \subseteq \mathcal{X}$, under labeling cost $c_{all}(y) = \mathbb{1}(\forall i, y_i \neq \perp)$:*

$$CC(V, S_X) \leq \sum_{i=1}^{n} CC(V_i, S_X^i)$$

Proof. We prove this by induction on the size of S_X.

Base Case: When $S_X = \mathcal{X} \Rightarrow S_X^i = \mathcal{X}_i$. So for all i, $|E(V_i, S_X^i)| \leq 1$.

It suffices to check that $CC(V, S_X) = 0 \Rightarrow \forall i, CC(V_i, S_X^i) = 0$.

Indeed, if $CC(V, S_X) = 0$, then $|E(V, \mathcal{X})| = 1$. Denote by h the only element of $E(V, \mathcal{X})$.

We must have $V = \{h\}$, which in turn implies that for all i $V_i = \{h_i\}$. Therefore, for all i, $|E(V, \mathcal{X})| = \{h_i\} = 1$, which implies $\forall i, CC(V_i, S_X^i) = 0$.

Induction Step:

Suppose the following holds for $S_X \subset X$ for $|S_X| = |\mathcal{X}|, ..., j+1$. Now let $|S_X| = j$ (note that $S_X \subset \mathcal{X}$).

We will analyze the three cases:

- $\exists i, CC(V_i, S_X^i) = -\infty$
- $\forall i, CC(V_i, S_X^i) \geq 0$ and $\forall i, CC(V_i, S_X^i) = 0$
- $\forall i, CC(V_i, S_X^i) \geq 0$ and $\exists i, CC(V_i, S_X^i) \geq 1$.

1. **If there is at least one i such that $CC(V_i, S_X^i) = -\infty$.**

   It suffices to verify that $\exists i, E(V_i, S_X^i) = \emptyset \Rightarrow E(V, S_X) = \emptyset$.

   This follows immediately from that $E(V, S_X) = \times_{i=1}^n E(V_i, S_X^i)$ (Lemma G.1).

2. **For all i, $CC(V_i, S_X^i)$ is at its base case and $CC(V_i, S_X^i) = 0$.**

   That is, we have $\forall i, |E(V_i, S_X^i)| = 1$.

   From Lemma G.1, we have that $E(V, S_X) = \times_{i=1}^n E(V_i, S_X^i)$, which means that $|E(V, S_X)| = 1$. And so, $CC(V, S_X) = 0 = \sum_{i=1}^n CC(V_i, S_X^i)$.

3. **Exists i such that $CC(V_1, S_X^1) \geq 1$, and $CC(V_i, S_X^i) \geq 0$ for all i.**

   Without loss of generality, $i = 1$.

   Note that if $|E(V, S_X)| \leq 1$, then $CC(V, S_X) \leq 0 \leq \sum_{=1}^n CC(V_i, S_X^i)$.

   And so, throughout the rest of the proof, we focus on the case that $|E(V, S_X)| \geq 2$. Also, recall that since $CC(V_1, S_X^1) \geq 1$ implies that $E(V_1, S_X^1) \geq 2$.

   Define
   $$x_1^* = \arg\min_{x \in \mathcal{X}_1 \setminus S_X^1} \max_{y \in \mathcal{Y}} \mathbb{1}(y \neq \perp) + CC(V_1[(x_1^*, y)], S_X^1 \cup \{x_1^*\})$$

   We may express:
   $$CC(V_1, S_X^1) = \max_{y \in \mathcal{Y}} \mathbb{1}(y \neq \perp) + CC(V_1[(x_1^*, y)], S_X^1 \cup \{x_1^*\})$$

   And since $x_1^* \in \mathcal{X}_1 \setminus S_X^1$, the set $X_1^* = \{x' \in \mathcal{X} \setminus S_X : x_1' = x_1^*\}$ is non-empty.

   Denote $L_X = \{x : (x, y) \in L\}$. Consider the following procedure:

   repeat
   $L = \emptyset$
   Query some $x \in X_1^*$
   Labeler returns y:
   $$y = \arg\max_y \mathbb{1}(y \neq \perp) + CC(V[L \cup \{(x, y)\}], S_X \cup L_X \cup \{x\})$$
   $X_1^* \leftarrow X_1^* \setminus \{x\}$
   $L \leftarrow L \cup \{(x, y)\}$
   until $y_1 \neq \perp$ **or** $X_1^* = \emptyset$

   Denote by $\hat{y}_1$ the value of y_1 at the end of the procedure, let $|L| = m$ and, in order, interaction history L is such that $L = \{(x^1, y^1), ..., (x^m, y^m)\}$. Let $L^i = \{(x_i, y_i) : (x, y) \in L, y_i \neq \perp\}$ index the binary labeled data for the ith task.

$$
\begin{aligned}
CC(V, S_X) &\leq \mathbb{1}(y^1 \neq \perp) + CC(V[(x^1, y^1)], S_X \cup \left\{x^1\right\}) \quad \text{(since } x^1 \in X_1^* \subseteq \mathcal{X} \setminus S_X\text{)} \\
&= CC(V[(x^1, y^1)], S_X \cup \left\{x^1\right\}) \quad \text{(since } y_1^1 = \perp\text{)} \\
&\leq ... \\
&\text{(unrolling according to } L\text{, which is possible as } CC(V, S_X) \geq 1 \Rightarrow CC(V[L], S_X \cup L_X) \geq 1\text{)} \\
&\leq \mathbb{1}(y^m \neq \perp) + CC(V[L], S_X \cup L_X) \\
&\leq \mathbb{1}(\hat{y}_1 \neq \perp) + CC(V[L], S_X \cup L_X) \\
&\qquad (\mathbb{1}(\forall i, y_i^m \neq \perp) \leq \mathbb{1}(\hat{y}_1 \neq \perp) \text{ since } y_1^m = \hat{y}_1) \\
&= \mathbb{1}(\hat{y}_1 \neq \perp) + CC(\times_{i \in [n]} V_i[L^i], S_X \cup L_X) \quad (V \text{ is a Cartesian product}) \\
&\leq \mathbb{1}(\hat{y}_1 \neq \perp) + \sum_{i=1}^{n} CC(V_i[L^i], (S_X \cup L_X)^i) \\
&\qquad \text{(using induction hypothesis as } |L_X| \geq 1\text{)} \\
&= \mathbb{1}(\hat{y}_1 \neq \perp) + CC(V_1[(x_1^*, \hat{y}_1)], S_X^1 \cup \{x_1^*\}) + \sum_{i=2}^{n} CC(V_i[L^i], (S_X \cup L_X)^i) \\
&\qquad (\diamond) \\
&\leq CC(V_1, S_X^1) + \sum_{i=2}^{n} CC(V_i[L^i], (S_X \cup L_X)^i) \quad \text{(by definition of } x_1^*\text{)} \\
&\leq CC(V_1, S_X^1) + \sum_{i=2}^{n} CC(V_i, S_X^i) \quad (\diamond\diamond)
\end{aligned}
$$

($\diamond$): For the fourth step, there are two cases:

- If upon exit, $X_1^* = \emptyset$:
  Then using the definition of S_X^1, since $\nexists x \in \mathcal{X} \setminus (S_X \cup L_X)$ with $x_1 = x_1^*$, we have that $(S_X \cup L_X)^1 = S_X^1 \cup \{x_1^*\}$.
  Therefore, $CC(V_1[L^1], (S_X \cup L_X)^1) = CC(V_1[(x_1^*, \hat{y}_1)], S_X^1 \cup \{x_1^*\})$.
- Otherwise, upon exit, $X_1^* \neq \emptyset$. Then, we must have that $\hat{y} \neq \perp$:
  So $\exists x \in \mathcal{X} \setminus (S_X \cup L_X)$ with $x_i = x_i^*$.
  Therefore, $(S_X \cup L_X)^1 = S_X^1$, hence $CC(V_1[L^1], (S_X \cup L_X)^1) = CC(V_1[(x_1^*, \hat{y}_1)], S_X^1)$.
  From Lemma G.3, we have that $CC(V_1[(x_1^*, \hat{y}_1)], S_X^1) = CC(V_1[(x_1^*, \hat{y}_1)], S_X^1 \cup \{x_1^*\})$.

($\diamond\diamond$): For the last step, consider each task i for $i \in \{2, \ldots, n\}$:

Define:

- $L_X^{i1} = \left\{x' : \exists (x, y) \in L, x_i = x', y_i \neq \perp \wedge x' \in (S_X \cup L_X)^i\right\}$
- $L_X^{i2} = \left\{x' : \forall (x, y) \in L, x_i = x', y_i = \perp \wedge x' \in (S_X \cup L_X)^i\right\}$
- $L_X^{i3} = \left\{x' : \exists (x, y) \in L, x_i = x', y_i \neq \perp \wedge x' \notin (S_X \cup L_X)^i\right\}$
- $L_X^{i4} = \left\{x' : \forall (x, y) \in L, x_i = x', y_i = \perp \wedge x' \notin (S_X \cup L_X)^i\right\}$

With these definitions, we have $(S_X \cup L_X)^i = S_X^i \cup L_X^{i1} \cup L_X^{i2}$. The binary labeled examples comprise of $L_X^i = L_X^{i1} \cup L_X^{i3}$.

We have that:

$$\begin{aligned}
CC(V_i[L^i], (S_X \cup L_X)^i) &= CC(V_i[L^i], S_X^i \cup L_X^{i1} \cup L_X^{i2}) \\
&= CC(V_i[L^i], S_X^i \cup L_X^{i1} \cup L_X^{i2} \cup L_X^{i3}) \\
&\qquad \text{(using Lemma G.3 on } L_X^{i3}) \\
&= CC(V_i[L^i \cup \left\{(x, \perp) : x \in L_X^{i2}\right\}], S_X^i \cup L_X^{i1} \cup L_X^{i2} \cup L_X^{i3}) \\
&\leq CC(V_i, S_X^i) \\
&\qquad \text{(iteratively applying Lemma G.2 on } L_X^{i1} \cup L_X^{i2} \cup L_X^{i3})
\end{aligned}$$

□

G.3 Lower Bound

Label Cost Function: From this point onwards, we assume that the label cost is (the more generous) c_{one}.

G.3.1 Negative Results

Lower Bound when there is Identifiability:

The following example leverages the fact that structure in the multi-task hypothesis class constrains the target hypotheses across all n tasks. And so, abstentions can lead to the multi-task setting requiring fewer samples than even the single-task setting with the highest sample complexity.

Proposition G.10. *There exists a non-Cartesian product version space V and query response S such that $CC(V_i, S_X^i) \geq 0$ for all i, but:*

$$CC(V, S_X) < \max_{i \in [n]} CC(V_i, S_X^i)$$

Proof. **Hypothesis Class:** Define all zero-classifier, $h_0(x) = 0$ for all x. Let $h_i = \mathbb{1}(x \in [i, i+1))$ for $i \in [n]$ be the ith interval.

Let g_1, g_2, g_3 be three distinct threshold functions, $g_1 = \mathbb{1}(x \geq 1/4), g_2 = \mathbb{1}(x \geq 1/2), g_3 = \mathbb{1}(x \geq 3/4)$ for $x \in [0, 1]$.

Set $\mathcal{H}$ to be $\left\{(h_0, g_1), (h_0, g_2), \left\{(h_i, g_3)\right\}_{i=1}^n\right\}$.

Data: Define $\mathcal{X} = \left\{[x_{11}, 0], ..., [x_{1n}, 0], [0, x_{21}], [0, x_{22}]\right\}$ where $x_{1i} = i + 1/2$ for $i \in [n]$ and $x_{21} = 1/3$, $x_{22} = 2/3$. By construction, $g_1(x_{21}) \neq g_2(x_{21})$ and $g_2(x_{22}) \neq g_3(x_{22})$.

Define $S = \left\{([0, x_{21}], [\perp, \perp])\right\}$. $S_X = \left\{[0, x_{21}]\right\}$, $S_X^1 = \{\}$, $S_X^2 = \{x_{21}\}$.

We have $V = \mathcal{H}[S] = \mathcal{H}$. $V_1 = \mathcal{H}_1 = \{h_0, h_1, h_2, h_3, ..., h_n\}$ and $V_2 = \mathcal{H}_2 = \{g_1, g_2, g_3\}$.

$g_1((\mathcal{X} \setminus S_X)_2) = g_2((\mathcal{X} \setminus S_X)_2) \Rightarrow (h_0, g_1), (h_0, g_2) \notin E(V, S_X)$.

We have $E(V, S_X) = \left\{(h_i, g_3)\right\}_{i=1}^n$, because for any $i \neq j$, (h_i, g_3) and (h_j, g_3) differ on $[x_{1j}, 0]$.

From this, we get that $CC(V, S_X) = n - 1$. Querying any point $[x_{1i}, 0]$ at any time removes only one model from the E-VS. Since the E-VS is of size n, $n - 1$ binary labeled examples are needed to reduce the E-VS size to at most 1.

On the other hand, we have that for $CC(V_1, S_X^1)$ with $|V_1| = n + 1$ and $S_X^1 = \emptyset$, $CC(V_1, S_X^1) = n > CC(V, S_X)$.

□

Lower Bound when there is no Identifiability even with Cartesian product assumption:

Proposition G.11. *There exists a Cartesian product version space V and query response S with $CC(V, S_X) < 0$ such that:*

$$CC(V, S_X) < \max_{i \in [n]} CC(V_i, S_X^i)$$

Proof. Let $\mathcal{H} = \{h_{11}, h_{12}\} \times \{h_{21}, h_{22}\}$, where $h_{11} = \mathbb{1}(x \geq 0), h_{12} = \mathbb{1}(x \geq 1)$ are intervals, and $h_{21} = \mathbb{1}(x \geq 0), h_{22} = \mathbb{1}(x \geq 1)$ are intervals.

$\mathcal{X} = \{[x_1, 0], [0, x_2]\}$ where $x_1 = 1/2$, $x_2 = 1/2$.

Labeling is: $S = \{([x_1, 0], [\perp, 1])\}$. $S_X = \{[x_1, 0]\}$, $S_X^1 = \{x_1\}$, $S_X^2 = \{0\}$.

So $V = \mathcal{H}[S] = \mathcal{H}$. $V_1 = \{h_{11}, h_{12}\}$ and $V_2 = \{h_{21}, h_{22}\}$.

Under S, we observe that $E(V, S_X) = \emptyset$, since (h_{11}, h) and (h_{12}, h) for $h \in V_2 = \{h_{21}, h_{22}\}$, predict the same on $\{[0, x_2]\} = \mathcal{X} \setminus S_X$. Hence, $CC(V, S_X) = -\infty$.

However, $CC(V_2, S_X^2) = CC(\{h_{21}, h_{22}\}, \{0\}) = 1 > CC(V, S_X)$.

$\square$

Remark G.12. *To prove the lower bound, need to impose both identifiability $CC(V, S_X) \geq 0$ *and* Cartesian product condition.*

G.3.2 Positive Results

Theorem G.13. *For all $V = \times_{i \in [n]} V_i$ and $S_X \subseteq \mathcal{X}$, if $CC(V, S_X) \geq 0$, then:*

$$CC(V, S_X) \geq \max_{i \in [n]} CC(V_i, S_X^i)$$

Proof. We prove this by induction on the size of S_X.

Base Case: When $S_X = \mathcal{X} \Rightarrow S_X^i = \mathcal{X}_i$, so for all i, $CC(V_i, S_X^i) \leq 0 \leq CC(V, S_X)$.

Induction Step: Suppose the following holds for $|S_X| = |\mathcal{X}|, ..., j + 1$.

Now let $|S_X| = j$. Note that this implies $S_X \subset X$.

First, consider the case when $CC(V, S_X) = 0$. We have that $|E(V, S_X)| = 1$. And so, using Lemma G.1, for all i, $|E(V_i, S_X^i)| = 1$. Thus, $CC(V_i, S_X^i) = 0$ for all i.

Now, we consider the case when $CC(V, S_X) \geq 1$.

Let $k = \arg\max_{i \in [n]} CC(V_i, S_X^i)$. It suffices to verify the statement when $CC(V_k, S_X^k) \geq 1$.

Since $\mathcal{X} \setminus S_X$ is non-empty due to $S_X \subset \mathcal{X}$, define:

$$x^{min} = \underset{x \in \mathcal{X} \setminus S_X}{\arg\min} \max_{y' \in \mathcal{Y}} \mathbb{1}(y' \neq \perp) + CC(V_x^{y'}, S_X \cup \{x\})$$

We have that $\mathcal{X}_k \setminus S_X^k = (\mathcal{X} \setminus S_X)_k = \{x' \in \mathcal{X}_k : \exists x \in \mathcal{X} \setminus S_X, x_k = x'\}$, and so $x_k^{min} \in \mathcal{X}_k \setminus S_X^k$ since $x^{min} \in \mathcal{X} \setminus S_X$.

Since $CC(V_k, S_X^k) \geq 1$, we know there exists $\tilde{y}_k$ such that:

$$CC(V_k, S_X^k) \leq \mathbb{1}(\tilde{y}_k \neq \perp) + CC(V_k[(x_k^{min}, \tilde{y}_k)], S_X^k \cup \{x_k^{min}\}).$$

Note in particular that $E(V_k[(x_k^{min}, \tilde{y}_k)], S_X^k \cup \{x_k^{min}\}) \neq \emptyset$, as otherwise $CC(V_k, S_X^k) \leq -\infty$ which would contradict our assumption that $CC(V_k, S_X^k) \geq 1$.

$$CC(V,S_X) = \max_{y'\in\mathcal{Y}} \mathbb{1}(y' \neq \perp) + CC(V^{y'}_{x^{min}}, S_X \cup \left\{x^{min}\right\})$$
$$\geq \mathbb{1}(y \neq \perp) + CC(\times_{i\in[n]}(V_i)^{y_i}_{x_i^{min}}, S_X \cup \left\{x^{min}\right\})$$
(setting $y' = y$ as constructed in Lemma G.14 and using that $V^{y}_{x^{min}} = \times_{i\in[n]}(V_i)^{y_i}_{x_i^{min}}$)
$$\geq \mathbb{1}(y \neq \perp) + \max_{i\in[n]} CC((V_i)^{y_i}_{x_i^{min}}, (S_X \cup \left\{x_i^{min}\right\})^i)$$
(using induction hypothesis since $x^{min} \notin S_X$, so $|S_X \cup \left\{x^{min}\right\}| = j+1$)
$$\geq \mathbb{1}(\tilde{y}_k \neq \perp) + CC((V_k)^{\tilde{y}_k}_{x_k^{min}}, (S_X \cup \left\{x^{min}\right\})^k)$$
($\mathbb{1}(y \neq \perp) \geq \mathbb{1}(y_k \neq \perp) = \mathbb{1}(\tilde{y}_k \neq \perp)$ as $y_k = \tilde{y}_k$ by construction)
$$\geq \mathbb{1}(\tilde{y}_k \neq \perp) + CC((V_k)^{\tilde{y}_k}_{x_k^{min}}, S_X^k \cup \left\{x_k^{min}\right\})$$
(note that $x_k^{min} \in (\mathcal{X} \setminus S_X)_k$, so $x_k^{min} \in \mathcal{X}_k \setminus S_X^k$ and $\diamond$)
$$\geq CC(V_k, S_X^k)$$

($\diamond$): Either we have $(S_X \cup \{x^{min}\})^k = S_X^k \cup \{x_k^{min}\}$ or $(S_X \cup \{x^{min}\})^k = S_X^k$. The former case yields equality and the statement holds.

For the latter case, we can use Lemma G.2 (for $\tilde{y}_k = \perp$) or Lemma G.3 (for $\tilde{y}_k \neq \perp$) to get that: $CC((V_k)^{\tilde{y}_k}_{x_k^{min}}, (S_X \cup \{x^{min}\})^k) = CC((V_k)^{\tilde{y}_k}_{x_k^{min}}, S_X^k) \geq CC((V_k)^{\tilde{y}_k}_{x_k^{min}}, S_X^k \cup \{x_k^{min}\})$.

□

Lemma G.14. *Suppose* $C(V,S_X) \geq 0$ *and* $x^{min} = \arg\min_{x\in\mathcal{X}\setminus S_X} \max_{y\in\mathcal{Y}} \mathbb{1}(y \neq \perp) + CC(V^y_x, S_X \cup \{x\})$. *If there* $\tilde{y}_k$ *such that* $CC(V_k, S_X^k) \leq \mathbb{1}(\tilde{y}_k \neq \perp) + CC(V_k[(x_k^{min}, \tilde{y}_k)], S_X^k \cup \{x_k^{min}\})$ *for* $CC(V_k, S_X^k) \geq 0$, *then there exists* y *such that its* k*th coordinate* $y_k = \tilde{y}_k$ *such that:*

$$CC(V[(x^{min}, y)], S_X \cup \left\{x^{min}\right\}) \geq 0$$

Proof. We explicitly construct some y such that $y_k = \tilde{y}_k$ and the above holds:

- Firstly, $CC(V,S_X) \geq 0$, which implies there exists $h \in E(V,S_X)$.

  $h \in V$ implies that $\forall i, h_i \in V_i$.

  Also, $CC(V_k[(x_k^{min}, \tilde{y}_k)], S_X^k \cup \{x_k^{min}\}) \geq CC(V_k, S_X^k) - 1 \geq 0$. This implies that there exists some $\tilde{h}_k \in E(V_k[(x_k^{min}, \tilde{y}_k)], S_X^k \cup \{x_k^{min}\})$.

- We claim that $y = (h_1(x_1^{min}), ..., \tilde{h}_k(x_k^{min}), ..., h_n(x_n^{min}))$ satisfies the condition.

  To show this, define $\tilde{h} = (h_1, ..., \tilde{h}_k, ..., h_n)$.

  Firstly, since $h_i \in V_i$ (for $i \neq k, i \in [n]$) and $\tilde{h}_k \in V_k$, we have that $\tilde{h} \in \times_{i\in[n]} V_i = V$. Also, $\tilde{h}(x^{min}) = y$. Therefore, $\tilde{h} \in V^y_{x^{min}}$.

- We will show that $\tilde{h} \in E(V^y_{x^{min}}, S_X \cup \{x^{min}\})$, which proves the result.

  From Lemma D.10, We have that:

  $$\tilde{h}_k \in E(V_k[(x_k^{min}, \tilde{y}_k)], S_X^k \cup \left\{x_k^{min}\right\}) \subseteq E(V_k[(x_k^{min}, \tilde{y}_k)], (S_X \cup \left\{x^{min}\right\})^k)$$

  since $S^k \cup \{x_k^{min}\} \supseteq (S_X \cup \{x^{min}\})^k$.

  For all $i \neq k$, we have:

  $$h \in E(V,S_X) \Rightarrow h_i \in E(V_i, S_X^i) \Rightarrow h_i \in E(V_i[(x_i^{min}, y_i)], S_X^i \cup \left\{x_i^{min}\right\})$$

since for all $h' \in V_i \setminus \{h_i\}$ with $h'(x_i^{min}) = y_i = h_i(x_i^{min})$, h' must be such that $h'(\mathcal{X} \setminus (S_X^i \cup \{x_i^{min}\})) \neq h_i(\mathcal{X} \setminus (S_X^i \cup \{x_i^{min}\}))$. Since this holds for all $h' \in V_i[(x_i^{min}, y_i)] \setminus \{h_i\}$, we have $h_i \in E(V_i[(x_i^{min}, y_i)], S_X^i \cup \{x_i^{min}\})$.

From Lemma D.10, We have that:

$$h_i \in E(V_i[(x_i^{min}, y_i)], S_X^i \cup \left\{x_i^{min}\right\}) \subseteq E(V_i[(x_i^{min}, y_i)], (S_X \cup \left\{x^{min}\right\})^i)$$

since $S_X^i \cup \{x_i^{min}\} \supseteq (S_X \cup \{x^{min}\})^i$.

Hence,

$$\tilde{h} \in \times_{i=1}^k E(V[(x_i^{min}, y_i)], (S_X \cup \left\{x^{min}\right\})^i)) \Rightarrow \tilde{h} \in E(V[(x^{min}, y)], S_X \cup \left\{x^{min}\right\}))$$

since from Lemma G.1, we have that:

$$E(V[(x^{min}, y)], S_X \cup \left\{x^{min}\right\})) = \times_{i=1}^k E(V[(x_i^{min}, y_i)], (S_X \cup \left\{x^{min}\right\})^i))$$

□

Remark G.15. *As $CC(\times_{i\in[n]}(V_i)^{y_i}_{x_i^{min}}, S_X \cup \{x^{min}\}) \geq 0$, the precondition for induction hypothesis holds.*

G.4 Multi-task Active Learning without Abstention

We also investigate the related multi-task, minimax active learning setting without abstention, which may be of independent interest. To our knowledge, this is also an open problem. Our goal is again to relate the multi-task complexity to the single-task complexity. Since abstention is the cause of several of the negative examples above, one can prove more general upper bounds when labels have to be given.

G.4.1 Game Setup

Without abstention, the state may now be tracked simply with VS (instead of E-VS). The analogous CC game may be defined as follows:

$$CC(V, S_X) = \begin{cases} -\infty & |V| = 0 \\ 0, & |V| = 1 \\ \min_{x\in\mathcal{X}\setminus S_X} \max_{y\in\{-1,+1\}} \left(1 + CC(V_x^y, S_X \cup \{x\})\right), & |V| \geq 2 \end{cases}$$

G.4.2 Lemmas Used

Lemma G.16. *For any S_X, $|V| \geq 1 \Leftrightarrow CC(V, S_X) \geq 0$.*

Proof. **Base Case:** We prove this by induction on $|S_X|$. If $S_X = \mathcal{X}$, then $|V| \geq 1 \Rightarrow |V| = 1 \Rightarrow CC(V, S_X) = 0$.

Induction Step: Suppose this is true for $|S_X| = |\mathcal{X}|, ..., j+1$. Now $|S_X| = j$. Let $h \in V$.

If $|V| = 1$, then the result holds.

Otherwise, $|V| \geq 2$. We will show that $|V| \geq 2 \Rightarrow CC(V, S_X) \geq 1$:

$$\begin{aligned} CC(V, S_X) &= \min_{x\in\mathcal{X}\setminus S_X} \max_{y\in\{+1,-1\}} 1 + CC(V_x^y, S_X \cup \{x\}) \\ &\geq 1 + CC(V[(x^*, h(x^*)], S_X \cup \{x^*\})) \\ &\quad \text{(for } x^* = \arg\min_{x\in\mathcal{X}\setminus S_X} \max_{y\in\{+1,-1\}} 1 + CC(V_x^y, S_X \cup \{x\})) \\ &\geq 1 \end{aligned}$$

The last step that $CC(V[(x^*, h(x^*)], S_X \cup \{x^*\})) \geq 0$ follows from induction hypothesis, whose precondition is satisfied because $h \in V \Rightarrow h \in V[(x^*, h(x^*)]$.

($\Leftarrow$) $|V| = 0 \Rightarrow CC(V, S_X) = -\infty < 0$, hence $CC(V, S_X) \geq 0 \Rightarrow |V| \geq 1$. □

Corollary G.17. *We have that:*

1. $CC(V, S_X) = -\infty \Leftrightarrow |V| = 0$
2. $CC(V, S_X) = 0 \Leftrightarrow |V| = 1$

Proof.

1. ($\Rightarrow$): Follows from that $CC(V, S_X) < 0 \Rightarrow |V| < 1 \Rightarrow |V| = 0$.

   ($\Leftarrow$): Follows from the base case definition of CC.

2. ($\Rightarrow$): From the above, we have that $|V| \geq 2 \Rightarrow CC(V, S_X) \geq 1$. And so, $CC(V, S_X) \leq 0 \Rightarrow |V| \leq 1$.

   The result follows since $CC(V, S_X) = 0 \neq -\infty \Rightarrow |V| \neq 0 \Rightarrow |V| = 1$.

   ($\Leftarrow$): Follows from the base case definition of CC.

□

Lemma G.18. *For $V' \subseteq V$ and any $S_X \subseteq \mathcal{X}$:*

$$CC(V, S_X) \geq CC(V', S_X)$$

Proof. We will prove this statement by induction on the size of S_X.

Base Case: $S_X = \mathcal{X}$. This means $CC(V, S_X), CC(V', S_X)$ are at the base-case. If $|V'| = 1 \Rightarrow |V| = 1$, and the statement holds. If $|V'| = 0$, the statement holds since RHS is equal to $-\infty$.

Induction Step: Suppose the statement holds for $|S_X| = |\mathcal{X}|, ..., j + 1$ and any $V' \subseteq V$. Consider some S_X such that $|S_X| = j$.

(a) First, we examine what happens if $|V| \leq 1$.

(i) if $|V| = 0 \Rightarrow |V'| = 0$, then $CC(V, S_X) = -\infty = CC(V', S_X)$

(ii) if $|V| = 1 \Rightarrow |V'| \leq 1$, so $CC(V, S_X) = 0 \geq CC(V', S_X)$.

(b) If $|V| \geq 2$ and $|V'| \leq 1$, then since $|V| \geq 1$, we have $CC(V, S_X) \geq 0 \geq CC(V', S_X)$ using Lemma G.16.

(c) The remaining case is when $|V| \geq 2$ and $|V'| \geq 2$.

We have that:

$$\begin{aligned}
CC(V, S_X) &= \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{+1,-1\}} 1 + CC(V_x^y, S_X \cup \{x\}) \qquad \text{(since } |V| \geq 2\text{, we can unroll)} \\
&\geq \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{+1,-1\}} 1 + CC((V')_x^y, S_X \cup \{x\}) \\
&\quad \text{(for all } x, y,\ V' \subseteq V \Rightarrow V'[(x, y)] \subseteq V[(x, y)]\text{, so we may apply induction hypothesis)} \\
&= CC(V', S_X)
\end{aligned}$$

□

Lemma G.19. *For any data point (x_1, y_1) for $x_1 \notin S_X$ and $y_1 \in \{+1, -1\}$:*

$$CC(V[(x_1, y_1)], S_X \cup \{x_1\}) \leq CC(V, S_X)$$

Proof. **Base Case:**

We first handle the case when $|V[(x_1, y_1)]| \leq 1$:

If $|V[(x_1, y_1)]| = 0$, then the result holds.

If $|V[(x_1, y_1)]| = 1 \Rightarrow |V| \geq 1$, and the result holds from Lemma G.16.

This covers the base case when $S_X = \mathcal{X}$.

Induction Step: Suppose the statement holds for when $|S_X| = |\mathcal{X}|, ..., j+1$. Let $|S_X| = j$.

It suffices to examine the case that $|V[(x_1, y_1)]| \geq 2$, which implies that $|V| \geq 2$.

Define

$$x' \in \arg\min_{x \in \mathcal{X} \setminus S_X} \max_y 1 + CC(V[(x', y)], S \cup \{x'\});$$

with this definition,

$$CC(V, S_X) = \max_y 1 + CC(V[(x', y)], S_X \cup \{x'\})$$

If $x' = x_1$, then the result follows.

If $x' \neq x_1$, then $x' \in \mathcal{X} \setminus S \cup \{x_1\}$, and we can write:

$$\begin{aligned}
CC(V[(x_1, y_1)], S_X \cup \{x_1\}) &\leq \max_y 1 + CC(V[(x_1, y_1), (x', y)], S_X \cup \{x_1, x'\}) \\
&\qquad \text{(as } |V[(x_1, y_1)]| \geq 2 \text{ so we can unroll with } x' \in \mathcal{X} \setminus S_X \cup \{x_1\}) \\
&\leq \max_y 1 + CC(V[(x', y)], S_X \cup \{x'\}) \\
&\qquad \text{(using induction hypothesis)} \\
&= CC(V, S_X)
\end{aligned}$$

□

Lemma G.20. *For $x \in \mathcal{X} \setminus S_X$ and some $y \in \{+1, -1\}$:*

$$CC(V[(x, y)], S_X) = CC(V[(x, y)], S_X \cup \{x\})$$

Proof. We show this by induction on size of S_X.

Base Case: Firstly, the version space are the same, $V[(x, y)]$.

So LHS is equal to RHS when $|V[(x, y)]| \leq 1$ in the base case. This covers the case when $S_X = \mathcal{X}$.

Induction Step: Suppose the statement holds for when $|S_X| = |\mathcal{X}|, ..., j+1$. Let $|S_X| = j$.

It suffices to consider when $|V[(x, y)]| \geq 2$. We may write:

$$CC(V, S_X) = \min_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1, -1\}} 1 + CC(V[(x', y')]], S_X \cup \{x'\})$$

Define $x^* \in \arg\min_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1,-1\}} 1 + CC(V[(x', y')]], S_X \cup \{x'\})$.

We will show that $x^* \neq x$.

In fact, for any $x' \in \mathcal{X} \setminus S_X$, $x' \neq x^*$ (which exists because $\{x\} \subset \mathcal{X} \setminus S_X$) we have:

$$\begin{aligned}
&\max_{y' \in \{+1, -1\}} 1 + CC(V_x^y[(x, y')]], S_X \cup \{x\}) \\
&= \max(1 + CC(V_x^y, S_X \cup \{x\}), 1 + CC(\emptyset, S_X \cup \{x\})) \\
&= 1 + CC(V_x^y, S_X \cup \{x\}) && \text{(maximized at when } y' = y) \\
&\geq \max_{y' \in \{+1, -1\}} 1 + CC(V_x^y[(x', y')]], S_X \cup \{x, x'\}) && \text{(using Lemma G.19)} \\
&= \max_{y' \in \{+1, -1\}} 1 + CC(V_x^y[(x', y')]], S_X \cup \{x'\}) \\
&\qquad \text{(using induction hypothesis since } |S_X \cup \{x'\}| = j+1)
\end{aligned}$$

And so,

$$
\begin{aligned}
CC(V[(x,y)], S_X) &= \min_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1,-1\}} 1 + CC(V_x^y[(x',y')]], S_X \cup \{x'\}) \\
&= \min_{x' \in \mathcal{X} \setminus (S_X \cup \{x\})} \max_{y' \in \{+1,-1\}} 1 + CC(V_{x'}^{y'}[(x,y)]], S_X \cup \{x'\}) && \text{(since } x^* \neq x) \\
&= \min_{x' \in \mathcal{X} \setminus (S_X \cup \{x\})} \max_{y' \in \{+1,-1\}} 1 + CC(V_{x'}^{y'}[(x,y)]], (S_X \cup \{x'\}) \cup \{x\}) && \text{(using induction hypothesis since } |S_X \cup \{x'\}| = j+1) \\
&= \min_{x' \in \mathcal{X} \setminus (S_X \cup \{x\})} \max_{y' \in \{+1,-1\}} 1 + CC(V_x^y[(x',y')]], (S_X \cup \{x\}) \cup \{x'\}) && \text{(rearranging)} \\
&= CC(V[(x,y)], S_X \cup \{x\})
\end{aligned}
$$

□

G.4.3 UPPER BOUND

Theorem G.21. *For all $V \subseteq \mathcal{H}$ and $S_X \subseteq \mathcal{X}$:*

$$CC(V, S_X) \leq \sum_{i=1}^{n} CC(V_i, S_X^i)$$

Proof. We will proceed by induction on the size of S_X:

Base Case: When $S_X = \mathcal{X}$. In this case, $S_X^i = \mathcal{X}_i$. So all CC's are at the base-case.

It suffices to check that if $CC(V, S_X) = 0 \Rightarrow \forall i, CC(V_i, S_X^i) = 0$.

This follows because $CC(V, S_X) = 0 \Leftrightarrow |V| = 1$. By definition of V_i, $|V_i| = 1$. And so, $CC(V, S_X) = 0 = \sum_{i=1}^{n} CC(V_i, S_X^i)$.

Induction Step:

Suppose the following holds for $S_X \subset X$ for $|S_X| = |\mathcal{X}|, ..., j+1$. Now let $|S_X| = j$ (with $S_X \subset \mathcal{X}$).

We consider three cases:

- $\exists i, V_i = \emptyset$
- $\forall i, |V_i| \geq 1$ and $\forall i, |V_i| = 1$
- $\forall i, |V_i| \geq 1$ and $\exists i, |V_i| \geq 2$

1. **If there is i such that $CC(V_i, S_X^i) = -\infty$.**

   Then $V_i = \emptyset \Rightarrow V = \emptyset$, and therefore, $CC(V, S_X) = -\infty$.

2. **For all i, $CC(V_i, S_X^i) = 0$.**

   This means that for all i, $|V_i| = 1$. And we wish to show that $|V| \leq 1$, which would imply that $CC(V, S_X) \leq 0 = \sum_{i=1}^{n} CC(V_i, S_X^i)$.

   Suppose not, there exists $h, h' \in V$. Then, $h \neq h' \Rightarrow \exists i$ such that $h_i \neq h'_i \Rightarrow h_i, h'_i \in V_i \Rightarrow |V_i| \geq 2$, which is a contradiction.

3. **Exists i such that $CC(V_i, S_X^i) \geq 1$, and $CC(V_j, S_X^j) \geq 0$ for all j.**

   Assume WLOG $i = 1$. Note that if $|V| \leq 1$, then $CC(V, S_X) \leq 0 \leq \sum_{i=1}^{n} CC(V_i, S_X^i)$.

   And so, we will consider the case when $|V| \geq 2$ and $|V_1| \geq 2$.

Define
$$x_1^* \in \operatorname*{arg\,min}_{x \in \mathcal{X}_1 \setminus S_X^1} \max_{y \in \{+1,-1\}} 1 + CC(V_1[(x_1^*, y)], S_X^1 \cup \{x_1^*\})$$
we may express:
$$CC(V_1, S_X^1) = \max_{y \in \{+1,-1\}} 1 + CC(V_1[(x_1^*, y)], S_X^1 \cup \{x_1^*\}) \tag{11}$$

Moreover, we have that $\exists x^* \in \mathcal{X} \setminus S_X$ with the first coordinate equal to x_1^*. And so,
$$CC(V, S_X) \leq \max_{y \in \{+1,-1\}^n} 1+CC(V[(x^*, y)], S_X \cup \{x^*\}) = 1+CC(V[(x^*, y')], S_X \cup \{x^*\})$$

With this,
$$\begin{aligned}
CC(V, S_X) &\leq 1 + CC(V[(x^*, y')], S_X \cup \{x^*\}) \\
&\leq 1 + \sum_{i=1}^{n} CC((V[(x^*, y')])_i, (S_X \cup \{x^*\})^i) \quad \text{(using induction hypothesis)} \\
&= 1 + CC((V[(x^*, y')])_1, (S_X \cup \{x^*\})^1) + \sum_{i=2}^{n} CC((V[(x^*, y')])_i, (S_X \cup \{x^*\})^i) \\
&\leq 1 + CC(V_1[(x_1^*, y_1')], S_X^1 \cup \{x_1^*\}) + \sum_{i=2}^{n} CC((V[(x^*, y')])_i, (S_X \cup \{x^*\})^i) \\
&\qquad \text{(using Lemma G.18 and } \diamond \text{ for task 1)} \\
&\leq CC(V_1, S_X^1) + \sum_{i=2}^{n} CC((V[(x^*, y')])_i, (S_X \cup \{x^*\})^i) \\
&\qquad \text{(using Equation 11)} \\
&\leq CC(V_1, S_X^1) + \sum_{i=2}^{n} CC(V_i[(x_i^*, y_i')], S_X^i \cup \{x_i^*\}) \\
&\qquad \text{(using Lemma G.18 and } \diamond \text{ for tasks 2 to } n) \\
&\leq CC(V_1, S_X^1) + \sum_{i=2}^{n} CC(V_i, S_X^i) \quad \text{(using Lemma G.19 for tasks 2 to } n)
\end{aligned}$$

For any task i:

Lemma G.22. *For any x, y and V,*
$$(V[(x, y)])_i \subseteq V_i[(x_i, y_i)]$$

Proof. We have that $h_i' \in (V[(x, y)])_i \Rightarrow \exists h \in V[(x, y)], h_i = h_i'$.
$h_i \in V_i[(x_i, y_i)]$, since $h \in V[(x, y)] \Rightarrow h_i \in V_i \wedge h_i(x_i) = y_i$ (from $h(x) = y$).
And so, we get that $h_i' = h_i \in V_i[(x_i, y_i)]$.

□

Using this lemma, we may apply Lemma G.18 to get that:
$$CC((V[(x^*, y')])_i, (S_X \cup \{x^*\})^i) \leq CC(V_i[(x_i^*, y_i')], (S_X \cup \{x^*\})^i)$$

We will show below that:
$$CC(V_i[(x_i^*, y_i')], (S_X \cup \{x^*\})^i) = CC(V_i[(x_i^*, y_i')], S_X^i \cup \{x_i^*\})$$

($\diamond$): There are two cases to consider:

- Case 1: $(S_X \cup \{x^*\})^i = S_X^i \cup \left\{x_i^*\right\}$; in this case, $CC(V_i[(x_i^*, y_i')], (S_X \cup \{x^*\})^i) = CC(V_i[(x_i^*, y_i')], S_X^i \cup \left\{x_i^*\right\})$ holds;
- Case 2: $(S_X \cup \{x^*\})^i = S_X^i$, in this case, $CC(V_i[(x_i^*, y_i')], (S_X \cup \{x^*\})^i) = CC(V_i[(x_i^*, y_i')], S_X^i) = CC(V_i[(x_i^*, y_i')], S_X^i \cup \left\{x_i^*\right\})$, where the last equality uses Lemma G.20.

□

G.4.4 Lower Bound

Example of non-Cartesian Product V can reverse inequality:

Proposition G.23. *There exists a non-Cartesian product version space V and S_X such that:*

$$CC(V, S_X) < \max_{i\in[n]} CC(V_i, S_X^i)$$

Proof. Consider $\mathcal{H} = \left\{(h_1, g_1), (h_2, g_1), (h_3, g_2)\right\}$. h_i and g_j's are thresholds.

Let $\mathcal{X} = \left\{[x_{11}, x_2], [x_{12}, x_2]\right\}$, where x_{11} separates h_1, h_2, x_{12} separates h_2, h_3 and x_2 separates g_1, g_2.

Let $S = \emptyset$, so $S_X = S_X^1 = S_X^2 = \emptyset$.

$V = \mathcal{H} = \left\{(h_1, g_1), (h_2, g_1), (h_3, g_2)\right\}$, $V_1 = \{h_1, h_2, h_3\}$, $V_2 = \{g_1, g_2\}$.

Then, we have that $CC(V_1, \emptyset) = 2$ for $V_1 = \{h_1, h_2, h_3\}$. However, $CC(V, \emptyset) = 1$, since one needs to query $[x_{11}, x_2]$ only. □

Remark G.24. *The observation is that x_{11} helps to distinguish between h_1 and $h_2 \in V_1$, while x_2 helps with distinguishing between g_1 and $g_2 \in V_2$, which in turn helps to distinguish between $\{h_1, h_2\}$ and $\{h_3\} \subset V_1$.*

Theorem G.25. *For all $V = \times_{i\in[n]} V_i$ and $S_X \subseteq \mathcal{X}$ such that $CC(V, S_X) \geq 0$:*

$$CC(V, S_X) \geq \max_{i\in[n]} CC(V_i, S_X^i)$$

Proof. We prove this by induction on the size of S_X.

Base Case: $S_X = \mathcal{X} \Rightarrow S_X^i = \mathcal{X}_i$.

If $CC(V, \mathcal{X}) = 0$, then $|V| = 1 \Rightarrow |V_i| = 1, \forall i \Rightarrow CC(V_i, S_X^i) = 0$ for all i.

Induction Step: Suppose the following holds for $|S_X| = |\mathcal{X}|, ..., j + 1$. Now let $|S_X| = j$, note that $S_X \subset X$.

We first handle the base cases.

If $CC(V, S_X) = 0$, then $V = \{h\} \Rightarrow \forall i, V_i = \{h_i\}$ (due to the Cartesian product structure of V) $\Rightarrow CC(V_i, S_X^i) = 0$.

Now, if $CC(V, S_X) \geq 1$ and if $k = \arg\max_{i\in[n]} CC(V_i, S_X^i)$, then it suffices to verify the statement when $CC(V_k, S_X^k) \geq 1$.

Define:

$$x^{min} = \operatorname*{arg\,min}_{x\in\mathcal{X}\setminus S_X} \max_{y'\in\mathcal{Y}} \mathbb{1}(y' \neq \perp) + CC(V_x^{y'}, S_X \cup \{x\})$$

From definition, $\mathcal{X}_k \setminus S_X^k = (\mathcal{X} \setminus S_X)_k = \left\{x' \in \mathcal{X}_k : \exists x \in \mathcal{X} \setminus S_X, x_k = x'\right\}$. And so $x_k^{min} \in \mathcal{X}_k \setminus S_X^k$ since $x^{min} \in \mathcal{X} \setminus S_X$. Since $CC(V_k, S_X^k) \geq 1$, we know there exists $\tilde{y}_k$ such that:

$$CC(V_k, S_X^k) \leq 1 + CC(V_k[(x_k^{min}, \tilde{y}_k)], S_X^k \cup \left\{x_k^{min}\right\})$$

Note in particular that $V_k[(x_k^{min}, \tilde{y}_k)] \neq \emptyset$ as otherwise $CC(V_k, S_X^k) \leq -\infty$ (which contradicts our assumption):

$$\begin{aligned}
CC(V, S_X) &= \min_{x \in \mathcal{X} \setminus S_X} \max_{y' \in \mathcal{Y}} 1 + CC(V_x^{y'}, S_X \cup \{x\}) \quad (\ \mathcal{X} \setminus S_X \text{ is non-empty, since } S_X \subset \mathcal{X}) \\
&= \max_{y' \in \mathcal{Y}} 1 + CC(V_{x^{min}}^{y'}, S_X \cup \left\{x^{min}\right\}) \\
&\geq 1 + CC(\times_{i \in [n]} (V_i)_{x_i^{min}}^{y_i}, S_X \cup \left\{x^{min}\right\}) \\
&\qquad \text{(setting } y' = y \text{ as constructed below (†) and using that } V_{x^{min}}^{y} = \times_{i \in [n]} (V_i)_{x_i^{min}}^{y_i}) \\
&\geq 1 + \max_{i \in [n]} CC((V_i)_{x_i^{min}}^{y_i}, (S_X \cup \left\{x_i^{min}\right\})^i) \\
&\qquad \text{(using induction hypothesis since } x^{min} \notin S_X \text{, so } |S_X \cup \left\{x^{min}\right\}| = j+1) \\
&\geq 1 + CC((V_k)_{x_k^{min}}^{\tilde{y}_k}, (S_X \cup \left\{x^{min}\right\})^k) \qquad \text{(by construction, } y_k = \tilde{y}_k) \\
&= 1 + CC((V_k)_{x_k^{min}}^{\tilde{y}_k}, S_X^k \cup \left\{x_k^{min}\right\}) \\
&\qquad \text{(note that } x_k^{min} \in (\mathcal{X} \setminus S_X)_k \text{, so } x_k^{min} \in \mathcal{X}_k \setminus S_X^k \text{ and } \diamond) \\
&\geq CC(V_k, S_X^k)
\end{aligned}$$

(†) : **Claim:** There exists some y such that $y_k = \tilde{y}_k$ and $V_{x^{min}}^{y} \neq \emptyset$ (that is, $(V_i)_{x_i^{min}}^{y_i} \neq \emptyset$ for each i).

Firstly, $CC(V, S_X) \geq 0 \Rightarrow |V| \geq 1$. This means that there exists $h \in V$, and that $\forall i, \exists h_i \in V_i$.

Since $V_k[(x_k^{min}, \tilde{y}_k)] \neq \emptyset$, there exists some $\tilde{h}_k \in V_k[(x_k^{min}, \tilde{y}_k)] \neq \emptyset$.

We claim that $y = (h_1(x_1^{min}), ..., \tilde{h}_k(x_k^{min}), ..., h_n(x_n^{min}))$ satisfies the property.

Let $h = (h_1, ..., \tilde{h}_k, ..., h_n)$. Then we have $h \in V_{x^{min}}^{y}$, since:

i) $h_i \in V_i, \tilde{h}_k \in V_k$ implies $h \in \times_{i \in [n]} V_i = V$

ii) $h(x^{min}) = y$.

And so, $|V_{x^{min}}^{y}| \geq 1 \Rightarrow CC(V_{x^{min}}^{y}, S_X \cup \left\{x^{min}\right\}) \geq 0$, which means we meet the precondition needed to use the induction hypothesis.

($\diamond$): For task k, We know that $(S_X \cup \left\{x^{min}\right\})^k$ is either S_X^k or $S_X^k \cup \left\{x_k^{min}\right\}$. In the latter case, equality holds.

In the former case, we may use Lemma G.20 to get that equality also holds:

$CC((V_k)_{x_k^{min}}^{\tilde{y}_k}, (S_X \cup \left\{x^{min}\right\})_k) = CC((V_k)_{x_k^{min}}^{\tilde{y}_k}, S_X^k) = CC((V_k)_{x_k^{min}}^{\tilde{y}_k}, S_X^k \cup \left\{x_k^{min}\right\})$.

□

H MISCELLANEOUS

H.1 COMPARISON OF REPRESENTATIONS

H.1.1 DATA-BASED GAME REPRESENTATION

We begin with defining a natural set of state representation, motivated by the definition of identifiability for determining the termination condition.

Definition H.1. *Given the set of labeled examples and their labels S, and the queried examples S_X, classifier $h \in \mathcal{H}$ is said to be identifiable with respect to (S, S_X), if (1) h is consistent with S; (2) for all $h' \in \mathcal{H}$ consistent with S,*

$$h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \implies h' = h$$

The above definition naturally motivates the following definition of effective version space:

Definition H.2. *Given the set of labeled examples and their labels S, and the queried examples S_X, define its induced effective version space as*

$$F(S, S_X) = \left\{h \in \mathcal{H} : h \text{ is identifiable with respect to } (S, S_X)\right\}$$

With this, it is natural to recursively define the learning game, taking the set of labeled examples and their labels S, and the queried examples S_X as the state representation.

$$f(S, S_X) = \begin{cases} -\infty & F(S, S_X) = \emptyset \\ 0, & |F(S, S_X)| = 1 \\ \min_{x \in \mathcal{X} \setminus S_X} \max \begin{pmatrix} f(S \cup \left\{(x, \perp)\right\}, S_X \cup \{x\}) \\ 1 + f(S \cup \left\{(x, +1)\right\}, S_X \cup \{x\}) \\ 1 + f(S \cup \left\{(x, -1)\right\}, S_X \cup \{x\}) \end{pmatrix}, & |F(S, S_X)| \geq 2, \end{cases}$$

Here, we use the base-case game payoffs to encode the labeler's promise of identifiability. Non-identifiability ($F(S, S_X) = \emptyset$) leads to a terminal payoff of $-\infty$. Identifiability constrains the labeler to not provide arbitrary labels and "string along" the learner for as long as possible. As we will later see, this constraint is not crucial, as the algorithm we develop is also robust to a labeler that does not guarantee identifiability.

H.1.2 VERSION SPACE-BASED GAME REPRESENTATION

We now turn to the version space game representation, which we use throughout, and prove it is correct.

Definition H.3. *Given a labeled dataset S and a set of classifiers V, define version space $V[S] = \left\{h \in V : \forall (x, y) \in S \wedge y \neq \perp, h(x) = y\right\}$ as the subset of classifiers in V consistent with S.*

Definition H.4. *Given the set of labeled examples and their labels S, and the queried examples S_X, classifier $h \in \mathcal{H}$ is said to be identifiable with respect to (S, S_X) if:*

- *h is consistent with S, $h \in \mathcal{H}[S]$.*
- *for all other consistent $h' \in \mathcal{H}[S]$: $h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \implies h' = h$, where for brevity we denote $h_1(S_X) = h_2(S_X) \iff \forall x \in S_X \,.\, h_1(x) = h_2(x)$.*

Definition H.5. *Given a set of classifiers V and a set of queried examples S_X, define*

$$E(V, S_X) = \left\{h \in V : \forall h' \in V \setminus \{h\} : h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)\right\}$$

as the effective version space (E-VS) with respect to V and S_X.

The following proposition relates the effective version space to the classical notion of version space:

Proposition H.6.

$$F(S, S_X) = E(\mathcal{H}[S], S_X)$$

Proof.

$$
\begin{aligned}
h \in F(S, S_X) &\Leftrightarrow h \in \mathcal{H}[S] \wedge \forall h' \in \mathcal{H}[S], h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \implies h' = h \\
&\Leftrightarrow h \in \mathcal{H}[S] \wedge \forall h' \in \mathcal{H}[S], h' \neq h \implies h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X) \\
&\qquad\qquad\text{(taking the contrapositive)} \\
&\Leftrightarrow h \in E(\mathcal{H}[S], S_X)
\end{aligned}
$$

□

Thus, another potential state space representation is using the version space and the data that has been queried. We may define the minimax game as follows, which corresponds to Protocol 4.

$$
CC(V, S_X) = \begin{cases} -\infty & E(V, S_X) = \emptyset \\ 0, & |E(V, S_X)| = 1 \\ \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{-1,+1,\perp\}} \left(\mathbb{1}(y \neq \perp) + CC(V[(x, y)], S_X \cup \{x\}) \right), & |E(V, S_X)| \geq 2 \end{cases}
$$

The following structural lemma justifies that this is also a valid representation.

Lemma H.7. $f(S, S_X) = CC(\mathcal{H}[S], S_X)$

Proof. We prove this by backward induction on S_X.

Base case: $S_X = \mathcal{X}$**.** In this case, $F(S, \mathcal{X}) = E(\mathcal{H}[S], \mathcal{X})$ has size 0 or 1; in both cases, $f(S, S_X) = CC(\mathcal{H}[S], S_X)$ by their respective definitions in the bases cases.

Inductive case. Suppose $f(S, S_X) = CC(\mathcal{H}[S], S_X)$ holds for any S and any S_X such that $|S_X| \geq j + 1$. Now consider any S and any S_X of size j.

If $F(S, S_X) = E(\mathcal{H}[S], S_X)$ has size 0 or 1, $f(S, S_X) = CC(\mathcal{H}[S], S_X)$ holds true.

Otherwise, $|F(S, S_X)| = |E(\mathcal{H}[S], S_X)| \geq 2$. By inductive hypothesis, for any $x \in \mathcal{X} \setminus S_X$:

$$
\begin{aligned}
f(S \cup \{(x, \perp)\}, S_X \cup \{x\}) &= CC(\mathcal{H}[S \cup \{(x, \perp)\}], S_X \cup \{x\}) \\
f(S \cup \{(x, +1)\}, S_X \cup \{x\}) &= CC(\mathcal{H}[S \cup \{(x, +1)\}], S_X \cup \{x\}) \\
f(S \cup \{(x, -1)\}, S_X \cup \{x\}) &= CC(\mathcal{H}[S \cup \{(x, -1)\}], S_X \cup \{x\})
\end{aligned}
$$

Therefore, for any x:

$$
\max \begin{pmatrix} f(S \cup \{(x, \perp)\}, S_X \cup \{x\}) \\ 1 + f(S \cup \{(x, +1)\}, S_X \cup \{x\}) \\ 1 + f(S \cup \{(x, -1)\}, S_X \cup \{x\}) \end{pmatrix} = \max \begin{pmatrix} CC(\mathcal{H}[S \cup \{(x, \perp)\}], S_X \cup \{x\}) \\ 1 + CC(\mathcal{H}[S \cup \{(x, +1)\}], S_X \cup \{x\}) \\ 1 + CC(\mathcal{H}[S \cup \{(x, -1)\}], S_X \cup \{x\}) \end{pmatrix}
$$

Taking minimum over $x \in \mathcal{X} \setminus S_X$, we also have $f(S, S_X) = CC(\mathcal{H}[S], S_X)$.

This completes the induction.

□

I Additional Related Works

Abstaining Classifiers: Prior works have studied the task of learning a predictor with the ability to abstain (Puchkin & Zhivotovskiy, 2021; Zhu & Nowak, 2022). Our settings differ in that we aim to learn the true classifier that does not abstain. Rather, it is the labeler that can abstain during the learning process to slow-down learning.

Cross space learning: One of our constructions is related to the cross space learning (Tao et al., 2022) setup, where each sample is represented in multiple instance spaces. The key observation is that a strategic labeler can force learning on the instance space with the highest sample complexity, by abstaining on all other instance spaces.

Strategic Machine Learning: Strategic ML is a line of work concerned with agent manipulation of inputs into the ML model (Hardt et al., 2016). Much of this topic has focused on inference-time feature manipulation to influence the model output. And among this large body of work, there is a subset that deal with strategic manipulation of labels. In these settings, there are multiple agents, each of whom can (mis)reports their data point label to manipulate the final model trained on all of their collective data (Perote & Perote-Pena, 2004; Dekel et al., 2010; Chen et al., 2018). This line of work largely focuses on the linear-regression setting, under various notions of strategyproofness.

Our work differs from this body of work in considering, at training time (instead of at inference time), how a single labeler can maximize the query complexity of a learner under general hypothesis classes, which includes the linear hypothesis class.

Economics of Knowledge Transfer: We note that the idea of strategically slowing down the transfer of knowledge is not a novel conception. It is a real strategy that people have been documented to use in apprenticeships for example (Garicano & Rayo, 2017; Fudenberg & Rayo, 2019), spanning across several industries such as law, entertainment and culinary arts. There are two reasons that motivate the slowed transfer of expertise.

Firstly, as described in (Garicano & Rayo, 2017; Fudenberg & Rayo, 2019), before the apprentice has learned everything and can graduate, he will be working for the teacher (or master as is often used in apprenticeship parlance) and performing labor for cheap. Thus, this incentivizes the master to slowly down training, so that the apprentice takes longer to graduate and the master can enjoy this cheap labor for longer.

Secondly, the master can better protect the value of his expertise by slowing down the transfer of his expertise. Overly fast transfer of the master's know-how would graduate too many apprentices too quickly, all of whom also have the same expertise and could thus reduce the value of the master's expertise.

In our setting, we consider the relationship between a human teacher (labeler) and a student (machine). There is a similar incentive at play in that, while the learner has yet to learn h^*, the labeler is paid by the learner for the training labels provided. But once h^* is identified, the student has no need for the teacher. And so, this incentivizes the labeler to slow down learning, in order to give and be paid for as many labels as possible. One difference we note is that in this setting, the transfer of expertise has more serious consequences in rendering the labeler's expertise obsolete, which is not the case in the apprenticeship setting.