PRIVATE COMPONENT-BY-COMPONENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We study differentially private learning problems in the realizable setting, where a hypothesis is specified by k components. A direct iteration of private component learners is obstructed by a simple difficulty: an approximate choice of the next component may destroy exact realizability of the labeled sample, even when the next component is locally accurate. We restore realizability using the LabelBoost procedure of Beimel et al. (2015) and recycle data through two alternating reservoirs. The resulting learner, for a target privacy ε, pays only $\widetilde{O}(\sqrt{k}/\varepsilon)$ overhead relative to the active sample requirement of a single component learning step at target accuracy $\Theta(\alpha/k)$. For learning d-dimensional halfspaces over a finite grid of size L, exact realizability makes the direct component-depth objective quasi-concave. Instantiating the framework with the IPConcave algorithm of Nissim et al. (2026b) and with the quasi-concave optimizer of Cohen et al. (2023) yields a realizable sample complexity of $\tilde{O}\left(\frac{1}{\varepsilon\alpha}\cdot\min\{d^{2.5}\log^* L,\ d^{2.5}+d^{1.5}2^{\log^* L}\}\right)$, which improves on the previously known bound of $\tilde{O}\left(\frac{1}{\varepsilon\alpha}\cdot\min\{\frac{1}{\alpha}\cdot d^{5.5}\log^* L,\ d^{2.5}2^{\log^* L}\}\right)$. We also apply the framework to Boolean compositions: given proper private learners for classes $\mathcal{H}_1,\ldots,\mathcal{H}_k$, we obtain a proper private learner for $G(\mathcal{H}_1,\ldots,\mathcal{H}_k)$ for any fixed Boolean function $G:\{0,1\}^k\to\{0,1\}$. The resulting sample bound incurs a $\widetilde{O}(\sqrt{k}/\varepsilon)$ overhead relative to the maximum of the component learning costs at accuracy $\Theta(\alpha/k)$ and the relevant VC-dimension term. Compared with the closure theorem of Alon et al. (2020), this reduces the overhead on a common component sample bound from $\widetilde{O}(k/\varepsilon)$ to $\widetilde{O}(\sqrt{k}/\varepsilon)$.

1 INTRODUCTION

Statistical analysis seeks to extract information about a population without exposing the individuals whose records make the analysis possible. Differential privacy provides a mathematical formulation of this goal by requiring that changing one person's record have only a limited effect on the distribution of the released result (Dwork et al., 2006). Formally, a randomized algorithm A is (ε,δ)-differentially private if, for every pair of datasets S, S' differing in one record and every event E,

$$\Pr[A(S)\in E]\le e^{\varepsilon}\Pr[A(S')\in E]+\delta.$$

The parameters ε and δ control the strength of the guarantee. Throughout this work, we study approximate differential privacy, where $\delta>0$ is small, and ask how much data is needed to learn while providing this protection.

1.1 PRIVATE LEARNING

Privacy is particularly relevant to learning: training examples may contain medical, financial, or other sensitive information about individuals, while the resulting model may be released or used to make predictions for many users. A private PAC learner combines the usual requirement of accurate prediction with differential privacy for its training sample (Kasiviswanathan et al., 2011). In the realizable setting considered here, the examples are labeled by an unknown concept in a class $\mathcal{C}$; the learner must output a hypothesis with population error at most α, with probability at least $1-\beta$.

How much harder does this privacy requirement make learning? Non-private sample complexity is characterized by the VC dimension, up to the dependence on accuracy and confidence (Blumer et al., 1989). For approximate private learning, a comparably tight quantitative characterization remains

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

unknown. There is a qualitative connection to online learning: private learnability is equivalent to finite Littlestone dimension, the parameter governing the optimal realizable online mistake bound (Alon et al., 2019; Bun et al., 2020). This connection, however, leaves a large gap between the known sample-complexity bounds, which can be separated by a tower of exponentials. It also shows that even thresholds on an infinite ordered domain, whose VC dimension is one, need not be privately learnable (Alon et al., 2019). We refer to (Nissim et al., 2026a) for an extended survey.

Finite domains give a different perspective. Every finite concept class can be privately learned using $O(\log |\mathcal{C}|)$ examples when accuracy, confidence, and privacy are fixed (Kasiviswanathan et al., 2011), but this may greatly exceed its VC dimension. As emphasized by Nissim et al. (2026a), known bounds for several natural finite classes are much closer to the non-private baseline, and we lack examples establishing a large separation of the sort suggested by the generic upper bounds. Determining the true cost of privacy is therefore a quantitative question even when a private learner is known to exist.

Halfspaces, which classify points using a linear decision boundary, are a fundamental class for studying the sample cost of privacy. Their non-private sample complexity is well understood, but how many additional examples privacy requires remains unresolved. For halfspaces over a d-dimensional domain of size L, the non-private sample complexity is $\Theta(d)$ at constant accuracy and confidence. The VC lower bound and the private threshold lower bound give $\Omega(d + \log^* L)$ examples as a necessary requirement, in the standard regime of small δ (Blumer et al., 1989; Bun et al., 2015; Alon et al., 2019). For proper learners, which must output a halfspace, the stronger lower bound $\Omega(d \log^* L)$ follows from Bun et al. (2015). On the other hand, the previous private upper bounds give

$$\widetilde{O}\Big(\min\Big\{d^{2.5}2^{\log^* L},\ d^{5.5}\log^* L\Big\}\Big),$$

where this comparison fixes accuracy, confidence, and ε and suppresses logarithmic factors, including the dependence on $1/\delta$. The first bound follows from the geometric constructions of Beimel et al. (2019); Kaplan et al. (2020b), using the improved quasi-concave optimizer of Cohen et al. (2023); the second is due to Nissim et al. (2026b). Both leave a substantial gap in the dimension d compared to the lower bounds.

1.2 Private Component-by-Component Learning

Many learning problems admit a representation of a hypothesis by several coordinates or components. Although choosing all components simultaneously may be difficult, choosing one component while allowing the others to vary can be more tractable. This suggests building the hypothesis sequentially. The challenge is that a private choice is typically only approximately optimal: after fixing it, the remaining components may no longer admit any completion consistent with the sample. Thus, even if the original learning problem is realizable, the intermediate problems need not be. We describe two settings in which this loss of realizability creates additional costs.

Halfspaces. The coordinate-by-coordinate approach to private halfspace learning was developed through private center points by Beimel et al. (2019) and a direct reduction to linear feasibility by Kaplan et al. (2020b), while Nissim et al. (2026b) subsequently improved its dependence on the domain size. In linear feasibility, the dataset is a collection $S = ((a_j, b_j))_{j=1}^n$ of constraints $\langle a_j, z\rangle \geq b_j$ on $z \in \mathbb{R}^d$. Its depth function is

$$\mathrm{depth}_S(z) := \big|\{j \in [n] : \langle a_j, z\rangle \geq b_j\}\big|.$$

Given a feasible system, a private algorithm seeks a point satisfying most of its constraints. A labeled halfspace example induces a linear constraint on the unknown separating hyperplane's parameters, so this task can be used to learn a halfspace (see Appendix I for the formal reduction).

After fixing a prefix $z_{<i}$, the natural objective for coordinate i is

$$Q_i^S(t) := \max_{z_{i+1},\ldots,z_d \in \mathbb{R}} \mathrm{depth}_S(z_{<i}, t, z_{i+1}, \ldots, z_d). \tag{1}$$

It measures how many constraints a completion can satisfy. A one-dimensional function Q is quasi-concave if $Q(t) \geq \min\{Q(u), Q(v)\}$ whenever $u \leq t \leq v$. This property makes private coordinate optimization possible with a very weak dependence on the search-domain size. However,

an approximate choice may leave no fully feasible completion, and subsequent depth objectives can fail to be quasi-concave.[1]

The linear-feasibility algorithms of Kaplan et al. (2020b) and Nissim et al. (2026b) therefore use the convexified depth function cdepth_S, whose superlevel sets are the convex hulls of the corresponding depth superlevel sets. Its coordinate objectives remain quasi-concave under every prefix. The price is a loss when converting back to actual constraint satisfaction:

$$\mathrm{depth}_S(z) \geq (d+1)\,\mathrm{cdepth}_S(z) - dn \qquad \text{(Kaplan et al., 2020b).}$$

Thus a convexified depth of $(1-\eta)n$ can become a depth of $(1-(d+1)\eta)n$. Both algorithms must therefore optimize convexified depth to an error about d times smaller than the desired error in actual depth.

The two approaches differ in how they optimize each coordinate. The general quasi-concave optimizer of Beimel et al. (2013), used by Kaplan et al. (2020b), has a $2^{O(\log^* N)}$ dependence on a finite search domain of size N. Cohen et al. (2023) sharpen this to $\widetilde{O}(2^{\log^* N})$ additive error at constant privacy. In contrast, Nissim et al. (2026b) use additional structure: the objective can be approximated by a random subsample. Their IPConcave procedure partitions the data, computes a maximizer on each block, and privately selects an interior point of these maximizers. Uniform approximation makes the block maximizers good for the full sample, and quasi-concavity makes every point between them good as well. The interior-point algorithm of Cohen et al. (2023), unlike a general quasi-concave optimizer, requires only $\widetilde{O}(\log^* N)$ block maximizers at constant privacy.

The approach of Nissim et al. (2026b) thus obtains an essentially optimal dependence on the domain size L, but incurs a further cost in the dimension: once exact feasibility has been lost, its analysis uses uniform additive VC approximation, requiring blocks of size $\widetilde{O}(d/\rho^2)$ for approximation error ρ. To achieve final error α, the loss over d coordinates and the conversion from convexified depth to depth together require $\rho = \Theta(\alpha/d^2)$. The resulting block size is $\widetilde{O}(d^5/\alpha^2)$.[2]

Our key observation is that maintaining realizability throughout the coordinate construction would allow two improvements. First, optimizing depth directly would eliminate the conversion loss, so that per-coordinate error $\rho = \Theta(\alpha/d)$ would suffice. Second, each block would remain feasible, and we would only need every solution satisfying all its constraints to satisfy all but a ρ fraction of the full sample. This realizable VC guarantee requires just $\widetilde{O}(d/\rho)$ constraints, giving blocks of size $\widetilde{O}(d^2/\alpha)$. Together, these changes save a factor of d^3 at constant accuracy; our halfspace application below realizes this improvement.

Boolean composition. Let $G : \{0,1\}^k \to \{0,1\}$ be fixed and let $\mathcal{H}_1, \ldots, \mathcal{H}_k$ be concept classes on a common domain. The composed class is

$$G(\mathcal{H}_1, \ldots, \mathcal{H}_k) := \{x \mapsto G(h_1(x), \ldots, h_k(x)) : h_i \in \mathcal{H}_i\}.$$

It includes, for example, intersections, unions, and majority votes. Given private learners for the component classes, we would like to learn the composition. The observed labels specify only the value of G, not the individual component labels, so they cannot be supplied directly to a realizable learner for $\mathcal{H}_i$.

For previously selected components $h_1, \ldots, h_{i-1}$, the natural objective for a candidate $h \in \mathcal{H}_i$ is its best completion error, which we seek to minimize:

$$Q_i^S(h) := \min_{c_{i+1} \in \mathcal{H}_{i+1}, \ldots, c_k \in \mathcal{H}_k} \mathrm{err}_S\big(G(h_1, \ldots, h_{i-1}, h, c_{i+1}, \ldots, c_k)\big). \tag{2}$$

Here $\mathrm{err}_S(f)$ is the fraction of examples in S that f misclassifies. Alon et al. (2020) use a relabeling transformation based on LabelBoost (Beimel et al., 2015): select a component labeling with small

[1] The example of Kaplan et al. (2020b) consists of the constraints $z_2 \geq z_1$ and $z_2 \leq -z_1$. The original system is feasible, but after fixing $z_1 = 1$, the remaining objective satisfies $Q_2^S(-1) = Q_2^S(1) = 1$ and $Q_2^S(0) = 0$, violating quasi-concavity.

[2] To maintain a fixed overall privacy guarantee, advanced composition across the d iterations requires reducing the privacy parameter of each iteration by a factor of $\widetilde{O}(\sqrt{d})$. This increases the number of blocks needed by the interior-point algorithm by the same factor, yielding the $d^{5.5}$ dependence in the final sample complexity.

completion error, then learn that labeling with the corresponding private learner. This makes the local learning call realizable. It does not, however, restore exact consistency of the selected global prefix with the original labels. Their construction uses a separate sample for each component and additive generalization bounds for the accumulated completion error. Consequently, its sample bound contains a sum of the component learning costs and a term of order $\widetilde{O}(k^3\,\mathrm{VC}(G(\mathcal{H}_1,\ldots,\mathcal{H}_k))/\alpha^2)$ at constant privacy (Alon et al., 2020, Theorem 29). When the component learners have a common sample requirement m, using k disjoint samples incurs a cost of km. Our construction draws samples from shared reservoirs and combines subsampling amplification with advanced composition, reducing the multiplier on the component learning cost to $\widetilde{O}(\sqrt{k})$ at constant privacy. Doing so requires accounting for the dependence introduced by repeated relabeling, as well as preserving the utility of later component choices.

1.3 Our Results

We give a general learner that preserves realizability while constructing a hypothesis component by component. After each component choice, we relabel the next active sample so that it is realizable by a concept extending the chosen prefix. The component oracles therefore need to guarantee both privacy and utility only on samples realizable by the current prefix class, while the resulting learner is private on arbitrary datasets.

Our framework provides two main advantages. First, we exploit these weaker requirements in our applications to replace uniform additive VC bounds by realizable bounds, reducing the corresponding dependence on inverse accuracy from quadratic to linear. This is particularly useful when errors accumulate across components and each step requires accuracy $\Theta(\alpha/k)$. Second, we perform repeated LabelBoost relabeling (Beimel et al., 2015) while reusing data across rounds, with a total sample overhead of $\widetilde{O}(\sqrt{k}/\varepsilon)$ over the oracle and generalization requirements of one round. Our extended privacy analysis accounts for the dependencies introduced by relabeling, allowing us to obtain this guarantee without allocating a separate sample to every component.

More formally, write each concept as $c_{y_1,\ldots,y_k}$ and let $\mathcal{C}(y_{\le i})$ be the concepts whose representations extend the chosen prefix. We assume oracles $A_1,\ldots,A_k$, where A_i receives a valid prefix $y_{<i}$ and a labeled sample S, and outputs a next component y_i. On every input, the output must extend the prefix to a valid representation. For each fixed prefix, A_i must be (ε_A,δ_A)-differentially private between neighboring samples realizable by $\mathcal{C}(y_{<i})$. If S is realizable by this class and contains at least $M_A(\tau,\beta_A,\varepsilon_A,\delta_A)$ examples, then, with probability at least $1-\beta_A$, the extended class $\mathcal{C}(y_{\le i})$ must contain a hypothesis with empirical error at most τ on S. The algorithm need not output this completion or preserve exact consistency (See Assumption B.4 for the formal assumed properties).

In the informal statements below, $\widetilde{O}$ suppresses logarithmic factors in the dimension, the number of components, accuracy, confidence, and privacy parameters, and lower-order factors in the iterated logarithms. When we abbreviate a component sample requirement by $M_A(\tau)$, the oracle's privacy parameter ε_A is constant, and its failure probability and δ_A are of order β/k and δ/k, respectively. The dependence on these parameters remains part of M_A.

Theorem 1.1 (Informal version of Theorem C.1)**.** *Let $\mathcal{C}$ have VC dimension v and a representation by k components. Suppose private component oracles as above have common sample requirement $M_A(\tau)$. Then, for $\alpha,\beta,\varepsilon,\delta\in(0,1)$ with $\delta\le\beta$, there is a proper (ε,δ)-differentially private realizable (α,β)-PAC learner for $\mathcal{C}$ using*

$$\widetilde{O}\left(\frac{\sqrt{k}}{\varepsilon}\max\left\{\frac{kv}{\alpha},\ M_A\Big(\frac{\alpha}{40k}\Big)\right\}\right)$$

examples. Its privacy guarantee holds on arbitrary input datasets.

The bound separates the cost of one accurate component step from the cost of combining k steps. The active sample must support the oracle at accuracy $\Theta(\alpha/k)$ and generalization within $\mathcal{C}$. The total sample size is only a $\widetilde{O}(\sqrt{k}/\varepsilon)$ multiple of this requirement.

After each component choice, we use the LabelBoost algorithm of Beimel et al. (2015) to relabel the next active sample by a hypothesis in the extended prefix class. The next oracle therefore again sees an exactly realizable problem, while the disagreement introduced by each relabeling is small

enough to accumulate over all k rounds. To control privacy without allocating a disjoint sample to every round (as done by Alon et al. (2020)), we draw the active samples from only two alternating reservoirs. An extended LabelBoost analysis bounds the privacy cost of two consecutive rounds, allowing subsampling amplification and advanced composition across repeated uses of each reservoir. For more details, we refer to Section 2.

1.3.1 APPLICATIONS

Halfspaces. We apply the framework with parameters $k = d + 1$ and VC dimension $v = d + 1$. A key observation is that the direct coordinate-depth objective in Equation (1) is quasi-concave whenever the current prefix has a fully feasible completion (Lemma G.2). In the underlying linear-feasibility problem, A_i receives the current prefix and constraint sample S, forms this objective, and privately selects a coordinate z_i with $Q_i^S(z_i) \geq (1-\tau)|S|$. The selected prefix therefore admits a completion satisfying all but a τ fraction of the constraints. Restoring realizability between rounds lets every A_i optimize depth directly, avoiding the loss from convexification.

Theorem 1.2 (Informal version of Theorem F.5)**.** *Suppose the coordinate oracles for d-dimensional halfspaces satisfy the component-oracle requirements with common sample requirement $M_A(\tau)$. Then there is a proper (ε, δ)-differentially private realizable (α, β)-PAC learner using sample complexity*

$$\widetilde{O}\left(\frac{\sqrt{d}}{\varepsilon}\max\left\{\frac{d^2}{\alpha},\ M_A\left(\frac{\alpha}{80(d+1)}\right)\right\}\right).$$

Let N be the largest finite coordinate-domain size used by the reduction.We instantiate A_i in two ways. The first applies the general quasi-concave optimizer of Cohen et al. (2023) directly to Q_i^S; achieving depth loss at most $\tau|S|$ requires $\widetilde{O}(2^{\log^* N}/\tau)$ constraints at constant privacy. The second uses IPCONCAVE of Nissim et al. (2026b): randomly partition S into blocks, find a fully feasible completion of the prefix on each block, and apply a private interior-point algorithm to these completions' ith coordinates. Realizable generalization makes each selected coordinate good for the full sample, and quasi-concavity ensures that every point between them is good as well. Blocks of size $\widetilde{O}(d/\tau)$ and $\widetilde{O}(\log^* N)$ block solutions suffice, giving a cost of $\widetilde{O}(d \log^* N/\tau)$. Combining these optimization costs with the framework's generalization requirement along with the fact that our choice of N satisfy $\log^* N \approx \log^* L$ yields the following learning bounds.

Theorem 1.3 (Informal versions of Theorems F.8 and F.14)**.** *For d-dimensional halfspaces over a finite grid of size L, there is a proper (ε, δ)-differentially private realizable (α, β)-PAC learner with sample complexity*

$$\widetilde{O}\left(\frac{1}{\varepsilon\alpha}\min\left\{d^{2.5} + d^{1.5}2^{\log^* L},\ d^{2.5}\log^* L\right\}\right).$$

Namely, the IPCONCAVE instantiation reduces the dimension dependence from $d^{5.5}$ to $d^{2.5}$ while retaining the $\log^* L$ dependence. The general quasi-concave instantiation reduces the coefficient of $2^{\log^* L}$ from $d^{2.5}$ to $d^{1.5}$, with an additional $d^{2.5}$ term. All these gains come from restoring realizability before each coordinate optimization.

Boolean composition. Suppose each $\mathcal{H}_i$ has a proper private realizable learner B_i. Given a prefix $h_1, \ldots, h_{i-1}$ and a sample S realizable by its restricted composition class, we define A_i to use the relabel-and-learn construction of Alon et al. (2020): First, the LabelBoost relabeling step selects $\widetilde{h}_i \in \mathcal{H}_i$ with small completion error $Q_i^S(\widetilde{h}_i)$ (defined in Equation (2)), using the exponential mechanism over the labelings of S induced by $\mathcal{H}_i$. It then replaces the sample labels by $\widetilde{h}_i$'s predictions and runs the empirical version of B_i on this realizable sample, returning its output $h_i \in \mathcal{H}_i$. Only h_i is released by the oracle; the relabeling is an internal step.

If $Q_i^S(\widetilde{h}_i) \leq \tau/2$ and h_i disagrees with $\widetilde{h}_i$ on at most a $\tau/2$ fraction of S, then replacing $\widetilde{h}_i$ by h_i in a good completion increases its error by at most $\tau/2$. Thus $Q_i^S(h_i) \leq \tau$, as required. Properness keeps the output in the component class, while the outer learner restores realizability of the global prefix before the next round and reuses the reservoirs across rounds.

Theorem 1.4 (Informal version of Theorem F.19)**.** *Let $G : \{0,1\}^k \to \{0,1\}$ and $\mathcal{C} = G(\mathcal{H}_1, \ldots, \mathcal{H}_k)$. Suppose each $\mathcal{H}_i$ has a proper private realizable PAC learner with sample requirement $m_i(\tau)$, at constant ε_i and failure and approximate-privacy parameters of order β/k and*

δ/k. *Set*

$$v_* := \max\left\{\mathrm{VC}(\mathcal{C}),\ \max_{i\in[k]}\mathrm{VC}(\mathcal{H}_i)\right\}.$$

Then $\mathcal{C}$ has a proper (ε,δ)-differentially private realizable (α,β)-PAC learner with sample complexity

$$\widetilde{O}\left(\frac{\sqrt{k}}{\varepsilon}\max\left\{\frac{kv_*}{\alpha},\ \max_{i\in[k]} m_i\Big(\frac{\alpha}{80k}\Big)\right\}\right).$$

For a common component sample bound, the multiplier on that bound improves from $\widetilde{O}(k/\varepsilon)$ in the closure construction of Alon et al. (2020) to $\widetilde{O}(\sqrt{k}/\varepsilon)$. Our dimension term is $\widetilde{O}(k^{3/2}v_*/(\varepsilon\alpha))$, using small-error generalization after each realizability repair. We remark that this comparison concerns proper component learners, while the theorem of Alon et al. (2020) also permits improper learners. Our approach may extend to improper learners as well; we focus on proper learners here to simplify the realizability and generalization arguments.

1.4 Related Work

Private threshold learning and the interior-point problem provide the one-dimensional primitives underlying private geometric learning. Following the early quasi-concave optimizer of Beimel et al. (2013) and the lower bounds of Bun et al. (2015) and Alon et al. (2019), Kaplan et al. (2020a) reduced the dependence on the domain size from exponential to polynomial in its iterated logarithm. Cohen et al. (2023) subsequently obtained nearly optimal bounds for thresholds and proved a separation from general quasi-concave optimization. The center-point and linear-feasibility constructions of Beimel et al. (2019) and Kaplan et al. (2020b) extend these primitives to halfspaces, while Nissim et al. (2026b) exploit approximation by subsamples to bypass the general quasi-concave lower bound for geometric objectives. Our halfspace results build on these constructions, with the improvements in dimension described above.[3]

A complementary line of work emphasizes computational efficiency. Nguy˜ên et al. (2020) give efficient private learners for large-margin halfspaces with sample complexity independent of the ambient dimension. More recently, Kaplan et al. (2025) develop efficient private algorithms for linear-algebraic tasks, including linear inequalities, and obtain halfspace learners as an application. Our results concern sample complexity without a margin assumption; computational efficiency remains a separate challenge.

Our general framework also belongs to the literature on transformations between private learning problems. LabelBoost was introduced by Beimel et al. (2015) to reduce the number of labeled examples needed for private learning, and its relabeling transformation also yields a reduction from agnostic learning to proper realizable learning. Alon et al. (2020) extend this reduction to improper learners and prove closure under arbitrary Boolean composition, alongside closure results for online learning. We use a more efficient version of their relabel-and-learn construction for our Boolean component oracles, and improve the cost of combining proper component learners by preserving realizability across rounds.

Specialized constructions exploit additional structure: Sadigurschi & Stemmer (2021) obtain nearly linear dependence on dimension for axis-aligned rectangles by deleting exposed records, avoiding generic composition costs. Bun & Fang (2026) give an efficient private PAC learner for decision lists with nearly non-private sample complexity, and a private Winnow algorithm for online learning of large-margin halfspaces. Our framework complements these specialized constructions by studying how realizability and data reuse can reduce the sample cost of sequential component choices.

1.5 Open questions

We have given a framework for private component-by-component learning that restores realizability after each component choice and shares samples across rounds with a $\widetilde{O}(\sqrt{k}/\varepsilon)$ overhead. Its applications improve the dependence on the dimension for halfspaces and on the number of components for Boolean composition.

[3] Huang et al. (2025) claim a stronger sample bound for halfspaces, but Nissim et al. (2026b) give a counterexample to the privacy guarantee of their proposed algorithm.

Our improved upper bounds for halfspaces still exceed the known lower bounds: $\Omega(d \log^* L)$ for proper learners (Bun et al., 2015), and $\Omega(d + \log^* L)$ for learners that may be improper (Alon et al., 2019). Closing these gaps, particularly the polynomial gap in d, would clarify how much of the current overhead is inherent to privacy.

Furthermore, our construction is primarily information-theoretic: selecting representatives of labelings, searching finite coordinate domains, and finding good completions may require superpolynomial time. Kaplan et al. (2025) give efficient private algorithms for linear-algebraic problems and, as an application, halfspace learning, with sample complexity polynomial in the dimension and polylogarithmic in the grid size L. It remains open whether efficient halfspace learners can achieve the much weaker $\log^* L$ dependence while retaining a small dependence on d. More generally, it would be useful to identify component representations and relabeling procedures for which our framework can be implemented efficiently.

1.6 Paper Organization

We give an overview of our main techniques in Section 2. Appendix A collects the technical preliminaries, including the properties of LabelBoost used in our analysis. In Appendix B, we formalize component-by-component learning and the requirements on the component oracles. Appendix C presents our generic CarryForwardLearner algorithm and its main guarantee, with the privacy and utility analyses in Appendices D and E, respectively. In Appendix F, we instantiate the framework for halfspace learning and closure under Boolean composition. Additional proofs appear in Appendices G to I.

2 Our Technique

In this section, we give a proof overview of our main result, Theorem 1.1.

2.1 The Setting

Let $\mathcal{C}$ be a concept class of VC dimension v, whose classifiers have the form $c_{y_1,\dots,y_k} : \mathcal{X} \to \{0,1\}$, with components $y_i \in \mathcal{Y}_i$. For a prefix $y_{<i} = (y_1, \dots, y_{i-1})$, let $\mathcal{C}(y_{<i})$ denote the concepts whose representations extend that prefix. We are given component oracles $A_1, \dots, A_k$. On input a valid prefix $y_{<i}$ and a labeled sample $S \in \mathcal{Z}^m$, where $\mathcal{Z} = \mathcal{X} \times \{0,1\}$, oracle A_i outputs a valid next component y_i.

Our goal is to combine these oracles into a proper (ε, δ)-differentially private learner for $\mathcal{C}$. Given i.i.d. examples from a distribution realizable by $\mathcal{C}$, the learner should output a classifier $c_{y_1,\dots,y_k}$ with population error at most α, with probability at least $1 - \beta$. Its privacy guarantee must hold on arbitrary input datasets. We seek to achieve this using only a $\widetilde{O}(\sqrt{k}/\varepsilon)$ overhead relative to the sample size needed for one component step and the accompanying generalization guarantee.

The oracle's privacy and accuracy guarantees are required only when S is realizable by $\mathcal{C}(y_{<i})$. More precisely, for every fixed prefix, $(\varepsilon_A, \delta_A)$-privacy holds between neighboring samples that are both realizable by this restricted class, possibly by different hypotheses (Definition A.6). For accuracy, with high probability the extended class $\mathcal{C}(y_{\le i})$ contains a completion with empirical error at most τ on S. The oracle need not identify that completion. These requirements are formalized in Assumption B.4.

A direct and naive attempt would run the oracles sequentially on the original sample. However, even when the sample is initially realizable, an approximately accurate component choice can leave no consistent completion of the selected prefix. The next oracle then receives an input outside its promised domain, so neither its privacy nor its utility guarantee applies. We must therefore restore exact realizability after each component choice, while controlling the error introduced by doing so.

2.2 Restoring Realizability via LabelBoost

We use the LabelBoost procedure of Beimel et al. (2015). Given a labeled sample S, an unlabeled sample T, and a concept class $\mathcal{C}$, the procedure considers the labelings of $U(S) \cup T$ induced by

$\mathcal{C}$, keeping one representative hypothesis for each labeling. It selects a representative using the exponential mechanism, giving greater weight to hypotheses that agree with more of the labels in S, and relabels both samples according to it. Here $U(S)$ denotes the sample with its labels removed, and $U(T)^h$ denotes T labeled by h. The resulting sample is exactly realizable by $\mathcal{C}$, even if the original sample was not. If $\mathcal{C}$ contains a hypothesis of small empirical error, LabelBoost introduces only a small additional error.

The selected hypothesis need not itself be private: its candidate set depends on the input points. Nevertheless, Beimel et al. (2015) show that LabelBoost followed by an (ε, δ')-private mechanism on the relabeled data is $(\varepsilon + 3, 4e\delta')$-private (Lemma A.11). Thus, a continuation with constant privacy cost gives constant privacy for the combined procedure; subsampling from a larger dataset can reduce this to a small privacy parameter. Our extended analysis also allows the continuation's privacy guarantee to hold only on realizable inputs (Lemma A.15).

The challenge is to perform k such relabel-and-learn steps without allocating a new large dataset to every step. In their construction for Boolean composition, Alon et al. (2020) use a separate sample for each component. When the component sample requirements are comparable, this incurs a factor of k. We instead reuse two reservoirs, with a privacy analysis that accounts for the intermediate relabeled samples passed between rounds.

2.3 Our CarryForwardLearner Algorithm

CarryForwardLearner (Algorithm 2) partitions its n input records, independently of their values, into three blocks of comparable size: an initialization block W^{init} and two reservoirs $W^{(0)}, W^{(1)}$. The active sample size is

$$m = \Theta\left(\frac{\varepsilon n}{\sqrt{k \log(2/\delta)}}\right),$$

with a sufficiently small constant of proportionality. To initialize, we draw a size-m subsample R_0 from W^{init} and apply LabelBoost with class $\mathcal{C}$. If $h_0 \in \mathcal{C}$ is the internally selected hypothesis, we set $S_0 = U(R_0)^{h_0}$. This preliminary relabeling ensures that even an arbitrary input dataset yields a realizable input to A_1.

At iteration i, the learner has a prefix $y_{<i}$ and an active sample S_{i-1} labeled by some $h_{i-1} \in \mathcal{C}(y_{<i})$. It proceeds as follows:

1. Compute $y_i \leftarrow A_i(y_{<i}; S_{i-1})$ and extend the prefix to $y_{\le i}$. If $i = k$, return $c_{y_1,\ldots,y_k}$.
2. Sample a fresh size-m subset $T_i \subseteq W^{((i-1) \bmod 2)}$ uniformly without replacement. After the first iteration, this is the reservoir opposite to the one carrying the current active sample.
3. Apply LabelBoost to the labeled sample S_{i-1} and the unlabeled sample $U(T_i)$, using the extended class $\mathcal{C}(y_{\le i})$. Let $h_i \in \mathcal{C}(y_{\le i})$ be its internally selected hypothesis.
4. Set $S_i = U(T_i)^{h_i}$ and continue to the next iteration.

The reservoirs are kept unchanged, so subsets sampled in different rounds may overlap. Only the final classifier is released. Note that by construction, S_i is exactly realizable by $\mathcal{C}(y_{\le i})$ in every execution, regardless of whether any preceding oracle met its accuracy guarantee (Lemma C.2).

Privacy analysis sketch. We use a constant oracle privacy parameter ε_A.

We first consider a single round with a fixed incoming prefix $y_{<i}$ and a fixed next carrier T_i. The round first computes $y_i \leftarrow A_i(y_{<i}; S_{i-1})$. By Assumption B.4, this oracle call is guaranteed to be $(\varepsilon_A, \delta_A)$-private only between neighboring samples that are both realizable by $\mathcal{C}(y_{<i})$, possibly by different hypotheses. This is the source of the realizability restriction in the privacy guarantee for the entire round. For every fixed output y_i, our carrier-only LabelBoost lemma (Lemma A.17) shows that releasing the labeling $S_i = U(T_i)^{h_i}$ of the fixed carrier costs only $c_{\text{LB}} = \log(4e)$ in privacy, even on arbitrary neighboring labeled inputs. Composing this relabeling step with the oracle call therefore gives $(\varepsilon_A + c_{\text{LB}}, \delta_A)$-privacy for (y_i, S_i) on the same realizable neighboring inputs (Claim D.3). This argument concerns changes in S_{i-1} with T_i fixed; it does not allow us to release S_i privately with respect to changes in T_i, since S_i retains its underlying records.

To handle that dependence, we group rounds i and $i+1$ into a *two-round macro-block*. For a fixed incoming state $(y_{<i}, S_{i-1})$ and fixed reservoir $W^{(i \bmod 2)}$, this mechanism maps T_i (chosen from the opposite reservoir $W^{((i-1) \bmod 2)}$) to (y_i, y_{i+1}, S_{i+1}). The first component y_i is computed before T_i is accessed. LabelBoost then relabels T_i, setting $S_i = U(T_i)^{h_i}$, and the second round applies the private continuation (the single round computation described above) to this relabeled sample S_i to obtain (y_{i+1}, S_{i+1}). Namely, the map from T_i to (y_{i+1}, S_{i+1}) is obtained by applying to the relabeled T_i a computation that is $(\varepsilon_A + c_{\mathrm{LB}}, \delta_A)$-differentially private on $\mathcal{C}(y_{\le i})$-realizable neighboring inputs. Our realizable-input extension of the LabelBoost lemma (Lemma A.15) therefore gives $(O(1), O(\delta_A))$-privacy for the macro-block with respect to arbitrary neighboring T_i, T_i' (Lemma D.2). At its end, the active sample is carried by the opposite reservoir: the records of T_i are no longer passed forward.

For each reservoir, it remains to compose roughly $k/2$ macro-blocks. Each block passes its chosen components and active sample to the next, and the privacy guarantee above holds for every possible incoming state. Adaptive composition therefore applies (Lemma D.9). Before composing, we amplify the privacy of each block: it accesses the reservoir through a random subsample containing a fraction

$$q = \frac{m}{|W^{(b)}|} = O\left(\frac{\varepsilon}{\sqrt{k \log(2/\delta)}}\right)$$

of the reservoir. Subsampling amplification reduces the constant privacy cost of its macro-block to $O(q)$, with approximate-privacy parameter $O(q\delta_A)$. Taking δ_A of order δ/k and applying advanced composition over at most $\lceil (k-1)/2 \rceil$ activations gives (ε, δ)-privacy for that reservoir. The initialization block is accessed only once and is handled by the same LabelBoost and amplification argument. Since a neighboring input changes one record in exactly one block, the complete learner is private on arbitrary input datasets. The formal analysis is in Appendix D.

Utility analysis sketch. Suppose the input is drawn independently from a distribution realizable by $h^\star \in \mathcal{C}$. At the start of round i, the active sample is labeled by h_{i-1}. With high probability, oracle A_i leaves a completion $g_i \in \mathcal{C}(y_{\le i})$ that disagrees with h_{i-1} on at most a τ fraction of that sample. For $i < k$, LabelBoost then selects $h_i \in \mathcal{C}(y_{\le i})$ with at most another τ of empirical error. Consequently,

$$\mathrm{disc}_{U(S_{i-1})}(h_i, h_{i-1}) \le 2\tau$$

on the success event, where disc denotes the fraction of points on which two hypotheses disagree (Corollary A.13). Thus, restoring realizability costs only $O(\tau)$ empirical disagreement per round.

The second step is to transfer this bound to the population, despite the adaptive choice of both hypotheses. We use a small-error VC guarantee that holds simultaneously for every pair in $\mathcal{C}$: for a sufficiently large sample, empirical disagreement at most 2τ implies population disagreement at most 20τ (Theorem A.9). The underlying points of each active sample are marginally an i.i.d. sample from the population, because the sampled indices are chosen independently of record values. Different rounds need not be independent. The uniform guarantee applies to the adaptively chosen hypotheses, and a union bound over rounds handles reuse of the reservoirs. Both this guarantee and LabelBoost's utility bound require only $\widetilde{O}(v/\tau)$ examples. With $\tau = \Theta(\alpha/k)$, this gives the $\widetilde{O}(kv/\alpha)$ requirement on each active sample. The active sample must also meet the oracle's sample requirement $M_A(\tau)$.

Initialization similarly makes h_0 close to $h^\star$. In the final round, the full prefix has a unique completion $h_k = c_{y_1,\ldots,y_k}$, so oracle accuracy and generalization directly bound its disagreement with h_{k-1}. Summing the population disagreements along $h^\star, h_0, \ldots, h_k$ gives error $O(k\tau)$. In particular, $\tau = \alpha/(40k)$ and failure probabilities of order β/k suffice for total error at most α with probability at least $1-\beta$ (Theorem E.2). Combining the active-sample requirement with the privacy calibration yields

$$n = \widetilde{O}\left(\frac{\sqrt{k}}{\varepsilon} \max\left\{\frac{kv}{\alpha},\ M_A\Big(\frac{\alpha}{40k}\Big)\right\}\right),$$

with the same parameter conventions as in Theorem 1.1.

Throughout both analyses, realizability of every oracle input is an unconditional invariant, rather than a high-probability utility event. This is why privacy only on realizable neighbors suffices for the component oracles, while the resulting learner provides standard differential privacy on all inputs.

AI USE STATEMENT

In this work, we used generative AI through OpenAI's ChatGPT and Codex to discuss and check mathematical arguments and to draft and revise parts of the proofs. It was also used for literature searches and comparisons, manuscript editing, and the preparation and verification of the LATEX source.

More importantly, AI helped develop CARRYFORWARDLEARNER (Algorithm 2) and prove its privacy guarantee. This assistance came after extensive interaction, starting with an initial incomplete version of the algorithm provided by the authors. The authors developed the abstraction of the generic component-by-component learner and its connection to learning halfspaces and closure under Boolean composition; in these parts, AI assistance was limited to lower-level tasks.

AI-assisted suggestions were discussed and revised during manuscript development; they do not replace verification of the mathematical arguments and cited results. The authors are responsible for the correctness, originality, and final content of this work.

ETHICS STATEMENT

This work studies theoretical guarantees for differentially private learning. It reports no human-subject experiments or empirical analysis of personal data. The results aim to reduce the data requirements of private learning; their privacy guarantees depend on the stated assumptions, adjacency relation, and privacy parameters.

REPRODUCIBILITY STATEMENT

All results are theoretical. The oracle assumptions and generic learner are specified in Assumption B.4 and Algorithm 2, with privacy and utility proofs in Appendices D and E. Component-oracle constructions and supporting proofs appear in Appendices F to I.

REFERENCES

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019*, pp. 852–860, 2019.

Noga Alon, Amos Beimel, Shay Moran, and Uri Stemmer. Closure properties for private classification and online prediction. In *Proceedings of the 33rd Conference on Learning Theory, COLT 2020*, volume 125, pp. 119–152, 2020.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *APPROX-RANDOM*, volume 8096, pp. 363–378, 2013. Journal version: *Theory of Computing*, 12(1):1–61, 2016.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Learning privately with labeled and unlabeled examples. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015*, pp. 461–477, 2015. arXiv:1407.2662v3.

Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. Private center points and learning of halfspaces. In *COLT*, Proceedings of Machine Learning Research, pp. 269–282. PMLR, 2019.

Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Mark Bun and William Fang. Privately learning decision lists and a differentially private winnow. In *Proceedings of The 37th International Conference on Algorithmic Learning Theory*, volume 313, pp. 1–27, 2026.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, pp. 634–649, 2015.

Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 389–402, 2020.

Edith Cohen, Xin Lyu, Jelani Nelson, Tamás Sarlós, and Uri Stemmer. Optimal differentially private learning of thresholds and quasi-concave optimization. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023*, 2023. Full version: arXiv:2211.06387.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pp. 265–284, 2006.

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2010.

Yi Hsiang Huang, Wei-Hong Chen, and Shi-Chun Tsai. On the sample complexity of privately learning half-spaces. In *Proceedings of the 16th Asian Conference on Machine Learning, ACML*, volume 260, pp. 655–670, 2025.

Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. Privately learning thresholds: Closing the exponential gap. In *Proceedings of Thirty Third Conference on Learning Theory, COLT 2020*, volume 125, pp. 2263–2285, 2020a.

Haim Kaplan, Yishay Mansour, Uri Stemmer, and Eliad Tsfadia. Private learning of halfspaces: Simplifying the construction and reducing the sample complexity. In *Advances in Neural Information Processing Systems 33, NeurIPS 2020*, volume 33, 2020b.

Haim Kaplan, Yishay Mansour, Shay Moran, Uri Stemmer, and Nitzan Tur. On differentially private linear algebra. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pp. 2362–2373, 2025.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Huy Lê Nguy˜ên, Jonathan Ullman, and Lydia Zakynthinou. Efficient Private Algorithms for Learning Large-Margin Halfspaces. In *Proceedings of the 31st International Conference on Algorithmic Learning Theory, ALT 2020*, volume 117, pp. 704–724, 2020.

Kobbi Nissim, Uri Stemmer, and Eliad Tsfadia. Invited open problem: Does differential privacy make pac learning much harder? In *Proceedings of Thirty Ninth Conference on Learning Theory, COLT 2026*, volume 336, pp. 7129–7135, 2026a.

Kobbi Nissim, Eliad Tsfadia, and Chao Yan. Differentially private quasi-concave optimization: Bypassing the lower bound and application to geometric problems. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2026*, pp. 5824–5842, 2026b.

Menachem Sadigurschi and Uri Stemmer. On the sample complexity of privately learning axis-aligned rectangles. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28286–28297, 2021.

N Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1): 145–147, 1972.

A PRELIMINARIES

A.1 NOTATION AND PAC LEARNING

Let $\mathcal{X}$ be an instance space, let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ be a binary concept class, and write $\mathcal{Z} := \mathcal{X} \times \{0,1\}$. Samples are finite sequences, with repetitions allowed and counted with multiplicity. We write $\circ$ for concatenation.

For a labeled sample $S = ((x_i, y_i))_{i=1}^n \in \mathcal{Z}^n$, its unlabeled projection is $U(S) := (x_i)_{i=1}^n$. For an unlabeled sample $U(S) = (x_i)_{i=1}^n$ and a hypothesis $h : \mathcal{X} \to \{0,1\}$, let $U(S)^h := ((x_i, h(x_i)))_{i=1}^n$, Thus, $U(S)^h$ replaces the labels of S by those assigned by h. For partially labeled samples, the same notation applies after discarding the existing labels.

For a distribution $\mathcal{D}$ over $\mathcal{Z}$, let $\mathcal{D}_X$ denote its marginal over $\mathcal{X}$. We write $S \sim \mathcal{D}^n$ and $U \sim \mathcal{D}_X^n$ for n i.i.d. draws from the respective distributions.

For hypotheses $h, h' : \mathcal{X} \to \{0,1\}$, define

$$\mathrm{err}_S(h) := \frac{1}{n}\sum_{i=1}^n \mathbf{1}\{h(x_i) \neq y_i\}, \qquad \mathrm{err}_{\mathcal{D}}(h) := \Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y],$$

$$\mathrm{disc}_{U(S)}(h,h') := \frac{1}{n}\sum_{i=1}^n \mathbf{1}\{h(x_i) \neq h'(x_i)\}, \quad \mathrm{disc}_{\mathcal{D}_X}(h,h') := \Pr_{x\sim\mathcal{D}_X}[h(x) \neq h'(x)].$$

We say that h is consistent with S if $\mathrm{err}_S(h) = 0$. The sample S is $\mathcal{C}$-realizable if some $h \in \mathcal{C}$ is consistent with S. Similarly, $\mathcal{D}$ is $\mathcal{C}$-realizable if $\mathrm{err}_{\mathcal{D}}(h) = 0$ for some $h \in \mathcal{C}$. If h' is consistent with S, then $\mathrm{err}_S(h) = \mathrm{disc}_{U(S)}(h,h')$. Likewise, if $\mathrm{err}_{\mathcal{D}}(h') = 0$, then $\mathrm{err}_{\mathcal{D}}(h) = \mathrm{disc}_{\mathcal{D}_X}(h,h')$.

Definition A.1 (Realizable PAC learning)**.** Let $\alpha, \beta \in (0,1)$. An algorithm $\mathcal{A}$ is an (α, β, n)-PAC learner for $\mathcal{C}$ if, for every $\mathcal{C}$-realizable distribution $\mathcal{D}$ over $\mathcal{Z}$, given $S \sim \mathcal{D}^n$, it outputs a hypothesis $h : \mathcal{X} \to \{0,1\}$ satisfying

$$\Pr[\mathrm{err}_{\mathcal{D}}(h) \leq \alpha] \geq 1 - \beta.$$

The probability is over the sample and the internal randomness of $\mathcal{A}$. The learner is *proper* if it always outputs $h \in \mathcal{C}$; otherwise, it is *improper*.

A.2 DIFFERENTIAL PRIVACY

We use substitute-one adjacency on fixed-size datasets: $S, S' \in \mathcal{Z}^n$ are neighboring, written $S \simeq S'$, if they differ in one indexed position.

Definition A.2 (Differential privacy)**.** A randomized mechanism M is (ε, δ)-differentially private if, for all neighboring datasets $S \simeq S'$ and all measurable output events E,

$$\Pr[M(S) \in E] \leq e^{\varepsilon} \Pr[M(S') \in E] + \delta.$$

Two random variables X and Y are (ε, δ)-indistinguishable if, for every measurable event E, $\Pr[X \in E] \leq e^{\varepsilon} \Pr[Y \in E] + \delta$, the same inequality holds with X and Y interchanged.

Thus, differential privacy requires $M(S)$ and $M(S')$ to be (ε, δ)-indistinguishable for every neighboring S, S'.

Definition A.3 (Private PAC learning (Kasiviswanathan et al., 2011))**.** Let $\mathcal{A}$ be an algorithm that receives as input a sample $S = (z_1, \ldots, z_n)$. We say that $\mathcal{A}$ is $(\alpha, \beta, \varepsilon, \delta, n)$-differentially private PAC learner for a concept class $\mathcal{C}$ over $\mathcal{X}$, using hypothesis class $\mathcal{H}$ and sample complexity m, if:

- **Privacy.** $\mathcal{A}$ is (ε, δ)-differentially private. (Definition A.2)
- **Utility.** $\mathcal{A}$ is an (α, β, n)-PAC learner for $\mathcal{C}$ using hypothesis class $\mathcal{H}$. (Definition A.1)

We call $\mathcal{A}$ a *realizable private PAC learner* when its utility guarantee is required only for distributions realizable by $\mathcal{C}$.

Theorem A.4 (Fixed-size subsampling amplification)**.** *(Balle et al., 2018, Theorem 9) Let a mechanism on a size-m dataset be (ε, δ)-DP under substitute-one adjacency. If its input is obtained by*

uniformly sampling m records without replacement from a size-M reservoir, with $q = m/M$, then the induced reservoir-level mechanism is

$$\big(\log\big(1 + q(e^{\varepsilon} - 1)\big), q\delta\big)\text{-DP}.$$

Theorem A.5 (Adaptive advanced composition)**.** *(Dwork et al., 2010, Definition III.1 and Theorem III.3) Consider an adaptive sequence of L mechanisms. In each round, the mechanism to be applied may depend on the outputs of the preceding rounds. Suppose that, for every round and every fixed realization of the preceding transcript, the mechanism selected for that round is (ε, δ)-differentially private. Then, for every $\delta' > 0$, the full composition is $(\varepsilon_{\mathrm{tot}}, L\delta + \delta')$-differentially private, where*

$$\varepsilon_{\mathrm{tot}} \le \sqrt{2L\log(1/\delta')}\,\varepsilon + L\varepsilon(e^{\varepsilon} - 1).$$

We also use the standard post-processing property and basic composition without further restatement.

Definition A.6 (Differential privacy on realizable inputs)**.** Fix a concept class $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$. A labeled sample $S = ((x_i, y_i))_{i=1}^{m}$ is $\mathcal{C}$-realizable if there exists $g \in \mathcal{C}$ such that $g(x_i) = y_i$ for every $i \in [m]$. A randomized mechanism

$$A : (\mathcal{X} \times \{0,1\})^m \to \mathcal{R}$$

is (ε, δ)-differentially private $\mathcal{C}$-realizable inputs if, for every pair of neighboring samples S, S' that are both $\mathcal{C}$-realizable, possibly by different hypotheses in $\mathcal{C}$, and every event $F \subseteq \mathcal{R}$,

$$\Pr[A(S) \in F] \le e^{\varepsilon}\Pr[A(S') \in F] + \delta.$$

A.3 Learning-Theoretic Tools

Definition A.7 (Trace, growth function, and VC dimension)**.** Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ and let $P = \{p_1, \ldots, p_r\} \subseteq \mathcal{X}$ be finite. The trace of $\mathcal{C}$ on P is

$$\Pi_{\mathcal{C}}(P) := \{(g(p_1), \ldots, g(p_r)) : C \in \mathcal{C}\}.$$

Thus, $\Pi_{\mathcal{C}}(P)$ is the set of all dichotomies induced by $\mathcal{C}$ on P.

The growth function of $\mathcal{C}$ is defined by

$$\Pi_{\mathcal{C}}(r) := \max_{P \subseteq \mathcal{X}:\, |P| = r} |\Pi_{\mathcal{C}}(P)|.$$

The VC dimension of $\mathcal{C}$, denoted $\mathrm{VC}(\mathcal{C})$, is the largest r for which there exists a set $P \subseteq \mathcal{X}$ of size r such that $|\Pi_{\mathcal{C}}(P)| = 2^r$.

Lemma A.8 (Sauer's lemma)**.** *(Sauer, 1972) We used Sauer's lemma in the form*

$$|\Pi_{\mathcal{C}}(P)| \le \left(\frac{er}{\mathrm{VC}(\mathcal{C})}\right)^{\mathrm{VC}(\mathcal{C})}$$

whenever $r > \mathrm{VC}(\mathcal{C})$

Theorem A.9 (Small-error all-pairs generalization)**.** *Beimel et al. (2015, Theorem B.5)*

Let C and $\mathcal{D}$ be a concept class and a distribution over a domain X. Let $\alpha, \beta > 0$, and $m \ge \frac{48}{\alpha}\left(10\,\mathrm{VC}(C)\log\left(\frac{48e}{\alpha}\right) + \log\left(\frac{5}{\beta}\right)\right)$. Suppose that we draw a sample $S = (x_i)_{i=1}^{m}$, where each x_i is drawn i.i.d. from μ. Then,

$$\Pr\left[\begin{array}{c}\exists\, c, h \in C \text{ s.t. } \mathrm{disc}_{\mathcal{D}_X}(c,h) \ge \alpha \\ \text{and } \mathrm{disc}_S(c,h) \le \alpha/10\end{array}\right] \le \beta.$$

Equivalently, with probability at least $1 - \beta$, simultaneously for every $h, g \in \mathcal{C}$,

$$\mathrm{disc}_S(h,g) \le \frac{\alpha}{10} \quad\Longrightarrow\quad \mathrm{disc}_{\mathcal{D}_X}(h,g) < \alpha.$$

Theorem A.10 (Realizable VC generalization bound). *Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ be a binary hypothesis class with $\mathrm{VC}(\mathcal{C}) \leq v$. Fix an unlabeled samples multiset S and let $B \subseteq S$ be a uniformly random subset of size m. For $\tau, \beta \in (0,1)$, There exists a universal constant $c > 0$ such that, if $|S| \geq m$ and*

$$m \geq c\frac{v\log(1/\tau) + \log(2/\beta)}{\tau}$$

Then, with probability at least $1-\beta$, simultaneously for every $h \in \mathcal{C}$,

$$\mathrm{disc}_B(h, h^\star) = 0 \quad \Longrightarrow \quad \mathrm{disc}_S(h, h^\star) \leq \tau.$$

Proof. The claim follows from Theorem A2.2 of Blumer et al. (1989), applied to the uniform distribution over S. Although their theorem is stated for independent draws with replacement, the same guarantee holds for a uniformly random subset of size r. Indeed, couple the independent sample with a uniformly random r-subset B containing all distinct sampled points. Every hypothesis consistent with B is also consistent with the independent sample, and the result follows. □

A.4 LABELBOOST

We use LABELBOOSTPROCEDURE of Beimel et al. (2015, Algorithm 2). Given a labeled sample S, an unlabeled sample T, and a nonempty concept class $\mathcal{C}$, LABELBOOSTPROCEDURE constructs one representative for every labeling of $U(S) \cup T$, then selects one representative hypothesis from $\mathcal{C}$ using the exponential mechanism with score equal to its agreement with S, and relabels $S \cup T$ according to the selected hypothesis. We use LABELBOOSTPROCEDURE with $D = \emptyset$ and score $q(S, g) = \sum_{j \in [|S|]} \mathbf{1}[g(x_j) = y_j]$

we will conduct extended privacy analysis of labelboost.

Algorithm 1 LabelBoostProcedure

Input: A partially labeled database $B = S \circ T \in (X \times \{0,1,\perp\})^*$, and a concept class $\mathcal{C} \subseteq \{0,1\}^X$.

1: Initialize $H = \emptyset$.
2: Let $P = \{p_1, \ldots, p_k\}$ be the set of all points $p \in X$ appearing at least once in $S \circ T$.
3: For every $(z_1, \ldots, z_k) \in \Pi_{\mathcal{C}}(P)$, add to H an arbitrary $g \in \mathcal{C}$ such that $g(p_i) = z_i$ for every $1 \leq i \leq k$.
4: Choose $h \in H$ using the exponential mechanism with privacy parameter $\varepsilon = 1$, solution set H, and database S.
5: Relabel $S \circ T$ using h, and denote the resulting labeled database by $(S \circ T)^h$.
6: **return** $(S \circ T)^h$.

Lemma A.11 (LabelBoost privacy (Beimel et al., 2015, Lemma 4.1)). *Let A be an (ε, δ)-differentially private algorithm. Then the composition obtained by applying* LABELBOOSTPROCEDURE *and subsequently running A on the relabeled sample is*

$$(\varepsilon + 3,\ 4e\delta)\text{-differentially private}.$$

Proposition A.12 (Exponential-mechanism utility (Beimel et al., 2015, Proposition A.1)). *Let F be a finite hypothesis class and let $\alpha := \min_{f \in F} \mathrm{err}_S(f)$. If h is selected from F by the exponential mechanism with privacy parameter ε and score equal to empirical agreement with S, then for every $\lambda > 0$,*

$$\Pr[\mathrm{err}_S(h) > \alpha + \lambda] \leq |F| \exp\left(-\frac{\varepsilon\lambda|S|}{2}\right).$$

Corollary A.13 (Approximate utility of LABELBOOSTPROCEDURE). *Let $\mathcal{C}$ be a concept class with $\mathrm{VC}(\mathcal{C}) \leq v$, and let $|S| = |T| = m$. Suppose that there exists $g \in \mathcal{C}$ such that $\mathrm{err}_S(g) \leq \alpha$. Let h be the hypothesis selected internally by* LABELBOOSTPROCEDURE$(S, T, \mathcal{C})$. *For a sufficiently large universal constant c, with a constant $\varepsilon = 1$ if*

$$m \geq \frac{cv}{\alpha}\log\left(\frac{2}{\alpha\beta}\right),$$

then, with probability at least $1-\beta$,

$$\mathrm{err}_S(h) \leq 2\alpha.$$

Proof. Apply the proof of Beimel et al. (2015, Claim 4.2), replacing its zero-error representative by the representative of g, whose empirical error is at most α. By Proposition A.12, except with probability β, the selected hypothesis incurs at most an additional α empirical error. Hence $\mathrm{err}_S(h) \leq 2\alpha$. The final bound on m implies the preceding displayed condition. □

A.5 Extended Privacy Analysis

This section refines the analysis of LabelBoostProcedure in a way that allows us, later on, to analyze algorithms that explicitly exploit the realizability of the current sample.Beimel et al. (2015) already proved that LabelBoost followed by $(\varepsilon, \delta) - DP$ algorithm is $(\varepsilon + 3, 4e\delta) - DP$. We want to offer a more precise analysis. Consider neighboring inputs:

$$B_1 = S_1 \circ T_1 \qquad \text{and} \qquad B_2 = S_2 \circ T_2$$

Let P_i denote the set of distinct points appearing in B_i, and let $K := P_1 \cap P_2$. By definition A.7, $\Pi_{\mathcal{C}}(K)$ is the set of all labelings for the common points K induced by hypotheses in $\mathcal{C}$. Following Beimel et al. (2015, Lemma 4.1), for every labeling $z \in \Pi_{\mathcal{C}}(K)$, define the weight $w_i(z)$ as the total probability assigned by the exponential mechanism to choose a representative hypothesis h_i which induces labeling z on K. Thus, $w_1(z)$ and $w_2(z)$ are the probability masses corresponding to the same common labeling z in the executions on B_1 and B_2. The following weight comparison is established in the proof of Beimel et al. (2015, Lemma 4.1).

Lemma A.14 (LabelBoost weight comparison)**.** *For every $z \in \Pi_{\mathcal{C}}(K)$,*

$$w_1(z) \leq 4e\, w_2(z) \qquad \textit{and} \qquad w_2(z) \leq 4e\, w_1(z).$$

For convenience, define $c_{\mathrm{LB}} := \ln(4e)$.

notice that for a fixed $z \in \Pi_{\mathcal{C}}(K)$ we can define the associated hypothesis by the labeling z as $h|z$, by the definition of LabelBoostProcedure there are one or two hypothesis that are associated with z and every labeling of any subset of K by $h|z$ an hypothesis associated with z is the same.

Lemma A.15 (LabelBoost with a realizable-private continuation)**.** *Fix a concept class $\mathcal{C}$. Suppose A is (ε, δ)-differentially private on $\mathcal{C}$-realizable inputs (Definition A.6). Define the mechanism $\mathcal{B}$ as follows: On input (S, T), run* LabelBoostProcedure *with concept class $\mathcal{C}$, let h be the selected hypothesis, and output $A(S^h \circ T^h)$Then $\mathcal{B}$ is*

$$(\varepsilon + c_{LB},\ 4e\delta)\textit{-differentially private}$$

on arbitrary neighboring inputs.

Remark A.16. Lemma A.15 also applies when the continuation uses only T^h. Indeed, define $\widetilde{A}(S^h, T^h) := A(T^h)$. Projecting neighboring pairs onto their second component produces equal or neighboring samples, and T^h remains realizable by $h \in \mathcal{C}$. Hence $\widetilde{A}$ has the same realizable-input privacy guarantee.

Proof. Consider neighboring inputs $B_1 = (S_1 \circ T_1)$, $B_2 = (S_2 \circ T_2)$, and denote K as the unlabeled common set. For every $z \in \Pi_{\mathcal{C}}(K)$, all representatives associated with z in either execution belong to $\mathcal{C}$. Hence every relabeled input B_i^h passed to A is realizable by $\mathcal{C}$. Moreover, for each fixed z, every relabeled samples from the first execution is equal or neighboring to every corresponding relabeled sample from the second execution, In particular assume that $B_1 = K \cup \{x\}$, $B_2 = K \cup \{x'\}$ for a fixed z any labeling according to $h|z$ has the same labels on K and we can conclude that $B_1^{h_1|z}$, $B_2^{h_2|z}$ has the same labeling on K therefore are neighboring for every $h_i|z$ Consequently, for every event F, and since B_i^h is realizable by h that associated with z,

$$\begin{aligned}
\Pr[\mathcal{B}(B_1) \in F] &= \sum_{z \in \Pi_{\mathcal{C}}(K)} w_1(z) \Pr\big[A(B_1^h) \in F \,\big|\, h|_K = z\big] \\
&\leq 4e \sum_{z \in \Pi_{\mathcal{C}}(K)} w_2(z) \left(e^{\varepsilon} \Pr\big[A(B_2^h) \in F \,\big|\, h|_K = z\big] + \delta\right) \\
&= 4e\, e^{\varepsilon} \Pr[\mathcal{B}(B_2) \in F] + 4e\delta \\
&\leq e^{\varepsilon + c_{LB}} \Pr[\mathcal{B}(B_2) \in F] + 4e\delta,
\end{aligned}$$

where the first inequality holds by Lemma A.14 and by the assumption on A. The reverse direction follows by interchanging B_1 and B_2. □

Lemma A.17 (Carrier-only LabelBoost release)**.** *Fix a concept class $\mathcal{C}$ and an unlabeled carrier T. Define the mechanism M_T as follows: on input a labeled sample S, compute*

$$T^h \leftarrow \textsc{LabelBoostProcedure}(S, T, \mathcal{C})$$

and output T^h. Then, for every fixed unlabeled carrier T, the mechanism M_T is $(c_{LB}, 0)$-differentially private with respect to its input sample S.

Proof. Let T be a fix unlabeled carrier, let S_1, S_2 be a neighboring labeled samples. and let h denote the selected hypothesis such that M_T outputs T^h. Let $B_1 = S_1 \circ T$ and $B_2 = S_2 \circ T$, Apply Lemma A.14 to the neighboring inputs B_1, B_2 Since the same carrier T appears in both inputs, every point appearing in T belongs to the common set K.

Fix $z \in \Pi_{\mathcal{C}}(K)$. Every representative hypothesis associated with z induce exactly the same labeling on T. Hence, for a fixed z, the labeling of T is determined and fixed in both executions.

Let F be any event in the space of labeled versions of T, let

$$\mathcal{Z}_F := \{z \in \Pi_{\mathcal{C}}(K) : T^z \in F\}.$$

Let $T^{h_i} \leftarrow M_T(S_i)$ then

$$\Pr[T^{h_1} \in F] = \sum_{z \in \mathcal{Z}_F} w_1(z).$$

By Lemma A.14,

$$\Pr[T^{h_1} \in F] \le 4e \sum_{z \in \mathcal{Z}_F} w_2(z) = 4e \Pr[T^{h_2} \in F].$$

Since $4e = e^{c_{\mathrm{LB}}}$,

$$\Pr[T^{h_1} \in F] \le e^{c_{\mathrm{LB}}} \Pr[T^{h_2} \in F].$$

The reverse inequality follows symmetrically, Hence

$$\Pr[M_T(S_1) \in F] \le e^{c_{\mathrm{LB}}} \Pr[M_T(S_2) \in F].$$

Therefore the carrier-only release M_T is $(c_{\mathrm{LB}}, 0)$-differentially private. □

B Component-by-Component Learning Framework

In this section, we present a learner that constructs a hypothesis one component at a time. To capture this process, we introduce an abstract component oracle for each step. The oracle receives the components already chosen by the learner together with the current labeled sample, and selects the next component. Its utility guarantee requires that, with high probability, the selected components can still be completed to a hypothesis in the full concept class that closely approximates the current sample. The following definitions formalize this framework.

B.1 Component Representations

Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ be a concept class with a k-component parameterization. Let $\mathcal{Y}_{\le k}$ denote the set of valid parameter tuples, and write $c_{y_1,\dots,y_k}$ for the concept represented by $(y_1, \dots, y_k)$. Thus,

$$\mathcal{C} = \{c_{y_1,\dots,y_k} : (y_1, \dots, y_k) \in \mathcal{Y}_{\le k}\}.$$

For $i \in [k]$, write $y_{\le i} := (y_1, \dots, y_i)$, and set $\mathcal{Y}_{\le 0} := \{\emptyset\}$. Define the set of valid prefixes of length i by

$$\mathcal{Y}_{\le i} := \{y_{\le i} : \exists\, y_{i+1}, \dots, y_k \text{ such that } (y_1, \dots, y_k) \in \mathcal{Y}_{\le k}\}.$$

Thus, $y_{\le i} \in \mathcal{Y}_{\le i}$ if it can be extended to a complete representation of a concept in $\mathcal{C}$.

For every valid prefix $y_{\le i}$, define the restricted class

$$\mathcal{C}(y_{\le i}) := \{c_{z_1,\dots,z_k} : (z_1, \dots, z_k) \in \mathcal{Y}_{\le k} \text{ and } (z_1, \dots, z_i) = y_{\le i}\}.$$

Hence, if $y_{\le i} = (y_{\le i-1}, y_i) \in \mathcal{Y}_{\le i}$, then

$$\mathcal{C}(y_{\le i}) \subseteq \mathcal{C}(y_{\le i-1}).$$

A hypothesis $g \in \mathcal{C}(y_{\le i})$ is called a *completion* of $y_{\le i}$. In particular, $\mathcal{C}(y_{\le k}) = \{c_{y_1,\dots,y_k}\}$.

Claim B.1. *Suppose that* $\mathrm{VC}(\mathcal{C}) = v$. *Then, for every* $i \in [k]$ *and every* $y_{\le i} \in \mathcal{Y}_{\le i}$,

$$\mathrm{VC}\big(\mathcal{C}(y_{\le i})\big) \le v.$$

Proof. Since $\mathcal{C}(y_{\le i}) \subseteq \mathcal{C}$, the claim follows from the monotonicity of VC dimension under restriction to a subclass. □

Definition B.2 (Realizable prefix state)**.** A pair $(y_{\le i}, S)$, where $y_{\le i} \in \mathcal{Y}_{\le i}$ and S is a labeled sample, is called a *realizable prefix state* if there exists $h_i \in \mathcal{C}(y_{\le i})$ such that

$$S = U(S)^{h_i}.$$

Any such h_i is called a witness for $(y_{\le i}, S)$. With this notations, $\mathrm{err}_S(h_i) = 0$. *realizable prefix state* $(y_{\le i}, S)$ admits that S is $\mathcal{C}(y_{\le i})-$realizable

Definition B.3 (Differential privacy on realizable prefix states)**.** A prefix state $(y_{\le i}, S)$ is realizable prefix state if S is $\mathcal{C}(y_{\le i})-$realizable

A randomized mechanism

$$A : \mathcal{Y}_{\le i} \times \mathcal{Z}^m \to \mathcal{R}$$

is (ε, δ)-differentially private on realizable prefix states if, for every fixed $y_{\le i} \in \mathcal{Y}_{\le i}$, the mechanism $A(y_{\le i}; \cdot)$ is (ε, δ)-differentially private on $\mathcal{C}(y_{\le i})$-realizable inputs (see Definition A.6). consequently we say that if $(y_{\le i}, S)$ and $(y_{\le i}, S')$ are realizable prefix states and S, S' are neighbors then $A(y_{\le i}, S)$ and $A(y_{\le i}, S')$ are (ε, δ) indistinguishable.

This guarantee is weaker than standard differential privacy: it imposes no requirement on neighboring S, S' if either $(y_{\le i}, S)$ or $(y_{\le i}, S')$ is not a realizable prefix state.

B.2 Private Component Oracles

Once several components have been fixed, the learner must choose the next component without losing the ability to complete them to an accurate hypothesis in the full class. We capture this task through a private component oracle. Given the current prefix and labeled sample, the oracle selects the next component while preserving, with high probability, a good completion of the extended prefix. Since the learner invokes these oracles on realizable prefix states, their privacy and utility guarantees are formulated for such inputs. The precise interface is given below.

Assumption B.4 (Private realizable component oracle)**.** *A sequence of* k *algorithms*

$$i = 1, \dots, k: \quad A_i : \mathcal{Y}_{\le i-1} \times \mathcal{Z}^m \longrightarrow \mathcal{Y}_i,$$

with privacy parameters ε, δ *and utility parameters* τ, β*, are called* private component oracles *with sample complexities* $M_{A,1}, \dots, M_{A,k}$*, if the following holds for any choice of* $\delta, \tau, \beta \in (0, 1)$ *with* $\delta \le \beta$ *and* $\varepsilon \in (0, 4]$*:*

1. ***Totality and validity.*** *For every* $S \in \mathcal{Z}^m$*, the output* $y_i \leftarrow A_i(y_{\le i-1}; S)$ *is defined and satisfies* $y_{\le i} = (y_{\le i-1}, y_i) \in \mathcal{Y}_{\le i}$.
2. ***Privacy on realizable state.*** *The mechanism* $A_i(y_{\le i-1}; \cdot)$ *is* (ε, δ)*-differentially private on realizable inputs*$-\mathcal{C}(y_{\le i-1})$ *(Definition B.3).*
3. ***Utility*** *Let* $(y_{\le i-1}, S)$ *be a realizable state and witness by* $h_{i-1} \in \mathcal{C}(y_{\le i-1})$*. Suppose* $m \ge M_{A,i}(\tau, \beta, \varepsilon, \delta)$*. Then, with probability at least* $1 - \beta$*, for* $y_i \leftarrow A_i(y_{\le i-1}; S)$ *and* $y_{\le i} = (y_{\le i-1}, y_i)$*, there exists completion* $g_i \in \mathcal{C}(y_{\le i})$ *such that* $\mathrm{disc}_{U(S)}(g_i, h_{i-1}) \le \tau$.

For a series $A_1, \dots A_k$ we write $M_A(\tau, \beta, \varepsilon, \delta) := \max_{i \in [k]} M_{A,i}(\tau, \beta, \varepsilon, \delta)$.

C Generic Learner

In this section, we present a generic algorithm CarryForwardLerner for $\mathcal{C}$, assuming access to component oracles $A_1, \dots, A_k$. The learner constructs the component vector sequentially using the component oracle, then uses LabelBoostProcedure to carry a labeled sample from one round to the next while keeping the labeled samples realizable with respect to $\mathcal{C}$. We are now ready to state the main guarantee of CarryForwardLearner, and in the next sections, we provide the privacy and utility analysis of the algorithm.

Theorem C.1 (Carry-Forward Learner Guarantee). *Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ be a concept class with a k-component parameterization and with $VC(\mathcal{C}) = v$. Let $A_1, \ldots, A_k$ be an oracle sequence for the components in the sense of Assumption B.4. For every $\varepsilon, \alpha, \beta, \delta \in (0,1)$ with $\delta \leq \beta$, there exists a (ε, δ)-differentially private (α, β, n)-proper PAC-learner $A_{\text{CarryForwardLearner}}$ with sample complexity*

$$n = O\left(\frac{\sqrt{k\log(2/\delta)}}{\varepsilon}\max\left\{\frac{kv}{\alpha}\log\left(\frac{320k^2}{\alpha\beta}\right),\ M_A\left(\frac{\alpha}{40k}, \frac{\beta}{4k}, 4, \frac{\delta}{16ek}\right)\right\}\right).$$

Algorithm 2 CarryForwardLearner

Input: A labeled sample S, target parameters $\alpha, \beta, \varepsilon, \delta \in (0,1)$, and component oracle sequence $A_1, \ldots, A_k$ satisfies assumption B.4

Parameters: Concept class $\mathcal{C}$ represented by k components, and active sample size $m = \left\lfloor \frac{\varepsilon n}{12e^9\sqrt{k\log(2/\delta)}} \right\rfloor$

Set $\tau \leftarrow \frac{\alpha}{40k}$, $\beta_A \leftarrow \frac{\beta}{4k}$, $\varepsilon_A = 4$, $\delta_A = \frac{\delta}{16ek}$
Partition S, independently of record values, into W^{init}, $W^{(0)}$, $W^{(1)}$, each of size $\lfloor \frac{n}{3} \rfloor$.
Set $y_{\leq 0} \leftarrow \emptyset$.
Sample uniformly without replacement $R_0 \subseteq W^{\text{init}}, |R_0| = m$.
Run LabelBoostProcedure$\big(R_0, \emptyset, \mathcal{C}(y_{\leq 0})\big)$, and let $h_0 \in \mathcal{C}(y_{\leq 0}) = \mathcal{C}$ be the internally selected hypothesis. Set $S_0 \leftarrow U(R_0)^{h_0}$.
for $i = 1, \ldots, k$ **do**
    $y_i \leftarrow A_i\big(y_{\leq i-1}; S_{i-1}\big)$.
    Set $y_{\leq i} \leftarrow (y_{\leq i-1}, y_i)$.
    if $i = k$
        return the hypothesis $c_{y_1,\ldots,y_k}$ represented by the full prefix $y_{\leq k}$.
    Sample uniformly without replacement

$$T_i \subseteq W^{((i-1) \bmod 2)}, \qquad |T_i| = m,$$

    Run LabelBoostProcedure$\big(S_{i-1}, U(T_i), \mathcal{C}(y_{\leq i})\big)$.
    Let $h_i \in \mathcal{C}(y_{\leq i})$ be the (internally) selected hypothesis and set $S_i \leftarrow U(T_i)^{h_i}$.

The key invariant is that every round, the state $(y_{\leq i}, S_i)$ remains realizable by $\mathcal{C}(y_{\leq i})$. Therefore, every component oracle is invoked within its promised realizable-input domain. We first establish this invariant.

Note that immediately before each invocation of A_i, the current prefix and active sample form a realizable prefix state. This property holds for every execution of the learner and does not depend on the utility success of the oracles.

Lemma C.2 (Iterated realizability). *At each iteration $i \in [k]$, before the oracle call $y_i \leftarrow A_i(y_{\leq i-1}; S_{i-1})$, there exists $h_{i-1} \in \mathcal{C}(y_{\leq i-1})$ such that $S_{i-1} = U(S_{i-1})^{h_{i-1}}$. Consequently, $(y_{\leq i-1}, S_{i-1})$ is a realizable prefix state (Definition B.2), and every component oracle is invoked on a realizable input.*

In particular for $i \geq 1$: for every $(y_{\leq i}, T_i)$ the state $(y_{\leq i}, S_i)$ is realizable prefix state.

Proof. For $i = 1$, the preliminary LabelBoost call selects $h_0 \in \mathcal{C}(y_{\leq 0}) = \mathcal{C}$ and sets $S_0 = U(R_0)^{h_0}$. Thus $(y_{\leq 0}, S_0)$ is realizable.

Now fix $i < k$. By the validity of A_i, its output y_i defines a valid prefix $y_{\leq i}$. The subsequent call to LabelBoostProcedure selects $h_i \in \mathcal{C}(y_{\leq i})$ and sets $S_i = U(T_i)^{h_i}$. Hence $(y_{\leq i}, S_i)$, which is the input state of A_{i+1}, is realizable. The claim follows by induction. □

The proof of Theorem C.1 is given in the following two sections: The privacy guarantee is proven in Appendix D and the utility guarantee is proven in Appendix E.

D PRIVACY ANALYSIS

We now analyze the privacy of CARRYFORWARDLEARNER. A natural approach would use one reservoir, amplify the privacy of each oracle–LabelBoost iteration, and then apply adaptive composition. This does not work directly because iteration i depends on the reservoir through both the carried sample S_{i-1} and the newly sampled carrier T_i. Moreover, S_i cannot simply be included in the private transcript: relabeling preserves its underlying records, so neighboring inputs may be distinguished by the presence of their differing record. Omitting S_i, on the other hand, leaves the state of the next iteration undetermined.

We therefore alternate between two reservoirs. When analyzing $W^{(b)}$, the incoming sample is carried by $W^{(1-b)}$. A record sampled from $W^{(b)}$ may enter S_i and affect the next component, but after the following relabeling the active sample is again carried by $W^{(1-b)}$. Hence a record from $W^{(b)}$ affect only the labeling at the end of 2 iterations. We group these two iterations into a *macro-block*, which captures the complete influence of one activation of $W^{(b)}$.

Throughout this section, let $A_1, \dots, A_k$ be a sequence of algorithms that satisfy Assumption B.4 with respect to sample complexity m as defined in Algorithm 2.

D.1 THE TWO-ROUND MACRO-BLOCK

We first analyze one reservoir activation at the carrier level. Fix $b \in \{0, 1\}$ and an iteration i in which $T_i \subseteq W^{(b)}$, and fix the state $r_i = (y_{\le i-1}, S_{i-1})$. For this fixed state, the macro-block $\mathrm{MB}_i^{r_i}$ is the mechanism that receives T_i and performs the next two iterations of CARRYFORWARDLEARNER.

The following carrier T_{i+1} is sampled from the disjoint reservoir $W^{(1-b)}$. Hence a change in T_i can affect the second iteration only through the relabeled sample S_i, while the underlying carrier T_{i+1} remains unchanged. Using the extended LabelBoost analysis of Appendix A.5, we prove privacy of $\mathrm{MB}_i^{r_i}$ with respect to neighboring carriers T_i, T_i'. Privacy with respect to the full reservoir $W^{(b)}$ is derived later by subsampling amplification.

Definition D.1 (Two round macro-block)**.** Fix $1 \le i \le k-1$, and recall that T_i is a size-m carrier drawn from a reservoir $W^{((i-1) \bmod 2)}$, and therefore, independent of the reservoir $W^{(i \bmod 2)}$. Define the state at the start of iteration i by

$$r_i := (y_{\le i-1}, S_{i-1}, W^{(i \bmod 2)}).$$

For a fixed state r_i, define $\mathrm{MB}_i^{r_i}(T_i)$ by the following computation:

1. Compute
$$y_i \leftarrow A_i(y_{\le i-1}, S_{i-1}), \qquad y_{\le i} := (y_{\le i-1}, y_i).$$
2. Run
$$h_i \leftarrow \text{LABELBOOSTPROCEDURE}(S_{i-1}, U(T_i), \mathcal{C}(y_{\le i})), \qquad S_i := U(T_i)^{h_i}.$$
3. Compute
$$y_{i+1} \leftarrow A_{i+1}(y_{\le i}, S_i), \qquad y_{\le i+1} := (y_{\le i}, y_{i+1}).$$
4. If $i \le k-2$, independently sample a fresh size-m carrier $T_{i+1} \subseteq W^{(i \bmod 2)}$ uniformly without replacement, and set
$$h_{i+1} \leftarrow \text{LABELBOOSTPROCEDURE}(S_i, U(T_{i+1}), \mathcal{C}(y_{\le i+1})), \qquad S_{i+1} := U(T_{i+1})^{h_{i+1}}.$$

For the privacy analysis, define the augmented output by

$$\mathrm{MB}_i^{r_i}(T_i) := \begin{cases} (y_i, y_{i+1}, S_{i+1}), & 1 \le i \le k-2, \\ (y_{k-1}, y_k, \emptyset), & i = k-1. \end{cases}$$

The real learner does not reveal S_{i+1}; it is included only as an analytical device for representing the state passed to future activations.

Lemma D.2 (Privacy of two-round macro-block)**.** *For every fixed state* $r_i := (y_{\le i-1}, S_{i-1}, W^{(i \bmod 2)})$ *and for every* $1 \le i \le k-1$*, the mechanism* $\mathrm{MB}_i^{r_i}$ *(Definition D.1) over size-*m *carriers* T_i *is* (C, Δ)*-differentially private, where*

$$C := \varepsilon_A + 2 \cdot c_{\mathrm{LB}}, \qquad \Delta := 4e\,\delta_A.$$

Proof.

The proof has two steps. First, we view the second round as a mechanism whose input is the relabeled carrier $S_i = T_i^{h_i}$, and prove that this mechanism is private on realizable neighboring inputs. We then apply Lemma A.15 to the preceding LabelBoost call. Since LabelBoost passes a realizable relabeled carrier to the private mechanism, the lemma gives privacy of the entire macro-block with respect to T_i.

Step 1: Analyzing the second round. For fixed $u = (y_{\le i}, W^{(i \bmod 2)})$, we define K_u as a mechanism of the second round of macro-block:

K_u, given an input S_i consists of labeled sample of size m, acts as follows:

1. Compute $y_{i+1} \leftarrow A_{i+1}(y_{\le i}, S_i)$.
2. If $i = k-1$, output $(y_{i+1}, \emptyset)$.
3. Choose uniformly an size-m subset $T_{i+1} \subseteq W^{(i \bmod 2)}$.
4. Execute $LabelBoostProcedure(S_i, U(T_{i+1}), \mathcal{C}(y_{\le i+1}))$, and let h_{i+1} be the internally chosen hypothesis.
5. Output (y_{i+1}, S_{i+1}), for $S_{i+1} = U(T_{i+1})^{h_{i+1}}$.

Note that steps 3-4 in $\mathrm{MB}_i^{r_i}$ can be replaced by calling $K_u(S_i)$ where $u = (y_{\le i}, W^{(i \bmod 2)})$.

Claim D.3. *For a fixed* u*, the mechanism* K_u *is* $(\varepsilon_A + c_{\mathrm{LB}}, \delta_A)$*-differentially private, on every pair of neighboring samples* S_i, S_i' *such that both* (u, S_i) *and* (u, S_i') *are* $\mathcal{C}(y_{\le i})$*-realizable prefix states according to Definition B.2*

Proof. For a fixed u, the class $\mathcal{C}(y_{\le i})$ is also fixed. Hence, Assumption B.4 implies that $A_{i+1}(u, S_i)$ and $A_{i+1}(u, S_i')$ are $(\varepsilon_A, \delta_A)$-indistinguishable For $i = k-1$, the proof of the claim follows. We next assume that $i \le k-2$.

Fix an output y_{i+1} and a sample carried $T_{i+1} \subseteq W^{(i \bmod 2)}$. Hence the class $\mathcal{C}(y_{\le i+1})$ and the carrier $U(T_{i+1})$ are fixed. Lemma A.17 therefore gives c_{LB}-differential privacy for the release of $S_{i+1} = U(T_{i+1})^{h_{i+1}}$. We conclude by (adaptive) basic composition that K_u is $(\varepsilon_A + c_{\mathrm{LB}}, \delta_A)$-differentially private on $\mathcal{C}(y_{\le i})$-realizable prefix states.

□

Step 2: Analyzing the entire process. Recall that $\mathrm{MB}_i^{r_i}(T_i)$, for $r_i = (y_{\le i-1}, S_{i-1}, W^{(i \bmod 2)})$, can be describe as follows:

1. Compute $y_i \leftarrow A_i(y_{\le i-1}, S_{i-1})$.
2. Run $h_i \leftarrow \textsc{LabelBoostProcedure}(S_{i-1}, U(T_i), \mathcal{C}(y_{\le i}))$, and let $S_i := U(T_i)^{h_i}$,
3. Compute $(y_{i+1}, S_{i+1}) \leftarrow K_u(S_i)$, where $u = (y_{\le i}, W^{(i \bmod 2)})$.
4. Output (y_i, y_{i+1}, S_{i+1}).

To prove that $\mathrm{MB}_i^{r_i}$ is differentially private, fix neighboring T_i, T_i' and consider the two executions $\mathrm{MB}_i^{r_i}(T_i)$ and $\mathrm{MB}_i^{r_i}(T_i')$. Note that computing the y_i part (step 1) is done independently of the

input. Therefore, we assume without loss of generality that this part is fixed in both exectuins The LabelBoost call is applied to the neighboring inputs

$$(S_{i-1}, U(T_i)) \sim (S_{i-1}, U(T_i')).$$

Since K_u is applied to the relabeled carrier produced by LabelBoost and is $(\varepsilon_A + c_{\mathrm{LB}}, \delta_A)$-differentially private on $\mathcal{C}(y_{\le i})$-realizable inputs, the case described in Remark A.16 applies. Hence, by Lemma A.15, the entire process is $(\varepsilon_A + 2 \cdot c_{\mathrm{LB}}, 4e\delta_A)$-differentially private.

□

D.2 RESERVOIR PRIVACY

We first establish the privacy of CARRYFORWARDLEARNER with respect to a change in a single record of W^{init}. Since the initial reservoir is accessed only once, privacy with respect to W^{init} does not require any additional composition analysis.

Lemma D.4 (Initialization privacy with respect to R_0)**.** *Fix the reservoirs $W^{(0)}$ and $W^{(1)}$. Viewed as a mechanism of R_0, the output of* CARRYFORWARDLEARNER *is*

$$(C, \Delta) = (\varepsilon_A + 2 \cdot c_{\mathrm{LB}},\ 4e\delta_A)\text{-}\textit{differentially private}.$$

Proof. Let $K_\emptyset$ denote the continuation following the preliminary LabelBoost call where On input S_0, it can be described as follow:

1. $y_1 \leftarrow A_1(y_{\le 0}; S_0)$
2. if k=1, output $(y_1, \emptyset)$
3. Choose uniformly size-m subset $T_1 \subseteq W^{(0)}$
4. Execute $LabelBoostProcedure(S_0, U(T_1), \mathcal{C}(y_{\le 1}))$, and let h_1 be the internally chosen hypothesis.
5. Output (y_1, S_1), for $S_1 = U(T_1)^{h_1}$.

We first analyze $K_\emptyset$ on neighboring $\mathcal{C}$-realizable inputs S_0, S_0'. By Assumption B.4, A_1 is $(\varepsilon_A, \delta_A)$-differentially private on these inputs. If $k = 1$, then $K_\emptyset = A_1$. Otherwise, after fixing the output y_1 of A_1 and the independently sampled carrier T_1, the class $\mathcal{C}(y_{\le 1})$ and the unlabeled carrier $U(T_1)$ are fixed. By Lemma A.17 and adaptive basic composition, $K_\emptyset$ is therefore $(\varepsilon_A + c_{\mathrm{LB}}, \delta_A)$-differentially private on neighboring $\mathcal{C}$-realizable inputs in either case.

The preliminary LabelBoost call produces $S_0 = U(R_0)^{h_0}$ for some $h_0 \in \mathcal{C}$. Applying Lemma A.15 to this call followed by $K_\emptyset$ shows that the block output is $(\varepsilon_A + 2c_{\mathrm{LB}}, 4e\delta_A)$- differentially private with respect to R_0.

The sample S_0 is not released separately. After the initialization block, the learner accesses R_0 and S_0 only through the block output (y_1, S_1), or through y_1 when $k = 1$. Since the reservoirs are fixed in this analysis, all subsequent computation is post-processing with respect to changes in R_0. Therefore the final output of CARRYFORWARDLEARNER has the same privacy guarantee. □

Definition D.5 (Amplified macro-block mechanism)**.** Fix a round i and a state r_i. Let $b = (i-1) \bmod 2$ The mechanism $\mathrm{AMB}_i^{r_i}$ takes a reservoir $W^{(b)}$ of size $\lfloor n/3 \rfloor$, samples $T_i \subseteq W^{(b)}$ uniformly without replacement with $|T_i| = m$, and outputs

$$\mathrm{AMB}_i^{r_i}(W^{(b)}) := \mathrm{MB}_i^{r_i}(T_i).$$

Corollary D.6 (Reservoir amplification)**.** *Let $\mathcal{M}$ be a (C, Δ)-differentially private mechanism on samples of size m. Given a dataset W of size $\lfloor n/3 \rfloor$, sample $T \subseteq W$ uniformly without replacement with $|T| = m$, and output $\mathcal{M}(T)$.*

Then the resulting mechanism on W is (η, ζ)-differentially private, where

$$q := \frac{m}{\lfloor n/3 \rfloor}, \qquad \eta := \log\bigl(1 + q(e^C - 1)\bigr), \qquad \zeta := q\Delta.$$

Using $\log(1+x) \le x$ *and* $\delta_A = \delta/(16ek)$*, we have*

$$\eta \le \frac{m}{\lfloor n/3 \rfloor}\left(e^{\varepsilon_A + 2\cdot c_{\mathrm{LB}}} - 1\right), \qquad \zeta = \frac{m}{\lfloor n/3 \rfloor}\frac{\delta}{4k} \le \frac{\delta}{4k}.$$

Proof. This follows directly from the fixed-size subsampling amplification guarantee of Theorem A.4. □

Corollary D.7 (Initialization-reservoir privacy)**.** *With respect to a change in one record of* W^{init}*, the complete* CARRYFORWARDLEARNER *is* (η, ζ)*-differentially private, where*

$$q = \frac{m}{\lfloor n/3 \rfloor}, \qquad \eta = \log\bigl(1 + q(e^C - 1)\bigr), \qquad \zeta = q\Delta.$$

Proof. By Lemma D.4, the complete learner is (C, Δ)-differentially private as a mechanism of R_0. Since R_0 is a uniform size-m subsample of W^{init}, the claim follows directly from Corollary D.6. □

Corollary D.8 (Privacy of the amplified macro-block)**.** *For every fixed round* i *and state* r_i*, the mechanism* $\mathrm{AMB}_i^{r_i}$ *is* (η, ζ)*-differentially private, where*

$$q = \frac{m}{\lfloor n/3 \rfloor}, \qquad \eta = \log\bigl(1 + q(e^C - 1)\bigr), \qquad \zeta = q\Delta.$$

Proof. By Lemma D.2, $\mathrm{MB}_i^{r_i}$ is (C, Δ)-differentially private on carriers of size m. The claim follows directly from Corollary D.6. □

D.3 PRIVACY CALIBRATION

In this section we want to conclude the final privacy of CARRYFORWARDLEARNER, Previously, we established the entire algorithm privacy with respect to a change in W^{init}. We now consider a change in $W^{(b)}, b \in (0,1)$.

Let $b = (i-1) \bmod 2$, fix $W^{init}, W^{(1-b)}$ in Appendix D.2 we established privacy of $\mathrm{AMB}_i^{r_i}$ with respect to a change in W^b and a fixed state $r_i = (y_{\le i-1}, S_{i-1}, W^{(1-b)})$. we now remove conditioning of r_i using adaptive composition. We record $\mathrm{AMB}_i^{r_i}$ output in a transcript and show that, conditioned on the preceding transcript, the state of the next activation is fixed.

In order to record the transcript fix one reservoir $W^{(b)}$, and enumerate its activations by j. let r_j denote $\mathrm{AMB}_j^{r_j}$ state, and let O_j be $\mathrm{AMB}_j^{r_j}$ augmented output.

Define the transcript through the jth activation by

$$\mathsf{tr}_j := (O_1, \dots, O_j), \qquad \mathsf{tr}_0 := \emptyset.$$

Lemma D.9 (Transcript-determined boundary)**.** *Fix* $b \in \{0,1\}$*, and fix* W^{init} *and* $W^{(1-b)}$*. Let* $i = 2j - 1 + b$ *be the round of the* j*th activation of* $W^{(b)}$*. Then the boundary state* r_i *is determined by* $(r_{1+b}, \mathsf{tr}_{j-1})$*. Consequently, conditioned on* $(r_{1+b}, \mathsf{tr}_{j-1})$*, the release* O_j *is* (η, ζ)*-differentially private with respect to* $W^{(b)}$*.*

Proof. The claim is immediate for $j = 1$. Let $j \ge 2$. Since consecutive activations of $W^{(b)}$ are separated by two rounds, the preceding output is $O_{j-1} = (y_{i-2}, y_{i-1}, S_{i-1})$. The preceding transcript therefore determines $y_{\le i-1}$ and S_{i-1}. Since $W^{(1-b)}$ is fixed, it also determines the complete boundary state r_i.

Thus, conditioned on $(r_{1+b}, \mathsf{tr}_{j-1})$, the boundary state r_i is fixed. By Corollary D.8, the conditional mechanism mapping $W^{(b)}$ to O_j is (η, ζ)-differentially private. □

Corollary D.10 (Privacy with respect to a reservoir)**.** *Fix* $b \in \{0,1\}$*, and consider neighboring inputs that differ in one record of* $W^{(b)}$*, while* W^{init} *and* $W^{(1-b)}$ *are unchanged. Then* CARRYFORWARDLEARNER *is* $(\varepsilon_{\mathrm{res}}, \delta_{\mathrm{res}})$*-differentially private with respect to this change, where*

$$\ell := \left\lceil \frac{k-1}{2} \right\rceil,$$

and, for every $\delta_{\mathrm{ac}} > 0$*,*

$$\varepsilon_{\mathrm{res}} \le \sqrt{2\ell \log(1/\delta_{\mathrm{ac}})}\,\eta + \ell\eta(e^{\eta} - 1), \qquad \delta_{\mathrm{res}} \le \ell\zeta + \delta_{\mathrm{ac}}.$$

Proof. The reservoir $W^{(b)}$ supplies at most ℓ carriers. By Lemma D.9, conditioned on the initial boundary r_{1+b} and each preceding transcript, every activation is (η, ζ)-differentially private with respect to $W^{(b)}$. Adaptive composition (Theorem A.5) gives the stated bounds for the joint transcript, and hence for the output of CARRYFORWARDLEARNER by post-processing, conditioned on r_{1+b}.

Before round $1 + b$, the algorithm does not access $W^{(b)}$. Since the other reservoirs are identical, coupling the two executions with the same randomness gives the same initial boundary r_{1+b}. Averaging over its common distribution removes the conditioning while preserving the same privacy parameters. □

Lemma D.11 (Privacy of CARRYFORWARDLEARNER)**.** *Let* $\varepsilon \le 1$ *and* $\delta \in (0, 1)$*. Under the parameter choices of Algorithm 2,* $\varepsilon_A = 4$*,* $\delta_A = \frac{\delta}{16ek}$*,* $m = \left\lfloor \frac{\varepsilon n}{12e^9\sqrt{k \log(2/\delta)}} \right\rfloor$*,* CARRYFORWARDLEARNER *is* (ε, δ)*-differentially private on arbitrary input samples.*

Proof. Let

$$q := \frac{m}{\lfloor n/3 \rfloor}, \qquad \eta := \log\bigl(1 + q(e^{C} - 1)\bigr), \qquad \zeta := 4eq\delta_A.$$

Using $\log(1 + x) \le x$, with $\varepsilon_A = 4, \delta_A = \frac{\delta}{16ek}$ and $C = 2c_{LB} + \varepsilon_A < 9$ we obtain

$$\eta \le q(e^{C} - 1) \le \frac{\varepsilon}{4\sqrt{k \log(2/\delta)}}, \qquad \zeta = \frac{q\delta}{4k} \le \frac{\delta}{4k}.$$

Set $\delta_{\mathrm{ac}} := \delta/2$. For $k \ge 2$, let

$$\ell := \left\lceil \frac{k-1}{2} \right\rceil \le k.$$

we have $\eta \le 1$, and hence $e^{\eta} - 1 \le 2\eta$. Therefore,

$$\begin{aligned} \varepsilon_{\mathrm{res}} &\le \sqrt{2\ell \log(2/\delta)}\,\eta + \ell\eta(e^{\eta} - 1) \\ &\le \frac{\sqrt{2}}{4}\varepsilon + \frac{2\varepsilon^2}{16 \log(2/\delta)} \le \varepsilon, \end{aligned}$$

and

$$\delta_{\mathrm{res}} \le \ell\zeta + \delta_{\mathrm{ac}} \le \frac{\delta}{4} + \frac{\delta}{2} < \delta.$$

Thus, by Corollary D.10, the complete learner is (ε, δ)-differentially private with respect to a change in either reservoir. By Corollary D.7, the same conclusion holds with respect to a change in W^{init}. When $k = 1$, only the latter case is required.

Finally, condition on the value-independent random partition of the input. For every fixed partition, neighboring inputs differ in one record of exactly one reservoir, while the other reservoirs are identical. Therefore either Corollary D.7 or Corollary D.10 applies. Since the same (ε, δ)-privacy guarantee holds for every fixed partition, averaging over the partition randomness preserves it. □

E UTILITY ANALYSIS

We now analyze the utility of CARRYFORWARDLEARNER. Throughout this section, let $A_1, \ldots, A_k$ be a sequence of algorithms that satisfy Assumption B.4 with maximal sample complexity M_A.

At the beginning of iteration i, the active sample S_{i-1} is labeled by a witness $h_{i-1} \in \mathcal{C}(y_{\le i-1})$. The component oracle selects y_i while ensuring that the new restricted class $\mathcal{C}(y_{\le i})$ contains a completion close to h_{i-1} on S_{i-1}. The subsequent LABELBOOSTPROCEDURE selects the hypothesis g_i used to label the next carrier, and Corollary A.13 ensures that g_i remains empirically close to h_{i-1}.

All-pairs generalization guarantee then converts this empirical bound into population disagreement $O(\tau)$. Thus every iteration changes the carried hypothesis by at most $O(\tau)$ in population disagreement. After accounting for initialization and telescoping over the k iterations, choosing $\tau = \Theta(\alpha/k)$ gives the desired error α. The following lemma establishes the one-round guarantee used in this argument.

Lemma E.1 (One-round utility guarantee)**.** *Let $\mathcal{C}$ be a concept class with $\mathrm{VC}(\mathcal{C}) = v$, Let $\beta_{VC}, \beta_A, \beta_{LB} \in (0,1)$, and fix $i < k$. Suppose that $(y_{\le i-1}, S_{i-1})$ is a realizable prefix state, witnessed by some $h_{i-1} \in \mathcal{C}(y_{\le i-1})$, and that $U(S_{i-1})$ consists of m i.i.d. samples from $\mathcal{D}_X$.*

Let $y_i \leftarrow A_i(y_{\le i-1}; S_{i-1})$, and let $g_i \in \mathcal{C}(y_{\le i})$ be the hypothesis selected by

LABELBOOSTPROCEDURE$(S_{i-1}, U(T_i), \mathcal{C}(y_{\le i}))$. *For a sufficiently large universal constant c, it suffices that*

$$m \ge \max\left\{ M_A(\tau, \beta_A, \varepsilon_A, \delta_A), \frac{c}{\tau}\left(v \log\frac{2}{\tau\beta_{\mathrm{LB}}} + \log\frac{1}{\beta_{\mathrm{VC}}} \right) \right\}.$$

Then, with probability at least $1 - \beta_A - \beta_{\mathrm{LB}}$,

$$\mathrm{err}_{S_{i-1}}(g_i) = \mathrm{disc}_{U(S_{i-1})}(g_i, h_{i-1}) \le 2\tau.$$

Moreover, with probability at least $1 - \beta_A - \beta_{\mathrm{LB}} - \beta_{\mathrm{VC}}$,

$$\mathrm{disc}_{\mathcal{D}_X}(g_i, h_{i-1}) < 20\tau.$$

Proof. By the assumed choice of m, the component-oracle, LabelBoost, and all-pairs generalization sample requirements are all satisfied.

By Assumption B.4, the component-oracle utility guarantee implies that, except with probability β_A, there exists a completion $h_i \in \mathcal{C}(y_{\le i})$ satisfying $\mathrm{disc}_{U(S_{i-1})}(h_i, h_{i-1}) \le \tau$. Since S_{i-1} is labeled by h_{i-1}, this is equivalent to $\mathrm{err}_{S_{i-1}}(h_i) \le \tau$.

By Corollary A.13, the LabelBoost utility guarantee implies that, except with an additional failure probability of at most β_{LB},

$$\mathrm{err}_{S_{i-1}}(g_i) = \mathrm{disc}_{U(S_{i-1})}(g_i, h_{i-1}) \le 2\tau.$$

Finally, by Theorem A.9, the all-pairs generalization guarantee holds simultaneously for every pair of hypotheses in $\mathcal{C}$. Hence, except with probability β_{VC}, the empirical bound above implies $\mathrm{disc}_{\mathcal{D}_X}(g_i, h_{i-1}) < 20\tau$. A union bound gives total failure probability at most $\beta_A + \beta_{\mathrm{LB}} + \beta_{\mathrm{VC}}$. □

Theorem E.2 (Iterated utility)**.** *Let $\mathcal{D}$ be realizable by some $h^\star \in \mathcal{C}$, where $\mathrm{VC}(\mathcal{C}) = v$, and suppose that each component oracle A_i satisfies assumption B.4. Set*

$$\tau := \frac{\alpha}{40k}, \qquad \beta_A, \beta_{\mathrm{LB}}, \beta_{\mathrm{VC}} \le \frac{\beta}{4k}.$$

It suffices to take

$$m = O\left(\max\left\{ \frac{kv}{\alpha}\log\left(\frac{320k^2}{\alpha\beta}\right),\ M_A\left(\frac{\alpha}{40k}, \frac{\beta}{4k}, 4, \frac{\delta}{16ek}\right) \right\} \right).$$

Then, with probability at least $1 - \beta$, CARRYFORWARDLEARNER *outputs*

$$h_k := c_{y_1,\ldots,y_k} \in \mathcal{C}$$

satisfying

$$\mathrm{err}_{\mathcal{D}}(h_k) \le \alpha.$$

Proof. Consider one round of CARRYFORWARDLEARNER for $i \in [k]$ .let $h_{i-1} \in \mathcal{C}(y_{\le i-1})$ be the realizable sate witness, let $g_i \leftarrow$ LABELBOOSTPROCEDURE$(S_{i-1}, U(T_i), \mathcal{C}(y_{\le i}))$ the hypothesis completion. For $0 < i < k$ define

$$E_i := \{\mathrm{disc}_{\mathcal{D}_X}(g_i, h_{i-1}) < 20\tau\}.$$

By Lemma C.2, every component oracle satisfies Assumption B.4 is invoked on a realizable prefix state, and hence its utility guarantee applies. Moreover, $U(S_0) = U(R_0)$, while $U(S_{i-1}) = U(T_{i-1})$

for $i \geq 2$. Each carrier is a uniform selection of m records from an i.i.d. reservoir, Consequently, $U(S_{i-1})$ consist m i.i.d samples from $\mathcal{D}_X$. Hence Lemma E.1 guarantee applies in every round.

For initialization, define $E_0 := \{\mathrm{disc}_{\mathcal{D}_X}(h_0, h^\star) < 20\tau\}$. Since $h^\star$ has zero empirical error on R_0 and R_0 is i.i.d samples from $\mathcal{D}_X$, by the preliminary LabelBoost guarantee and Theorem A.9 imply $\Pr[E_0] \geq 1 - (\beta_{\mathrm{LB}} + \beta_{\mathrm{VC}})$.

For the last round, define $E_k := \{\mathrm{disc}_{\mathcal{D}_X}(h_{k-1}, h_k) < 20\tau\}$. by oracle utility guarantee and Theorem A.9 $\Pr[E_k] = 1 - \beta_A - \beta_{VC}$.

A union bound over $E_0, E_1, \ldots, E_k$ now gives

$$\Pr\left[\bigcap_{i=0}^{k} E_i\right] \geq 1 - (k\beta_A + k\beta_{\mathrm{LB}} + (k+1)\beta_{\mathrm{VC}}) \geq 1 - \beta.$$

On $\bigcap_{i=0}^{k} E_i$, the triangle inequality gives

$$\begin{aligned}
\mathrm{err}_{\mathcal{D}}(h_k) &= \mathrm{disc}_{\mathcal{D}_X}(h_k, h^\star) \\
&\leq \mathrm{disc}_{\mathcal{D}_X}(h_0, h^\star) + \mathrm{disc}_{\mathcal{D}_X}(h_{k-1}, h_k) + \sum_{i=1}^{k-1} \mathrm{disc}_{\mathcal{D}_X}(g_i, h_{i-1}) \\
&< 20(k+1)\tau \leq \alpha.
\end{aligned}$$

Hence by probability $1 - \beta$ CarryForwaedLearner outputs $c_{y_1 \ldots y_k}$ and $\mathrm{err}_{\mathcal{D}_X}(h_k) \leq \alpha$ □

Corollary E.3 (CarryForwardLearner utility)**.** *Let $\mathcal{C}$ be a concept class with k-component parameterization and $\mathrm{VC}(\mathcal{C}) = v$. Let $\mathcal{D}$ be a distribution realizable by some $h^\star \in \mathcal{C}$, and let $A_1, \ldots, A_k$ be the input oracle sequence with maximal sample complexity M_A. Let $S \sim \mathcal{D}^n$ be an i.i.d. sample of size n given as input to* CarryForwardLearner*. Then, for every $\varepsilon.\alpha, \beta, \delta \in (0,1)$. where*

$$n = O\left(\frac{\sqrt{k\log(2/\delta)}}{\varepsilon} \max\left\{\frac{kv}{\alpha}\log\left(\frac{320k^2}{\alpha\beta}\right),\ M_A\left(\frac{\alpha}{40k}, \frac{\beta}{4k}, 4, \frac{\delta}{16ek}\right)\right\}\right).$$

Then, with probability at least $1 - \beta$, CarryForwardLearner *outputs $c_{y_1,\ldots,y_k} \in \mathcal{C}$ satisfying $\mathrm{err}_{\mathcal{D}}(c_{y_1,\ldots,y_k}) \leq \alpha$.*

Proof. by Algorithm 2

$$n = O\left(\frac{m\sqrt{k\log(2/\delta)}}{\varepsilon}\right).$$

by theorem E.2

$$m = O\left(\max\left\{\frac{kv}{\alpha}\log\left(\frac{320k^2}{\alpha\beta}\right),\ M_A\left(\frac{\alpha}{40k}, \frac{\beta}{4k}, 4, \frac{\delta}{16ek}\right)\right\}\right).$$

□

F Applications

In this section, we present applications of the generic CarryForwardLearner framework. Each application is obtained by constructing a sequence of component oracles satisfying Assumption B.4. We give three applications: the first two concern private learning of halfspaces, while the third concerns closure under Boolean composition.

F.1 Common Halfspace Coordinate Framework

We develop a common coordinate framework for the two component-oracle instantiations below: the private quasi-concave optimizer of Cohen et al. (2023) and the IPConcave algorithm of Nissim et al. (2026b). Both applications optimize a quasi-concave function for the linear feasibility problem.

The Linear Feasibility Problem In this problem, we are given a finite grid $\mathcal{G} = [[\pm X]] := \{z \in \mathbb{Z} : |z| \leq X\}$ for some $X \in \mathbb{N}$. A dataset $S^{\mathrm{LP}} \in (\mathcal{G}^d \times \mathcal{G})^*$, such each $(x, y) \in S^{\mathrm{LP}}$ represents the linear constraint $\langle x, a\rangle \geq y$ which defined by the halfspace $h_{x,y} \in \mathbb{R}^d$ on a point $a \in \mathbb{R}^d$. The depth of a with respect to S^{LP} is $\mathrm{depth}_{S^{\mathrm{LP}}}(a) := \left|\left\{(x, y) \in S^{\mathrm{LP}} : \langle x, a\rangle \geq y\right\}\right|$, namely, the number of constraints in S^{LP} satisfied by a. The linear-feasibility problem is to find a point $a \in \mathbb{R}^d$ of large depth.

For the halfspace application, we identify the labels $\{0, 1\}$ with $\{-1, 1\}$. Thus, $S = ((x_j, \ell_j))_{j=1}^n \in (\mathcal{X} \times \{-1, 1\})^n$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is finite, the oracle consider the standard reduction from halfspace to linear-feasibility in order to optimize the depth function over $\mathcal{X}$, each coordinate oracle output represent a matching halfspace coordinate.

A key observation is that optimizing the depth function one coordinate at a time in over a realizable dataset result with an quasi-concave function. hence we able to to use CARRYFORWARDLEARNER to preserve realizable prefix state and optimize depth one coordinate at a time using quasi-concave optimization.

In Appendix I, we prove that the reduction from halfspaces to linear feasibility preserves realizability: For the linear-feasibility to halfspace reduction, we assume general position (defined by Kaplan et al. (2020b)): a set of points in $\mathbb{R}^d$ is in general position if no $d + 1$ of them lie on the same hyperplane. Under this assumption, the transfer from constraint violations to classification errors incurs at most d additional errors.

Since general position need not hold initially, we follow Kaplan et al. (2020b) and apply a sufficiently small perturbation that preserves realizability of the initial labeled sample and places its points in general position with high probability. In the coordinate construction and statements below, S denotes an active sample on the fixed perturbed and rescaled grid described in Appendix I.

To simplify notation, we write depth_S for the depth with respect to S^{LP}. Thus $\mathrm{depth}_S(p)$ counts the constraints induced by S that are satisfied by a full parameter point $p = (a, w) \in \mathbb{R}^{d+1}$.

Write the halfspace parameter vector in the order

$$y = (w, a_1, \ldots, a_d), \qquad k = d + 1.$$

Definition F.1 (Coordinate depth objective)**.** Fix a prefix $y_{\leq i-1}$. For a labeled sample S, define

$$Q_i^S(z) := \max_{y_{i+1},\ldots,y_k \in \mathbb{R}} \mathrm{depth}_S\big(y_{\leq i-1}, z, y_{i+1}, \ldots, y_k\big).$$

Thus $Q_i^S(z)$ is the largest number of constraints induced by S that can still be satisfied after fixing the ith coordinate to z. Changing one example in S changes one induced constraint, so Q_i^S has sensitivity one.

Both Kaplan et al. (2020b) and Nissim et al. (2026b) use the objective cdepth in order to apply a quasi-concave optimization algorithm. We use the iterated realizability property of CARRYFORWARDLEARNER, which ensures that every $(y_{\leq i-1}, S_{i-1})$ is a realizable prefix state (see Lemma C.2). The following lemma shows that, under prefix realizability, the coordinate objective Q_i^S, defined directly from depth, is quasi-concave on its finite domain.

Definition F.2 (Finite coordinate domains.)**.** We set $\widetilde{\mathcal{Y}}_1 := \{-1, 1\}$. For $i = 2, \ldots, k$, after fixing $y_{\leq i-1} = (w, a_1, \ldots, a_{i-2})$, with each preceding coordinate chosen from its corresponding finite domain, let $\widetilde{\mathcal{Y}}_i(y_{\leq i-1})$ be the finite domain for a_{i-1} obtained from the construction of Kaplan et al. (2020b, Definition 4.4), described in appendix I, applied to the d-dimensional linear-feasibility instance obtained after fixing w. The domains are constructed for the perturbed and rescaled finite grid as in Kaplan et al. (2020b, proof of Theorem B.7) [4]

Throughout both oracle instantiations, let

$$N := \max_{\substack{i \in [k],\, y_{\leq i-1} \in \mathcal{Y}_{\leq i-1} \\ y_j \in \widetilde{\mathcal{Y}}_j(y_{\leq j-1}) \text{ for all } j<i}} \left|\widetilde{\mathcal{Y}}_i(y_{\leq i-1})\right|.$$

[4] We use the corrected perturbation resolution from Lemma I.6, implemented in Lemma I.7. This enlarges the grid and the resulting finite-domain size bounds, but adds only $O(1)$ to the bound on $\log^* N$. Consequently, the asymptotic sample complexity of our applications is unchanged.

Lemma F.3 (Realizable completion in finite domains). *Let $i \in [k]$, and let $(y_{\le i-1}, S)$ be a realizable prefix state with $m = |S|$, where the prefix is generated by the domains of Definition F.2. Then Q_i^S is sensitivity-one and quasi-concave on $\widetilde{\mathcal{Y}}_i(y_{\le i-1})$, and there exists $y_i \in \widetilde{\mathcal{Y}}_i(y_{\le i-1})$ such that $y_{\le i}$ admits a completion satisfying all m linear constraints induced by S. Equivalently,*

$$Q_i^S(y_i) = \max_{z \in \widetilde{\mathcal{Y}}_i(y_{\le i-1})} Q_i^S(z) = m.$$

Proof. See Lemma I.9. □

Corollary F.4 (Depth to disagreement transfer). *Let $(y_{\le i-1}, S)$ be a realizable prefix state, witnessed by $h_{i-1} \in \mathcal{C}(y_{\le i-1})$, where the prefix is generated by the domains of Definition F.2. Suppose that $U(S)$ is in general position. If $y_i \in \widetilde{\mathcal{Y}}_i(y_{\le i-1})$ admits a completion $g_i \in \mathcal{C}(y_{\le i})$ satisfying at least $m - E$ of the linear constraints induced by S, where $m = |S|$, then*

$$\mathrm{disc}_{U(S)}(g_i, h_{i-1}) \le \frac{E + d}{m}.$$

Proof. Apply Corollary I.10 to the parameter vector representing g_i. It makes at most $E + d$ errors on S. Since $S = U(S)^{h_{i-1}}$, these errors are precisely its disagreements with h_{i-1} on $U(S)$. □

Theorem F.5 (Halfspace learning from component oracles under general position). *Let $\mathcal{C}$ be the class of halfspaces over the finite grid $\mathcal{X} \subseteq \mathbb{R}^d$, with the coordinate representation above and $k = d + 1$. Suppose that the component oracles satisfy Assumption B.4, except that their utility guarantees are required only on realizable prefix states in general position. Let M_A be their common sample requirement for the perturbed and rescaled grid.*

For every $\alpha, \beta, \varepsilon, \delta \in (0, 1)$ with $\delta \le \beta$, there exists an (ε, δ)-differentially private proper (α, β)-PAC learner for $\mathcal{C}$ in the realizable setting, with sample complexity

$$n = O\left(\frac{\sqrt{k \log(4/\delta)}}{\varepsilon} \max\left\{ \frac{kd}{\alpha} \log\left(\frac{1280k^2}{\alpha\beta} \right),\ M_A\left(\frac{\alpha}{80k}, \frac{\beta}{8k}, 4, \frac{\delta}{32ek} \right) \right\} \right).$$

Proof. See Appendix I appendix I □

F.2 A Coordinate Oracle via Private Quasi-Concave Optimization

In this application we use the quasi-concave optimizer of Cohen et al. Section 4.1.

Cohen et al. (2023) reduces private quasi-concave optimization to a cumulatively private interior-point problem, Cohen et al. (2023, Theorem 4.2) provides the required interior-point algorithm. we defined the algorithm as QCOPT. Together with Appendix F.1 the function Q_i^S (see Definition F.1) is quasi concave with sensitivity 1, we apply QCOPT with Q_i^S and samples S on finite domain in order to yield the following black-box guarantee.

Proposition F.6 (Cohen et al., Section 4.1 and the proof of Theorem 4.2). *(Cohen et al., 2023) Let $\mathcal{Y}$ be a finite totally ordered domain of size N, and let $Q(S, \cdot) : \mathcal{Y} \to \mathbb{R}$ be a sensitivity-one quasi-concave quality function. Then there exists an $(\varepsilon_A, \delta_A)$-differentially private algorithm* QCOPT *such that, with probability at least $1 - \delta_A$, it outputs $\widehat{y} \in \mathcal{Y}$ satisfying*

$$Q(S, \widehat{y}) \ge \max_{y \in \mathcal{Y}} Q(S, y) - E_{\mathrm{QC}},$$

where

$$E_{\mathrm{QC}} = \widetilde{O}\left(\frac{2^{\log^* N} \log(1/\delta_A)}{\varepsilon_A} \right).$$

Here $\widetilde{O}$ suppresses polynomial factors in $\log^* N$.

In the following, we denote by $\mathsf{QCOpt}\,(S, Q, \mathcal{Y}, \varepsilon, \delta)$ the algorithm QCOpt executed on a dataset S over a finite domain $\mathcal{Y}$, and optimizes the function Q using privacy parameters ε, δ.

Coordinate oracle. By Lemma F.3, Q_i^S satisfies the sensitivity and quasi-concavity requirements of Proposition F.6 on every realizable prefix state. On input $(y_{\le i-1}; S)$, the oracle constructs Q_i^S and $\widetilde{\mathcal{Y}}_i(y_{\le i-1})$ as above and returns

$$A_i^{\mathrm{QC}}(y_{\le i-1}; S) := \mathsf{QCOpt}_{\varepsilon_A,\delta_A}\left(S, Q_i^S, \widetilde{\mathcal{Y}}_i(y_{\le i-1})\right).$$

Total extension. For every fixed prefix, the domain $\widetilde{\mathcal{Y}}_i(y_{\le i-1})$ is finite and nonempty. We use the following total extension of QCOpt. The mechanism executes the construction of Cohen et al.; if the construction fails to produce a valid input for one of its subroutines, or if its output does not belong to $\widetilde{\mathcal{Y}}_i(y_{\le i-1})$, it returns

$$\min \widetilde{\mathcal{Y}}_i(y_{\le i-1})$$

instead.

Lemma F.7 (Realizable coordinate oracle)**.** *The coordinate oracle A_i satisfies Assumption B.4 under general position with failure parameter $\beta_A \ge \delta_A$. Hence, with probability at least $1-\beta_A$, its output y_i that admits a completion $g_i \in \mathcal{C}(y_{\le i})$ satisfying*

$$\mathrm{disc}_{U(S)}(g_i, h_{i-1}) \le \frac{E_{\mathrm{QC}} + d}{m}.$$

Hence, if

$$m \ge \frac{E_{\mathrm{QC}} + d}{\tau},$$

then $\mathrm{disc}_{U(S)}(g_i, h_{i-1}) \le \tau$. Consequently, recall $E_{\mathrm{QC}} = \widetilde{O}\Big(\frac{2^{\log^* N}\log(1/\delta_A)}{\varepsilon_A}\Big)$ hence:

$$M_A(\tau, \delta_A, \varepsilon_A, \delta_A) = \widetilde{O}\left(\frac{2^{\log^* N}\log(1/\delta_A)/\varepsilon_A + d}{\tau}\right).$$

Proof. Let $(y_{\le i-1}, S)$ be the oracle input. Validity follow from the oracle definition, when $(y_{\le i-1}, S)$ is realizable prefix the totality follow from A_i definition otherwise totality guarantee from the Total extension we provided.

For privacy and utility we let $(y_{\le i-1}, S)$ be a realizable prefix state, witnessed by $h_{i-1} \in \mathcal{C}(y_{\le i-1})$, and let $m = |S|$.

For privacy analysis fix $y_{\le i-1}$, and let S, S' be neighboring samples such that both $(y_{\le i-1}, S)$ and $(y_{\le i-1}, S')$ are realizable prefix states. By Lemma F.3, the objectives Q_i^S and $Q_i^{S'}$ are sensitivity-one and quasi-concave. Proposition F.6 therefore implies that $A_i(y_{\le i-1}; \cdot)$ is $(\varepsilon_A, \delta_A)$-differentially private on realizable input (see Definition B.3) and satisfying the privacy requirement of Assumption B.4.

Moreover Lemma F.3 gives $\max_{z \in \widetilde{\mathcal{Y}}_i(y_{\le i-1})} Q_i^S(z) = m$. Hence, with probability at least $1-\beta_A$, Proposition F.6 gives $Q_i^S(y_i) \ge m - E_{\mathrm{QC}}$.

Assume witness $h_{i-1} \in \mathcal{C}(y_{\le i-1})$ for S such $S = U(S)^{h_{i-1}}$. by the definition of Q_i^S, the prefix $y_{\le i}$ admits a completion satisfying at least $m - E_{\mathrm{QC}}$ constraints induced by S. under general position Corollary F.4 implies that exist a corresponding halfspace $g_i \in \mathcal{C}(y_{\le i})$ that disagrees on at most $E_{\mathrm{QC}} + d$ sample points. Therefore

$$\mathrm{disc}_{U(S)}(g_i, h_{i-1}) \le \frac{E_{\mathrm{QC}} + d}{m}.$$

□

Theorem F.8 (Private realizable learning of halfspaces with quasi-concave optimization)**.** *Let $\mathcal{C}$ be the class of d-dimensional halfspaces over the finite grid $\mathcal{X}$. For every $\alpha, \beta, \varepsilon, \delta \in (0,1)$, $\beta \ge \delta$there exists an (ε, δ)-differentially private proper (α, β)-PAC learner for $\mathcal{C}$ in the realizable setting with sample complexity*

$$\widetilde{O}\left(\frac{d^{5/2} + d^{3/2}2^{\log^* N}}{\varepsilon\alpha}\right).$$

Proof. Use Theorem F.5, assume $\beta_A \geq \delta_A$, with the parameterization of Theorem F.5 $\delta_A = \frac{\delta}{16ek} \leq \frac{\delta}{8k} \leq \beta_A$ with $\varepsilon_A = 4$ by Lemma F.7 the oracle A_i satisfy Assumption B.4 under general position and $M_A(\tau, \beta_A, 4, \delta_A) = \widetilde{O}\Big(\frac{2^{\log^* N}+d}{\tau}\Big)$. Hence substituting this bound into Theorem F.5, with $k = d+1$ and $\tau = \Theta(\alpha/d)$, proves the claim. □

For every prefix $y_{\leq i-1}$ and active sample S, the component oracle internally constructs the objective Q_i^S from the constraints induced by S, and constructs $\widetilde{\mathcal{Y}}_i(y_{\leq i-1})$ from the prefix and the public grid. These objects are not supplied to the oracle by CARRYFORWARDLEARNER. The privacy and failure parameters, and the number of blocks t when applicable, are fixed in advance and suppressed from the oracle input.

F.3 A COORDINATE ORACLE VIA IPCONCAVE

We instantiate the component oracle using the IPCONCAVE algorithm of Nissim et al. (2026b), with the coordinate objective depth and finite domains established in Appendix F.1.

Theorem F.9 (Cohen et al. (2023))**.** *Let $\mathcal{X}$ be a finite ordered domain. There exists an (ε, δ)-differentially private algorithm* PRIVATEIP *that on input $S \in \mathcal{X}^n$ outputs an interior point with probability $1-\beta$, provided that $n > n_{\mathrm{IP}}(|\mathcal{X}|, \beta, \varepsilon, \delta)$, for $n_{\mathrm{IP}}(|\mathcal{X}|, \beta, \varepsilon, \delta) \in O\left(\frac{\log^* |\mathcal{X}| \cdot \log^2\left(\frac{\log^* |\mathcal{X}|}{\beta\delta}\right)}{\varepsilon}\right)$.*

Algorithm 3 IPCONCAVE Nissim et al. (2026b)

Require: Dataset $S \in \mathcal{X}^n$, finite domain $\widetilde{\mathcal{X}} \subseteq \mathbb{R}$, score function $Q : \mathcal{X}^* \times \widetilde{\mathcal{X}} \to \mathbb{R}$, and parameters $\alpha, \beta, \varepsilon, \delta, t$.

1: Randomly partition S into $B_1, \ldots, B_t$, each of size at least $\lfloor n/t \rfloor$.
2: **for** $i \in [t]$ **do**
3: $\quad y_i \leftarrow \arg\max_{x \in \widetilde{\mathcal{X}}} Q(B_i, x)$.
4: **return** $\hat{x} \leftarrow$ PRIVATEIP$_{\widetilde{\mathcal{X}}, \beta, \varepsilon, \delta}(y_1, \ldots, y_t)$.

Theorem F.10 (Nissim et al. (2026b))**.** *For every $\varepsilon > 0$ and $\alpha, \beta, \delta \in (0,1)$,* IPCONCAVE *is (ε, δ)-differentially private.*

Coordinate oracle. On input $(y_{\leq i-1}; S)$, the oracle constructs Q_i^S and $\widetilde{\mathcal{Y}}_i(y_{\leq i-1})$ as above, compute $t \geq n_{IP}(|\widetilde{\mathcal{Y}}_i(y_{\leq i-1})|.\beta_A, \varepsilon_A, \delta_A)$ and returns

$$A_i^{\mathrm{IP}}(y_{\leq i-1}; S) := \text{IPCONCAVE}_{\beta_A, \varepsilon_A, \delta_A, t}\left(S, Q_i^S, \widetilde{\mathcal{Y}}_i(y_{\leq i-1})\right).$$

The following lemma replaces the generic $1/\tau^2$ approximation bound used for IPCONCAVE Nissim et al. (2026b) by a realizable VC bound with $1/\tau$ dependence.

Lemma F.11 (Realizable transfer for linear feasibility)**.** *Let $S = ((x_j, y_j))_{j=1}^n$ be a realizable labeled sample, and let B be a uniformly random submultiset of m occurrences from S. Suppose that*

$$m \geq \frac{c}{\rho}\left((d+1)\log\frac{1}{\rho} + \log\frac{1}{\beta}\right)$$

for a sufficiently large universal constant c. Then, with probability at least $1-\beta$, every parameter point $p = (a, w) \in \mathbb{R}^{d+1}$ satisfying $\mathrm{depth}_B(p) = m$ also satisfies $\mathrm{depth}_S(p) \geq (1-\rho)n$.

Proof. see I.2 □

Lemma F.12 (Realizable transfer for block maximizers)**.** *Let $(y_{\leq i-1}, S)$ be realizable prefix state, witnessed by $h_{i-1} \in \mathcal{C}(y_{\leq i-1})$, and let $m = |S|$. Suppose that every block B_j satisfies*

$$|B_j| \geq \frac{c}{\rho}\left(d\log\frac{1}{\rho} + \log\frac{t}{\beta_{\mathrm{net}}}\right)$$

for a sufficiently large universal constant c. Then, with probability at least $1-\beta_{\mathrm{net}}$, every block maximizer z_j satisfies

$$Q_i^S(z_j) \geq (1-\rho)m.$$

Proof. Fix a block B_j. Since $B_j \subseteq S$, the state $(y_{\leq i-1}, B_j)$ is realizable prefix state. By Lemma F.3,

$$\max_{z \in \widetilde{\mathcal{Y}}_i(y_{\leq i-1})} Q_i^{B_j}(z) = |B_j|.$$

Hence every block maximizer z_j admits a completion satisfying all constraints induced by B_j.

Since B_j is a uniformly random submultiset of S, its induced constraints form a uniformly random submultiset of the constraints induced by S. Lemma F.11, applied with failure probability β_{net}/t, therefore implies that the same completion satisfies at least $(1-\rho)m$ constraints induced by S. By the definition of Q_i^S,

$$Q_i^S(z_j) \geq (1-\rho)m.$$

A union bound over the t blocks proves that this holds simultaneously for every $j \in [t]$, except with probability β_{net}. □

Lemma F.13 (Coordinate oracle via IPConcave)**.** *For every $i \in [k]$, the algorithm A_i^{IP} satisfies Assumption B.4 under general position with sample requirement*

$$|S| \geq M_A(\tau, \beta_A, \varepsilon_A, \delta_A) := \widetilde{O}\left(\frac{d}{\tau} + \frac{d \log^* N}{\varepsilon_A \tau}\right).$$

Proof. For every fixed prefix, Nissim et al. (2026b, Theorem 9) implies that A_i^{IP} is $(\varepsilon_A, \delta_A)$-differentially private on arbitrary labeled samples, and hence also on realizable neighboring samples. Its output belongs to $\widetilde{\mathcal{Y}}_i(y_{\leq i-1})$,totality and validity follow directly from its definition.

For utility, suppose that $(y_{\leq i-1}, S)$ is a realizable prefix state and that $U(S)$ is in general position. Choose $\rho = \tau/2$ and set $\beta_{\mathrm{IP}} = \beta_{\mathrm{net}} = \beta_A/2$. Let y_i be the output of IPConcave.

For every $j \in [t]$, let z_j be a maximizer of $Q_i^{B_j}$. By Lemma F.12, except with probability β_{net}, simultaneously for every $j \in [t]$, $Q_i^S(z_j) \geq (1-\rho)m$. Except with probability β_{IP}, PrivateIP outputs $y_i \in [\min_j z_j, \max_j z_j]$. By Lemma F.3, Q_i^S is quasi-concave, and therefore

$$Q_i^S(y_i) \geq \min_{j \in [t]} Q_i^S(z_j) \geq (1-\rho)m.$$

A union bound shows that this holds with probability at least $1 - \beta_{\mathrm{net}} - \beta_{\mathrm{IP}} = 1 - \beta_A$.

Let $h_{i-1} \in \mathcal{C}(y_{\leq i-1})$ be a witness realizing S. By the definition of Q_i^S, the prefix $y_{\leq i}$ admits a completion $g_i \in \mathcal{C}(y_{\leq i})$ satisfying at least $(1-\rho)m$ constraints induced by S. Since $U(S)$ is in general position, Corollary F.4 gives $\mathrm{disc}_{U(S)}(g_i, h_{i-1}) \leq \rho + d/m$. Since $\rho = \tau/2$, requiring $m \geq 2d/\tau$ gives $\mathrm{disc}_{U(S)}(g_i, h_{i-1}) \leq \tau$.

Finally, PrivateIP uses $t = \widetilde{O}(\log^* N/\varepsilon_A)$ block maximizers, while Lemma F.12 requires $r = |B_j| = \widetilde{O}(d/\rho) = \widetilde{O}(d/\tau)$ examples per block. Ensuring $m \geq tr$, together with $m \geq 2d/\tau$, gives

$$M_A(\tau, \beta_A, \varepsilon_A, \delta_A) = \widetilde{O}\left(\frac{d}{\tau} + \frac{d \log^* N}{\varepsilon_A \tau}\right).$$

□

Theorem F.14 (Private realizable learning of halfspaces via IPConcave)**.** *Let $\mathcal{C}$ be the class of d-dimensional halfspaces over the finite grid $\mathcal{X}$. For every $\alpha, \beta, \varepsilon, \delta \in (0,1)$, $\beta \geq \delta$there exists an (ε, δ)-differentially private proper (α, β)-PAC learner for $\mathcal{C}$ in the realizable setting with sample complexity*

$$\widetilde{O}\left(\frac{d^{5/2} \log^* N}{\varepsilon \alpha}\right).$$

Proof. Use Theorem F.5. By Lemma F.13, with $\varepsilon_A = 4$, A_i satisfy Assumption B.4 under general position with sample complexity $M_A(\tau, \beta_A, 4, \delta_A) = \widetilde{O}\left(\frac{d \log^* N}{\tau}\right)$. Substituting this bound into Theorem F.5, with $k = d+1$ and $\tau = \Theta(\alpha/d)$, proves the claim. □

F.4 Closure under Boolean Composition

In the preceding applications, each component oracle directly selects a component while preserving a nearby completion in the current restricted global class. In other settings, however, the available primitive may only be a private learner B_i for a local class $\mathcal{H}_i$. A sample realizable by $\mathcal{C}(y_{\le i-1})$ need not be realizable by $\mathcal{H}_i$, so the realizable utility guarantee of B_i cannot be applied directly.

We use the relabeling construction of Alon et al. (2020, Algorithm 2) to bridge this gap. The construction selects a local hypothesis that admits a good global completion, relabels the sample by this hypothesis, and then applies B_i. Properness ensures that the learned hypothesis remains a valid local component, while the structural conditions below lift its local utility guarantee to a nearby hypothesis in the global class.

The following conversion was proved by Bun et al. (2015, Lemma 5.9) and is restated as Lemma 18 by Alon et al. (2020).

Lemma F.15 (PAC-to-empirical conversion)**.** *Let B be an (ε, δ)-differentially private, (α, β)-accurate PAC learner for a concept class $\mathcal{H}$, with sample complexity m. Given a realizable labeled sample of size $9m$, let $\widehat{B}$ sample m examples uniformly with replacement and apply B to the resulting sample. Then $\widehat{B}$ is an (ε, δ)-differentially private, (α, β)-accurate empirical learner for $\mathcal{H}$. Moreover, if B is proper, then so is $\widehat{B}$.*

Proposition F.16 (Component-oracle construction)**.** *Fix $i \in [k]$, a prefix $y_{\le i-1}$, and a labeled sample $S = ((x_j, y_j))_{j=1}^m$, Let B_i be the proper local $(\varepsilon_B, \delta_B)$-differentially private $(\tau, \beta_B) - accurate$ PAC learner for $\mathcal{H}_i$. For $h \in \mathcal{H}_i$, define the completion score*

$$q_i(S, h) := \min_{g \in \mathcal{C}(y_{\le i-1}, h)} \sum_{j=1}^{m} \mathbf{1}[g(x_j) \neq y_j], \tag{3}$$

Let $\widehat{B}_i$ be the proper empirical learner obtained from B_i by Lemma F.15. Define A_i by

$$(\widetilde{S}, \emptyset) \leftarrow \textsc{ARelabel}(S, \emptyset; \mathcal{H}_i, q_i), \qquad y_i \leftarrow \widehat{B}_i(\widetilde{S}),$$

and return y_i

Corollary F.17 (Local proper learners give component oracles)**.** *Suppose that every $\mathcal{H}_i$ admits a proper $(\varepsilon_B, \delta_B)$-differentially private realizable PAC learner B_i, with sample complexity $m_{B,i}(\alpha, \beta, \varepsilon_B, \delta_B)$. Assume that, for every $i \in [k]$ and every valid prefix $y_{\le i-1}$, the following conditions hold:*

1. *Every $h \in \mathcal{H}_i$ is a valid extendable component: $(y_{\le i-1}, h) \in \mathcal{Y}_{\le i}$ and $\mathcal{C}(y_{\le i-1}, h) \neq \emptyset$.*
2. *The local components cover the current restricted class: $\mathcal{C}(y_{\le i-1}) = \bigcup_{h \in \mathcal{H}_i} \mathcal{C}(y_{\le i-1}, h)$.*
3. *For every $h, h' \in \mathcal{H}_i$ and every $\bar{g} \in \mathcal{C}(y_{\le i-1}, h)$, there exists $g' \in \mathcal{C}(y_{\le i-1}, h')$ such that $\mathbf{1}[g'(x) \neq \bar{g}(x)] \le \mathbf{1}[h'(x) \neq h(x)]$ for every $x \in \mathcal{X}$.*

Then Proposition F.16 yields component oracles $A_1, \ldots, A_k$ satisfying Assumption B.4 with $\varepsilon_A = \varepsilon_B + 3$ and $\delta_A = 4e\delta_B$.

Writing $d_i = \mathrm{VC}(\mathcal{H}_i)$, it suffices to take

$$M_{A,i}(\tau, \beta_A, \varepsilon_A, \delta_A) = \widetilde{O}\left(\max\left\{ \frac{d_i + \log(1/\beta_A)}{\tau}, m_{B,i}\left(\frac{\tau}{2}, \frac{\beta_A}{2}, \varepsilon_B, \delta_B \right) \right\} \right).$$

A common sample requirement is $M_A = \max_{i \in [k]} M_{A,i}$.

Proof. See Appendix H. □

Lemma F.18 (Closure oracles for CarryForwardLearner)**.** *Let $G : \{0,1\}^k \to \{0,1\}$, let $\mathcal{H}_1, \ldots, \mathcal{H}_k \subseteq \{0,1\}^{\mathcal{X}}$ be nonempty classes, and define $\mathcal{C} = G(\mathcal{H}_1, \ldots, \mathcal{H}_k)$.*

Suppose that every $\mathcal{H}_i$ admits a proper $(\varepsilon_B, \delta_B)$-differentially private PAC learner B_i with sample complexity $M_i(\alpha, \beta, \varepsilon_B, \delta_B)$.

Then the construction of Proposition F.16 provides component oracles $A_1, \ldots, A_k$ satisfying Assumption B.4, with the privacy parameters and sample requirements stated in Corollary F.17.

Proof.

Fix a prefix $h_{\le i-1} = (h_1, \dots, h_{i-1})$. For $h \in \mathcal{H}_i$, define

$$\mathcal{C}(h_{\le i-1}, h) = \{G(h_1, \dots, h_{i-1}, h, c_{i+1}, \dots, c_k) : c_j \in \mathcal{H}_j\}.$$

Hence the structural conditions of Corollary F.17 hold immediately. Indeed, every $h \in \mathcal{H}_i$ is a valid extendable component, $\mathcal{C}(h_{\le i-1}) = \bigcup_{h \in \mathcal{H}_i} \mathcal{C}(h_{\le i-1}, h)$, and local replacement is obtained by keeping the same suffix $c_{i+1}, \dots, c_k$ and replacing only h by h'. If $h(x) = h'(x)$, the inputs to G are identical at x, and hence the two resulting global hypotheses agree there. □

Theorem F.19 (Private realizable learning of Boolean compositions)**.** *Let $G : \{0,1\}^k \to \{0,1\}$, let $\mathcal{H}_1, \dots, \mathcal{H}_k$ be nonempty classes, and define $\mathcal{C} = G(\mathcal{H}_1, \dots, \mathcal{H}_k)$. Assume that every $\mathcal{H}_i$ admits a proper private realizable PAC learner B_i with sample complexity $m_i(\alpha, \beta, \varepsilon_B, \delta_B)$. Let $d_i := \mathrm{VC}(\mathcal{H}_i)$.*

For every $\alpha, \beta, \varepsilon, \delta \in (0,1)$, $\beta \ge \delta$ there exists an (ε, δ)-differentially private (α, β)-PAC learner for $\mathcal{C}$ in the realizable setting with sample complexity

$$\widetilde{O}\left(\frac{\sqrt{k}}{\varepsilon} \max\left\{\frac{k}{\alpha} \max\left\{\mathrm{VC}(\mathcal{C}), \max_{i\in[k]} d_i\right\}, \max_{i\in[k]} m_i\left(\frac{\alpha}{80k}, \frac{\beta}{8k}, 1, \frac{\delta}{64e^2k}\right)\right\}\right).$$

Proof. Choose $\varepsilon_B = 1$, $\delta_B = \delta/(64e^2k)$, $\tau = \alpha/(40k)$, and $\beta_A = \beta/(4k)$. By Lemma F.18 and Corollary F.17, the resulting component oracles have $\varepsilon_A = 4$, $\delta_A = \delta/(16ek)$, and common sample requirement

$$M_A = \widetilde{O}\left(\max\left\{\frac{k}{\alpha} \max_{i\in[k]} d_i, \max_{i\in[k]} m_i\left(\frac{\alpha}{80k}, \frac{\beta}{8k}, 1, \frac{\delta}{64e^2k}\right)\right\}\right).$$

Applying Theorem C.1 with $v = \mathrm{VC}(\mathcal{C})$ gives

$$n = \widetilde{O}\left(\frac{\sqrt{k}}{\varepsilon} \max\left\{\frac{k\,\mathrm{VC}(\mathcal{C})}{\alpha}, M_A\right\}\right),$$

and substituting the bound on M_A proves the claim. □

G Quasi-Concavity of the Depth Function

Let S consist of affine inequalities $h_j(y) \le 0$, $j \in [m]$, over $y \in \mathbb{R}^k$, and define

$$\mathrm{depth}_S(y) := \big|\{j \in [m] : h_j(y) \le 0\}\big|.$$

We use the following standard convexity fact; see Boyd & Vandenberghe (2004, Sections 2.3.1–2.3.2).

Proposition G.1 (Convex feasible sets and convex projections)**.** *Fix $i \in [k]$, a prefix $y_{\le i-1}$, and a set $T \subseteq [m]$. Define the feasible completion set*

$$F_T(y_{\le i-1}) := \big\{(z,u) \in \mathbb{R} \times \mathbb{R}^{k-i} : h_j(y_{\le i-1}, z, u) \le 0 \text{ for every } j \in T\big\}.$$

Each constraint defines a halfspace in the remaining variables, and hence $F_T(y_{\le i-1})$ is convex. Consequently, its projection onto the z-coordinate,

$$I_T(y_{\le i-1}) := \big\{z \in \mathbb{R} : \exists u \in \mathbb{R}^{k-i} \text{ with } (z,u) \in F_T(y_{\le i-1})\big\},$$

is convex and therefore an interval, possibly empty or unbounded.

Lemma G.2 (One coordinate depth quasi concave)**.** *Fix $i \in [k]$, a prefix $y_{\le i-1}$, and a labeled sample S of size m, inducing affine constraints $h_j(y) \le 0$ for $j \in [m]$. Suppose that $(y_{\le i-1}, S)$ is realizable prefix state; that is, there exists a completion $(y_i^\star, \dots, y_k^\star)$ such that*

$$h_j(y_{\le i-1}, y_i^\star, \dots, y_k^\star) \le 0 \qquad \text{for every } j \in [m].$$

Define

$$Q_i^S(z) := \max_{y_{i+1}, \dots, y_k} \mathrm{depth}_S\big(y_{\le i-1}, z, y_{i+1}, \dots, y_k\big).$$

Then

$$Q_i^S : \mathbb{R} \longrightarrow \{0, 1, \ldots, m\}$$

is quasi-concave. Consequently, its restriction to the finite ordered domain $\widetilde{\mathcal{Y}}_i(y_{\le i-1}) \subseteq \mathbb{R}$ *is quasi-concave with respect to the induced order. Explicitly, for every* $z_1 \le z_2 \le z_3$ *in* $\widetilde{\mathcal{Y}}_i(y_{\le i-1})$*,*

$$Q_i^S(z_2) \ge \min\{Q_i^S(z_1), Q_i^S(z_3)\}.$$

Proof. For $t \in \{0, 1, \ldots, m\}$, define the superlevel set

$$L_t(y_{\le i-1}) := \{z \in \mathbb{R} : Q_i^S(z) \ge t\}.$$

The claim is immediate for $t = 0$. Fix $t \in [m]$. For every $T \subseteq [m]$ with $|T| = t$, let $I_T(y_{\le i-1})$ denote the interval from Proposition G.1. By the definition of Q_i^S,

$$L_t(y_{\le i-1}) = \bigcup_{\substack{T \subseteq [m] \\ |T|=t}} I_T(y_{\le i-1}). \tag{4}$$

Indeed, if $Q_i^S(z) \ge t$, then some completion of z satisfies at least t constraints. Hence, $z \in I_T(y_{\le i-1})$ for some $T \subseteq [m]$ of size t. Conversely, if $z \in I_T(y_{\le i-1})$, then some completion of z satisfies all constraints indexed by T, and therefore $Q_i^S(z) \ge t$.

By prefix realizability, there exists a completion $(y_i^\star, \ldots, y_k^\star)$ satisfying every constraint in S. Consequently,

$$y_i^\star \in I_T(y_{\le i-1}) \qquad \text{for every } T \subseteq [m].$$

Thus, all the intervals in (4) contain the common point $y_i^\star$. Their union is therefore an interval, so $L_t(y_{\le i-1})$ is convex.

Since Q_i^S is integer-valued, it suffices to consider integer superlevel thresholds. Hence, every superlevel set of Q_i^S is convex, and the superlevel-set characterization of quasi-concavity (Boyd & Vandenberghe, 2004, §3.4.1) implies that Q_i^S is quasi-concave on $\mathbb{R}$.

Finally, Since

$$\widetilde{\mathcal{Y}}_i(y_{\le i-1}) \subseteq \mathbb{R}$$

is equipped with the order induced by $\mathbb{R}$, every ordered triple $z_1 \le z_2 \le z_3$ in this finite domain is also an ordered triple in $\mathbb{R}$. Therefore, quasi-concavity of Q_i^S on $\mathbb{R}$ implies

$$Q_i^S(z_2) \ge \min\{Q_i^S(z_1), Q_i^S(z_3)\}.$$

Hence the restriction of Q_i^S to $\widetilde{\mathcal{Y}}_i(y_{\le i-1})$ is quasi-concave with respect to the induced order. □

H Proof of the Local Proper Learner Corollary

Corollary H.1 (Restatement of Corollary F.17)**.** *Fix* $i \in [k]$ *and a valid prefix* $y_{\le i-1}$*, and let* A_i *be the component oracle defined in Proposition F.16. Throughout the proof, we assume the structural conditions stated in Corollary F.17.*

Proof. **Totality and validity.** The algorithms ARelabel and $\widehat{B}_i$ is an empirical leaner from lemma F.15 are defined on every labeled input. Since $\widehat{B}_i$ is proper, its output satisfies $y_i \in \mathcal{H}_i$. Consequently, by the first structural condition in Corollary F.17,

$$y_{\le i} = (y_{\le i-1}, y_i) \in \mathcal{Y}_{\le i}, \qquad \mathcal{C}(y_{\le i}) \ne \emptyset.$$

Privacy. The third structural condition in Corollary F.17 implies that q_i has matched sensitivity at most one in the sense of Alon et al. (2020, Definition 24). Indeed, suppose that $h, h' \in \mathcal{H}_i$ agree on the carrier of a common subsample. Every completion of h then has a completion of h' agreeing with it on that carrier. Hence the two completion scores can differ only on the additional example, and therefore by at most one. The reverse direction follows symmetrically.

lemma F.15 learner preserves the privacy parameters of B_i. Applying Lemma 25 of Alon et al. to ARELABEL followed by the private continuation $\widehat{B}_i$ therefore gives

$$(\varepsilon_B + 3,\ 4e\delta_B) = (\varepsilon_A, \delta_A)\text{-differential privacy.}$$

This holds on arbitrary labeled inputs and, in particular, on realizable neighboring inputs as required by Assumption B.4.

Utility on realizable prefix states. Suppose

$$S = U(S)^{w_{i-1}}, \qquad w_{i-1} \in \mathcal{C}(y_{\le i-1}).$$

By the second structural condition in Corollary F.17, there exists $h_i^\star \in \mathcal{H}_i$ such that $w_{i-1} \in \mathcal{C}(y_{\le i-1}, h_i^\star)$. Since w_{i-1} realizes S, it follows from (3) that $q_i(S, h_i^\star) = 0$. hence $\min_{h_i \in \mathcal{H}_i} q_i(S, h_i) = 0$

Apply Claim 26 of Alon et al. with error parameter $\tau/2$ and failure probability $\beta_A/2$. Let h be the hypothesis selected by the exponential mechanism, then, except with probability at most $\beta_A/2$,

$$q_i(S,h) \le \min_{h_i \in \mathcal{H}_i} q_i(S,h_i) + \frac{\tau}{2}|S| \le \frac{\tau}{2}|S|$$

where

$$|S| = \widetilde{O}\left(\frac{d_i + \log(1/\beta_A)}{\tau}\right).$$

By definition $q_i(S,h)$ is the minimum disagreement of hypothesis $g_i \in \mathcal{C}(y_{\le i-1}, h)$, with that there exists $\bar{g}_i \in \mathcal{C}(y_{\le i-1}, h)$ such that disagreement $\mathrm{disc}_{U(S)}(\bar{g}_i, w_{i-1}) \le \frac{\tau}{2}$.

The relabeled sample $\widetilde{S} = U(S)^h$ is realizable by $\mathcal{H}_i$ By the empirical guarantee of the empirical learner except with probability at most $\beta_A/2$ $\widehat{B}_i$ return $h' \in \mathcal{H}_i$ such $\mathrm{err_S(h')} \le \frac{\tau}{2}$ equivalently $\mathrm{disc}_{U(S)}(h', h) \le \frac{\tau}{2}$, provided with

$$|S| \ge m_{B,i}\left(\frac{\tau}{2}, \frac{\beta_A}{2}, \varepsilon_B, \delta_B\right).$$

By the third structural condition in Corollary F.17, there exists $g_i \in \mathcal{C}(y_{\le i-1}, y_i) = \mathcal{C}(y_{\le i})$ such that

$$\mathrm{disc}_{U(S)}(g_i, \bar{g}_i) \le \mathrm{disc}_{U(S)}(y_i, h_i) \le \frac{\tau}{2}.$$

The two preceding events each fail with probability at most $\beta_A/2$. Hence, by a union bound, with probability at least $1 - \beta_A$,

$$\mathrm{disc}_{U(S)}(g_i, w_{i-1}) \le \mathrm{disc}_{U(S)}(g_i, \bar{g}_i) + \mathrm{disc}_{U(S)}(\bar{g}_i, w_{i-1}) \le \tau.$$

Thus y_i admits a completion $g_i \in \mathcal{C}(y_{\le i})$ satisfying the utility requirement of Assumption B.4. □

I HALFSPACES AND LINEAR FEASIBILITY REDUCTION

For the halfspace application, the input domain is the public finite grid $\mathcal{X} = [[\pm X]]^d \subseteq \mathbb{R}^d$, where $[[\pm X]] := \{-X, -X+1, \ldots, X\}$ and $X \in \mathbb{N}$. The grid is part of the problem specification, not computed from the private sample. Examples lie in $\mathcal{X}$, whereas the halfspace parameters (a, w) lie in $\mathbb{R}^{d+1}$. The oracles evaluate the linear constraints induced by a labeled sample; the sample itself remains labeled throughout the learner.

Linear-feasibility representation. We use the reduction of Kaplan et al. (2020b, Section 5.1), also used by Nissim et al. (2026b, Section 5.1). We use the convention $h_{a,w}(x) = 1$ if $\langle a, x\rangle \ge w$, and -1 otherwise. Every labeled example $(x, y) \in \mathcal{X} \times \{-1, 1\}$ induces the constraint

$$H_{x,y} := \{(a, w) \in \mathbb{R}^{d+1} : y(\langle a, x\rangle - w) \ge 0\}.$$

For $p \in \mathbb{R}^{d+1}$, let $r_p := \{H_{x,y} : (x, y) \in \mathcal{X} \times \{-1, 1\},\ p \in H_{x,y}\}$. This is the collection of induced constraints satisfied by p. The range family $\{r_p : p \in \mathbb{R}^{d+1}\}$ has VC dimension at most $d + 1$ by Nissim et al. (2026b, Lemma 3 and Appendix B). For $S = ((x_j, y_j))_{j=1}^n$, write $\mathrm{depth}_S(p) := |\{j \in [n] : p \in H_{x_j,y_j}\}|$. Thus depth counts satisfied constraints, including repeated occurrences. These definitions also apply to samples in $\mathbb{R}^d$, in particular to the perturbed samples below.

Lemma I.1 (Realizable halfspace to linear feasibility)**.** *Let $S = ((x_j, y_j))_{j=1}^n \in (\mathbb{R}^d \times \{-1, 1\})^n$ be realizable by $h^\star_{u,b}$, and let $S^{\mathrm{LP}} = (y_j(\langle a, x_j\rangle - w) \geq 0)_{j=1}^n$ be its induced multiset of constraints. Then $p^\star = (u, b)$ satisfies every constraint in S^{LP}, so $\mathrm{depth}_S(p^\star) = n$. Moreover, every submultiset B of m occurrences from S satisfies $\mathrm{depth}_B(p^\star) = m$.*

Proof. Since $h^\star_{u,b}$ realizes S, if $y_j = 1$, then $\langle u, x_j\rangle - b \geq 0$; if $y_j = -1$, then $\langle u, x_j\rangle - b < 0$. In both cases, $y_j(\langle u, x_j\rangle - b) \geq 0$. Hence $p^\star$ satisfies all constraints induced by S, and therefore also all constraints induced by B. □

Corollary I.2 (Restatement of lemma F.11)**.** *Let $S = ((x_j, y_j))_{j=1}^n$ be a realizable labeled sample, and let B be a uniformly random submultiset of m occurrences from S. For $\rho, \beta \in (0, 1)$, suppose that $m \leq n$ and $m \geq \frac{c}{\rho}((d+1)\log(2/\rho) + \log(1/\beta))$ for a sufficiently large universal constant c. Then, with probability at least $1 - \beta$, every $p \in \mathbb{R}^{d+1}$ satisfying $\mathrm{depth}_B(p) = m$ also satisfies $\mathrm{depth}_S(p) \geq (1 - \rho)n$.*

Proof. By Lemma I.1, a point $p^\star$ satisfies every constraint induced by S. Apply Theorem A.10 to the range family $\{r_p : p \in \mathbb{R}^{d+1}\}$, whose VC dimension is at most $d + 1$. With the stated probability, every range containing all constraints induced by B contains all but a ρ-fraction of those induced by S. For repeated examples, distinguish their occurrences as in Nissim et al. (2026b, Theorem 5, Footnote 5). This preserves the VC dimension, realizability, depth, and uniform sampling, so the same argument applies to multisets. □

Coordinate depth objective. Write the halfspace parameters in the order $y = (w, a_1, \ldots, a_d)$, with $k = d + 1$. This only reorders the entries of $p = (a, w)$. Recall from Definition F.1 that

$$Q_i^S(z) := \max_{y_{i+1}, \ldots, y_k \in \mathbb{R}} \mathrm{depth}_S(y_{\leq i-1}, z, y_{i+1}, \ldots, y_k).$$

The maximization is over real parameter completions, not over example points in $\mathcal{X}$. The finite domains below restrict the next coordinate z supplied to the oracle.

Finite coordinate domains. We use the coordinate domains of Definition F.2, with parameter order $y = (w, a_1, \ldots, a_d)$ and $\widetilde{\mathcal{Y}}_1 = \{-1, 1\}$. Excluding $w = 0$ rules out the parameter vector $(a, w) = (0, 0)$, which satisfies every induced constraint regardless of the labels. The proof below shows that some realizing representation remains.

Definition I.3 (finite coordinate domains (Kaplan et al., 2020b, Definition 4.4))**.** We define the domains $\{\widetilde{\mathcal{Y}}_i\}_{i=1}^{d+1}$ iteratively. For $i = 1$, let $\widetilde{\mathcal{Y}}_1 := \{-1, 1\}$. For $i = 2$, let $\widetilde{\mathcal{Y}}_2 := \widetilde{\mathcal{Y}}_2' / \widetilde{\mathcal{Y}}_2''$ where $\widetilde{\mathcal{Y}}_2' := [\![\pm(d \cdot d!) \cdot X^d]\!]$ and $\widetilde{\mathcal{Y}}_2'' := ([\![\pm d! \cdot X^d]\!]) \setminus \{0\}$. For $i > 2$ and given $a^*_{j-1} = s_j / t_j \in \widetilde{\mathcal{Y}}_j' / \widetilde{\mathcal{Y}}_j'' = \widetilde{\mathcal{Y}}_j$ for $j \in \{2, \ldots, i-1\}$, define $\widetilde{\mathcal{Y}}_i = \widetilde{\mathcal{Y}}_i(w^*, a_1^* \ldots, a_{i-2}^*) := \widetilde{\mathcal{Y}}_i' / \widetilde{\mathcal{Y}}_i''$ where $\widetilde{\mathcal{Y}}_i' := [\![\pm(d \cdot d!)^{i-1} \cdot X^{d(i-1)}]\!]$ and $\widetilde{\mathcal{Y}}_i'' = \widetilde{\mathcal{Y}}_i''(t_{i-1}) := ([\![\pm d! \cdot X^d]\!] \cdot t_{i-1}) \setminus \{0\}$.

Lemma I.4 (Attaining full depth on the finite domain (Kaplan et al., 2020b, Lemma 4.5, realizable case))**.** *Let $i \in \{2, \ldots, k\}$, and let $y_{\leq i-1} = (w, a_{\leq i-2})$ be generated by the domains of Definition I.3. Suppose that this prefix admits a completion satisfying all $m = |S|$ constraints in S_w^{LP}. Then there exists $y_i \in \widetilde{\mathcal{Y}}_i(y_{\leq i-1})$ admitting a completion satisfying all m constraints. Equivalently,*

$$Q_i^S(y_i) = \max_{z \in \widetilde{\mathcal{Y}}_i(y_{\leq i-1})} Q_i^S(z) = m.$$

Proof. The cited finite-domain guarantee provides a coordinate $a_{i-1} \in \widetilde{\mathcal{Y}}_i(y_{\leq i-1})$ and a full completion a attaining the maximum $\mathrm{cdepth}_{S_w^{\mathrm{LP}}}$ under the fixed prefix. This maximum equals m, since the prefix admits a completion satisfying every constraint. By Kaplan et al. (2020b, Claim 4.1),

$$\mathrm{depth}_{S_w^{\mathrm{LP}}}(a) \geq (d+1)\,\mathrm{cdepth}_{S_w^{\mathrm{LP}}}(a) - dm = m.$$

Thus a satisfies all m constraints. Setting $y_i = a_{i-1}$ proves the claim. □

Definition I.5 (Independent perturbation)**.** Let $A \subset \mathbb{R}$ be a finite nonempty set, and let $S = ((x_j, y_j))_{j=1}^n$ be a labeled sample. An independent perturbation of S from A replaces each (x_j, y_j) by $(x_j + z_j, y_j)$, where every coordinate of every $z_j \in A^d$ is sampled independently and uniformly from A. Thus only the sample points are perturbed; their labels are retained.

Lemma I.6 (General position after perturbation (Kaplan et al., 2020b, Lemma B.6))**.** *Let S be a labeled sample of n points in $\mathbb{R}^d$, possibly with repeated occurrences, and let $\beta_{\mathrm{GP}} \in (0, 1)$. Suppose that $A \subset \mathbb{R}$ is finite and* [5]

$$|A| \geq \frac{dn^{d+1}}{\beta_{\mathrm{GP}}}$$

If $\widetilde{S}$ is obtained from S by Definition I.5, then, with probability at least $1 - \beta_{\mathrm{GP}}$, $U(\widetilde{S})$ is in general position.

Lemma I.7 (Realizability under small perturbations)**.** *Let $X \in \mathbb{N}$, and let $S \in (\llbracket \pm X \rrbracket^d \times \{-1, 1\})^n$ be realizable. Choose A as in the Lemma I.6, with sample size n and parameter β_{GP}. If $\widetilde{S}$ is obtained by independently perturbing S from A, then $\widetilde{S}$ remains realizable. The same holds after a common positive rescaling of its points.*

Proof. By Kaplan et al. (2020b, Lemma B.5), there exists a halfspace realizing S with margin at least $1/X'$, where X' is the quantity specified in that lemma. Following the construction in Kaplan et al. (2020b, Theorem B.7), with the corrected resolution from Lemma I.6, set $\Delta := \left\lceil \frac{dn^{d+1}}{2\beta_{\mathrm{GP}}} \right\rceil$, $\Delta' := \left\lceil 2\Delta X' \sqrt{d} \right\rceil$, $A := \frac{\llbracket \pm \Delta \rrbracket}{\Delta'}$. Every perturbation $z \in A^d$ satisfies $\|z\|_2 \leq 1/(2X') < 1/X'$. Hence the perturbations preserve realizability, and $\widetilde{S}$ is realizable. A common positive rescaling of its points also preserves realizability. □

Corollary I.8 (Perturbation throughout the carry-forward learner)**.** *Let $\mathcal{X} \subseteq \llbracket \pm X \rrbracket^d$, and let $S \in (\mathcal{X} \times \{-1, 1\})^n$ be realizable. Obtain $\widetilde{S}$ by independently perturbing S using A from Lemma I.7, followed by a fixed positive rescaling onto an integer grid.*

Then $\widetilde{S}$ is realizable, and every active sample S_i, $0 \leq i < k$, generated by CarryForwardLearner *forms a realizable prefix state with $y_{\leq i}$. Moreover, with probability at least $1 - \beta_{\mathrm{GP}}$, $U(\widetilde{S})$ and all active samples are in general position.*

Proof. By Lemma I.7, $\widetilde{S}$ is realizable. By Lemma I.6, its points are in general position with probability at least $1 - \beta_{\mathrm{GP}}$; the rescaling preserves this property.

On this event, every carrier selected without replacement from $\widetilde{S}$ inherits general position. Relabeling leaves its points unchanged: $U(S_0) = U(R_0)$ and $U(S_i) = U(T_i)$. Finally, Lemma C.2 gives realizability of every prefix state in every execution. The same general-position event therefore covers all rounds. □

Finite coordinate domains after perturbation. We use the domains of Definition I.3 with $y = (w, a_1, \ldots, a_d)$. By Corollary I.8, every active prefix state remains realizable after perturbation and rescaling, and its sample is in general position on the stated high-probability event. Thus, for every such prefix and $i \geq 2$, Lemma I.4 guarantees an element of $\widetilde{\mathcal{Y}}_i(y_{\leq i-1})$ attaining $Q_i^S = m$. Following Lemma I.7, rescaling by Δ' places the perturbed points in $\llbracket \pm \widetilde{X} \rrbracket^d$, where $\widetilde{X} := \Delta'(X + \Delta)$. The corrected resolution increases this grid parameter by at most a factor $O(n^2)$ compared with the construction in Kaplan et al. (2020b, Theorem B.7).

Lemma I.9 (Restatement of lemma F.3)**.** *Let $i \in [k]$, and let $(y_{\leq i-1}, S)$ be a realizable prefix state with $m = |S|$, where the prefix is generated by the domains of Definition F.2. Then Q_i^S is*

[5] We use n^{d+1} in place of the n^d bound stated in Kaplan et al. (2020b, Lemma B.6). The event E_k in that proof involves $k + 1$ sample occurrences, rather than k, so the corresponding union bound must also account for the additional occurrence. With this correction, the remaining perturbation analysis is unchanged.

sensitivity-one and quasi-concave on $\widetilde{\mathcal{Y}}_i(y_{\le i-1})$. Moreover, there exists $y_i \in \widetilde{\mathcal{Y}}_i(y_{\le i-1})$ such that $y_{\le i}$ admits a completion satisfying all m constraints induced by S. Equivalently,

$$Q_i^S(y_i) = \max_{z \in \widetilde{\mathcal{Y}}_i(y_{\le i-1})} Q_i^S(z) = m.$$

Proof. Sensitivity follows from Definition F.1. By realizability and Lemma I.1, the fixed prefix admits a completion satisfying all induced constraints. Quasi-concavity therefore follows from Lemma G.2.

It remains to show that the finite domain contains a coordinate admitting a completion of depth m.

For $i = 1$, we choose the threshold w. A positive rescaling of any realizing halfspace lets us assume that $w \in \{-1, 0, 1\}$. If $w \in \{-1, 1\}$, then $w \in \widetilde{\mathcal{Y}}_1$ already, and the claim follows from Lemma I.1.

Suppose that $w = 0$, and let $c_{a,0}$ realize S. If S contains no negatively labeled point, then $c_{0,-1}$ realizes S. Otherwise, define

$$\gamma := \min_{(x,-1)\in S} \big(-\langle a, x\rangle\big) > 0,$$

and choose $L > 1/\gamma$. Every positively labeled point satisfies $\langle La, x\rangle \ge 0 > -1$, while every negatively labeled point satisfies $\langle La, x\rangle \le -L\gamma < -1$. Thus $c_{La,-1}$ realizes S. In either case, some $w \in \{-1, 1\}$ admits a realizing completion and hence a completion satisfying all m constraints by Lemma I.1.

For $i > 1$, the threshold w is fixed. Realizability and Lemma I.1 imply that the remaining coefficients admit a completion satisfying every by Lemma I.4, the finite domain $\widetilde{\mathcal{Y}}_i(w, a_{\le i-2})$ contains a coordinate a_{i-1} admitting a completion with the same maximum value. □

Returning to halfspaces.

Corollary I.10 (Depth-to-disagreement transfer)**.** *Let $(y_{\le i-1}, S)$ be a realizable prefix state witnessed by h_{i-1}, and suppose that $U(S)$ is in general position. Let $m = |S|$. If $y_{\le i}$ admits a completion $g_i = h_{a,w} \in \mathcal{C}(y_{\le i})$, with $w \in \{-1, 1\}$, satisfying at least $m - E$ induced constraints, then g_i makes at most $E + d$ classification errors on S, and*

$$\mathrm{disc}_{U(S)}(g_i, h_{i-1}) \le \frac{E + d}{m}.$$

Proof. A classification error that does not violate its induced constraint can occur only at a negatively labeled point with $\langle a, x\rangle = w$. If $a \neq 0$, general position bounds the number of such points by d; if $a = 0$, there are none because $w \neq 0$. Thus g_i makes at most $E + d$ errors on S. Since h_{i-1} realizes S, these errors are precisely the disagreements between g_i and h_{i-1} on $U(S)$. Hence $\mathrm{disc}_{U(S)}(g_i, h_{i-1}) \le \frac{E+d}{m}$. □

Utility after perturbation. Let $\widetilde{\mathcal{D}}$ denote the distribution obtained by drawing $(x, y) \sim \mathcal{D}$, independently perturbing x from A, and retaining y, before rescaling. A halfspace returned on the rescaled grid is interpreted in the original coordinates by composing it with the fixed rescaling. Its error on the perturbed distribution then equals the error of the returned halfspace on the rescaled distribution.

Corollary I.11 (Utility on the original distribution)**.** *Let $\mathcal{D}$ be a realizable distribution on the original $\mathcal{X} \times \{-1, 1\}$. Run* CarryForwardLearner *with the perturbation and rescaling of Corollary I.8 and component oracles whose utility guarantees hold on realizable prefix states in general position. Set $\beta_{\mathrm{GP}} = \beta/2$, $\beta_A, \beta_{\mathrm{LB}}, \beta_{\mathrm{VC}} \le \beta/(8k)$, and $\tau = \alpha/(80k)$, and choose m to satisfy the active-sample requirements of Theorem E.2. Then, with probability at least $1 - \beta$, the returned halfspace h_k, interpreted in the original coordinates, satisfies*

$$\mathrm{err}_{\mathcal{D}}(h_k) \le \alpha.$$

Proof. Apply Lemma I.7 to the entire finite grid, labeled by a halfspace realizing $\mathcal{D}$. The resulting realizing halfspace agrees with the retained labels under every allowed perturbation. Hence $\widetilde{\mathcal{D}}$ is realizable.

Independent perturbations preserve i.i.d. sampling. Each carrier is marginally i.i.d. from the perturbed marginal, because its indices are selected independently of the record values. Thus the all-pairs generalization argument in Theorem E.2 applies, including after rescaling and adaptive relabeling. This sampling argument is applied without conditioning on general position.

By Corollary I.8, all active samples are in general position except with probability β_{GP}. Combining this event with the oracle and generalization events in the proof of Theorem E.2, the total failure probability is at most

$$\beta_{\mathrm{GP}} + k\beta_A + k\beta_{\mathrm{LB}} + (k+1)\beta_{\mathrm{VC}} \le \beta.$$

On simultaneous success,

$$\mathrm{err}_{\widetilde{\mathcal{D}}}(h_k) \le 20(k+1)\tau \le \frac{\alpha}{2}.$$

It remains to transfer this guarantee to $\mathcal{D}$. For any (x, y) misclassified by h_k, and any $z \in A^d$, at least one of $x + z$ and $x - z$ is also misclassified with respect to the retained label y. Indeed, if both were correctly classified, their midpoint x would lie on the same correctly labeled side of the separating hyperplane. Since A is symmetric, z and $-z$ have the same probability. Thus every misclassified example remains misclassified after perturbation with probability at least $1/2$. Averaging over $\mathcal{D}$, as in Kaplan et al. (2020b, the proof of Theorem B.7), gives

$$\mathrm{err}_{\mathcal{D}}(h_k) \le 2\,\mathrm{err}_{\widetilde{\mathcal{D}}}(h_k) \le \alpha.$$

□

Proof of Theorem F.5. Apply the perturbation construction above with $\beta_{\mathrm{GP}} = \beta/2$, and run CARRYFORWARDLEARNER with target parameters $(\alpha/2, \beta/2, \varepsilon, \delta/2)$. Its prescribed parameters are therefore $\tau = \alpha/(80k)$, $\beta_A = \beta/(8k)$, $\varepsilon_A = 4$, and $\delta_A = \delta/(32ek)$.

The privacy assumptions are unchanged, so the privacy analysis of Theorem C.1 gives (ε, δ)-differential privacy. The preliminary perturbation preserves this guarantee: coupling the noise on corresponding records maps neighboring inputs to neighboring perturbed inputs.

By Corollary I.8 the preliminary perturbation preserve realizable prefix state at each iteration hence we can apply the oracle privacy and utility guarantee.

By Corollary I.11, the output, interpreted in the original coordinates, is a halfspace with population error at most α, with probability at least $1 - \beta$. Finally, substituting the chosen parameters and $\mathrm{VC}(\mathcal{C}) = O(d)$ into the sample requirement of Theorem C.1 gives the stated bound. □