Semi-private learning via low dimensional structures

Yaxi Hu ETH Zürich
Francesco Pinto University of Oxford
Amartya Sanyal ETH Zürich
Fanny Yang ETH Zürich

Abstract

Differential Privacy (DP) is the most popular statistical definition of privacy that is now widely applied for a range of Machine Learning (ML) algorithms. Roughly, DP guarantees that the output of the ML algorithm does not expose significantly more information about any single training data point than is already available from third party sources. However, this guarantee comes at a cost in terms of accuracy the algorithm. A common empirical approach to close this gap between private and non-private accuracy is through Semi-Supervised Learning (SSL) using public unlabelled data. However, a rigorous understanding of when SSL helps has been largely missing. In this work, we use the concept of compatibility functions to show how unlabelled data can be used to provide better private labelled sample complexity for learning common hypothesis classes like disjunctions and linear halfspaces. Intuitively, first, our approach uses the unlabelled data and the guarantee of *compatibility* to uncover low dimensional structures in the data. Then, private supervised training is carried out on this low-dimensional structure thereby leading to better accuracy with less private data.

On the empirical side, we show that this can be used to provide an explanation for the excellent performance of SSL in private learning of deep neural networks. We use state-of-the-art SSL algorithms like MoCov3 with ResNet50 and WideResNet on CIFAR10 and CIFAR100 to show that SSL indeed uncovers a low dimensional structure of the data — a low dimensional linear subspace where the margin is preserved. This helps in learning a linear classifier on top of the SSL representations privately with high test accuracy. We hope these insights will result in developing new algorithms for semi-supervised private learning.

1 Introduction

Machine Learning (ML) algorithms have been shown to achieve remarkable performances in a wide range of tasks including computer vision, natural language processing, and reinforcement learning. However, recent works have shown that commonly used ML algorithms are extremely vulnerable to privacy attacks [39]. Such attacks are capable of leaking sensitive user data, the algorithm is trained on. As a result, there has been a large interest towards developing ML algorithms with privacy guarantees. Differential Privacy (DP) [21] is the current de-facto standard for such guarantees. Intuitively, DP limits the sensitivity of the ML algorithm to any single data point in the dataset. However, the guarantee of differential privacy comes at a cost on the algorithm's utility. Starting from some of the earliest works in this field, the overarching message has been that nearly every problem that is learnable in the non-private setting is also learnable in the private setting, albeit with a larger sample size requirement [13, 29]. Since, then a range of work have characterised this exact cost in terms of sample complexity [8, 9, 22].

Analogous to non-private risk minimisation, a common approach for ensuring DP in the standard offline machine learning setting is through Differentially Private Empirical Risk Minimisation (DP-ERM). This line of work was started by Chaudhuri et al. [16]. For the case of DP-ERM for convex loss functions, Bassily et al. [7], Bun et al. [14] gave nearly optimal algorithms and showed the

Preprint. Under review.

necessity of a polynomial dependence of the sample size on the dimension of the problem; something that is not observed with non-private Empirical Risk Minimisation (ERM). Several works have tried to improve this dependence for specific learning problems (see Section 2 for a detailed discussion.)

Semi-Supervised Learning (SSL) [15] has shown to be incredibly effective in a wide range of tasks and is a common component of the pipeline for state-of-the-art ML algorithms models [10, 30, 34]. In the SSL framework, in addition to a small amount of labelled data S_{lab}, a large but finite amount of unlabelled data S_{unl} is available to the learning algorithm. Alon et al. [3] formally defined the notion of Semi-Private Learning (SPL) to merge the definitions of DP and SSL in the PAC sense. Intuitively, SPL protects the privacy of the labelled dataset S_{lab} but not the unlabelled data S_{unl}. They proposed a generic algorithm for semi-privately learning any infinite VC class $\mathcal{H}$ up to error α using $O\left(\frac{\text{VC}(\mathcal{H})}{\alpha}\right)$ unlabelled samples and $O\left(\frac{\text{VC}(\mathcal{H})}{\alpha}\right)$ labelled samples. However, they have two main limitations: their results are mainly applicable for infinite sized VC classes like thresholds but not finite sized classes like disjunction. More importantly, their results are distribution agnostic and thus cannot adapt to "nicer" distributions to provide better labelled sample complexity.

We show how to overcome this problem with the simple yet elegant notion of compatibility function, introduced in Balcan and Blum [4]. Intuitively, for a hypothesis class $\mathcal{H}$, the notion of compatibility dictates some kind of agreement that the ground truth hypothesis should have with the underlying data. Their characterisation has been used, implicitly, in a wide range of works to show the benefits of SSL algorithms [5, 23, 25, 44] in non-private learning. In this work, we extend this notion to the case of SPL for two important hypothesis classes: disjunctions and linear halfspaces. We present the first results showing how compatibility functions can be used to exploit the underlying low-dimensional structure of the data to yield low labelled private sample complexity.

We also present experimental results using state-of-the art deep neural networks on common vision datasets (CIFAR10 and CIFAR100). We show that a standard contrastive training algorithm (MoCov3 [17]), trained on ImageNet yields representations on CIFAR10/CIFAR100 that preserves a large margin even when projected on to a low dimensional subspace. This is in fact the same notion of compatibility we use to prove our theoretical results for learning linear halfspaces. This, perhaps provides a partial explanation for why SSL representations obtains competitive accuracies under private training in practice [19, 32, 40, 45]. Shi et al. [38] suggests that when the pre-training and the downstream tasks are aligned, supervised pre-training yields better results, in terms of downstream non-private accuracies compared to SSL representations. We show that this intuition carries over to private learning as well.

Contributions In summary, we have the following contributions.

- We prove the first results showing how exploiting compatibility functions in conjunction with underlying low-dimensional structures in the data can lead to better private labelled sample complexity.
- We provide concrete examples of this for two hypothesis classes - disjunctions and linear halfspaces.
- We show that this theory provides a partial explanation for why pretraining yields large improvements in practice for private learning. In addition, we also discuss the difference in the use of semi-supervised and supervised pre-training.

2 Related work and Preliminaries

Differentially Private Empirical Risk Minimisation (DP-ERM) Chaudhuri et al. [16] provided the first generic algorithms for conducting Empirical Risk Minimisation (ERM) while maintaining Differential Privacy (DP). In the case of non-private ERM for convex loss functions, it is well known that gradient based methods enjoy dimension independent rates of convergence. However, the seminal work of Bassily et al. [7] showed that, for convex DP-ERM, the excess empirical risk necessarily suffers polynomially on the dimension of the problem. This dimension dependence has been successfully avoided for various special cases. For generalised linear models with strongly convex regularisation, Jain and Thakurta [27] proved a dimension independent bound for DP-ERM. A stronger dimension independent bound for the case of large margin halfspaces was recently showed by Lê Nguyên et al. [31]. Their algorithm uses the standard technique of random projections [42] to reduce the dimensionality of the problem to $O\left(1/\gamma\right)$, where γ is the margin of the problem. In this work, we show that using unlabelled data, we can exploit further lower dimensional structures thereby reducing the sample complexity even further.

Semi-Supervised Learning (SSL) Balcan and Blum [4] defined the notion of compatibility function and used it to understand what properties of the data distribution enables SSL algorithms to learn with fewer labelled samples. Many different types of compatibility have been studied in the literature — feature independence [12, 18] and weak label dependence [11] for co-training, expansion [5, 44], and two-sided disjunctions [6] among others. Göpfert et al. [24] suggested that such a notion is, in fact, necessary for SSL to reduce the labelled sample complexity. From a causal perspective, this was echoed in Schölkopf et al. [37] who showed that SSL algorithms provided an edge of supervised learning only when the conditional distribution of the label given input $\mathbb{P}[Y|X]$ was not independent of the marginal distribution $\mathbb{P}[X]$, something that is implied by compatibility functions as well.

Semi-Private Learning (SPL) Recently, there has been a surge in empirical works showing the benefits of SSL for private learning. As mentioned in Section 1, this has been defined in Alon et al. [2] as SPL. Tramer and Boneh [40] argued, empirically, that for differentially private learning to compete with non-private supervised learning, the algorithm either needs more training data or better features. They made the interesting observation that hand-crafted features from ScatterNet [35] provided a better privacy-accuracy trade-off compared to end-to-end supervised training. However, their work attributed the difference mainly to optimisation issues and without stressing on the need to "learn" better representations. De et al. [19] provided a number of architectural and optimisation tricks to yield state-of-the-art accuracies with SSL and supervised pre-training. Similar benefits for NLP tasks have been observed with large language models in Li et al. [32], Yu et al. [45].

2.1 Preliminaries

Before introducing the theoretical results, we will first define the relevant notions. Defined in Dwork et al. [20, 21], a learning algorithm $\mathcal{A}$ is said to be (ϵ, δ)-DP, if for any two datasets S, S' that differ in exactly one entry and for any output set Q in the range of the algorithm $\mathcal{A}$,

$$\mathbb{P}_{h\sim\mathcal{A}(S)}[h(x) \in Q] \leq e^{\epsilon}\mathbb{P}_{h\sim\mathcal{A}(S')}[h(x) \in Q] + \delta$$

Balcan and Blum [4] proposed the notion of compatibility (defined in Definition 1) between a hypothesis and a data distribution and used it to give PAC style guarantees for SSL. They argued that, if the notion of compatibility is satisfied, unlabelled data can be used to refine the hypothesis class to a smaller set that only contains hypothesis "compatible" with the underlying data distributions.

Definition 1 (Compatibility)**.** *Let $\mathcal{X}$ be an instance space, $\mathcal{H}$ be a hypothesis class, and D_X be the marginal data distribution over $\mathcal{X}$. A compatibility score of a classifier at a point $x \in \mathcal{X}$ is defined as $\chi : \mathcal{C} \times \mathcal{X} \to [0, 1]$. Then, the compatibility between the classifier h and the distribution D_X is*

$$\chi(h, D_X) = \mathbb{E}_{x\sim D_X}[\chi(h, x)]$$

Next, in Definition 2, we define the set of compatible distributions.

Definition 2 (Compatible distributions)**.** *For a compatibility function χ and hypothesis class $\mathcal{H}$, define $\mathcal{D}_{\chi,\mathcal{H}}$ as the set of corresponding compatible distributions if for all $D \in \mathcal{D}_{\chi,\mathcal{H}}$, $\exists f \in \mathcal{H}$ such that $\chi(f, D_X) = 1$ and $\mathbb{P}_{(x,y)\sim\mathcal{D}}[f(x) \neq y] = 0$ where D_X is the marginal distribution of D.*

Now, we are ready to provide a formal definition for private semi-supervised learnability in the PAC sense [29, 41] with respect to a family of distributions. Let $\mathcal{D}$ be a family of distributions over $\mathcal{X} \times \mathcal{Y}$.

Definition 3 ($(\alpha, \beta, \epsilon, \delta)$-Private semi-supervised PAC learnability on a family of distributions $\mathcal{D}$)**.** *For any $\alpha, \beta, \delta \in (0, 1)$, $\epsilon > 0$, the hypothesis class $\mathcal{H}$ is $(\alpha, \beta, \epsilon, \delta)$-private semi-supervised PAC learnable on $\mathcal{D}$ if there exists an (ϵ, δ)-DP algorithm $\mathcal{A}$ such that for any distribution $D \in \mathcal{D}$, given an unlabelled dataset S_{unl} of size n_{unl} sampled from D_X and a labelled dataset S_{lab} of size n_{lab} sampled from D, $\mathcal{A}$ outputs a hypothesis h satisfying*

$$\mathbb{P}[\mathbb{P}_{x,y}(h(x) \neq y) \leq \alpha] \geq 1 - \beta$$

where the first probability is over the sampling of S_{unl}, S_{lab}, and the intrinsic randomness of the learning algorithm $\mathcal{A}$. In addition, both n_{unl} and n_{lab} should depend polynomially on the size of $\mathcal{X}$ and the PAC parameters $\frac{1}{\alpha}, \frac{1}{\beta}$. n_{lab} should also be bounded by polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}$.

Here, n_{unl} is also referred to as the unlabelled sample complexity and n_{lab} is referred to as the private labelled samplexity.

3 Theoretical Results

In this section, we provide sample complexity bounds for private semi-supervised learning of two hypothesis classes. We consider a finite hypothesis class, disjunctions, and the infinite hypothesis class of linear halfspaces. Using different compatibility functions for disjunctions and linear halfspaces, we show results for both of them. Finally, we also discuss and compare our results with existing works to show the advantage of compatibility functions for private semi-supervised learning. For both settings, we will consider binary classification *i.e.*$\mathcal{Y} = \{-1, 1\}$.

3.1 Disjunction

Let $\mathcal{X}$ be the d-dimensional boolean hypercube $\{0,1\}^d$. We define the class of k-literal disjunctions DISJ_d^k as the set of functions of the form $f(x) = x[i_1] \vee x[i_2] \vee ... \vee x[i_k]$ where $i_j \in [d]$. For instance, $f = x[1] \vee x[3] \in \text{DISJ}_3^2$ such that $f((1,0,0)) = 1$ and $f((0,1,0)) = -1$. For a disjunction $f = x[i_1] \vee ... \vee x[i_k] \in \text{DISJ}_d^k$, denote the set of positive indicators as $V^+(f) = \{x[i_1], ..., x[i_k]\}$. For an instance $x = (x[1], ..., x[d]) \in \mathcal{X}$, denote the set of active indicators in x as $\hat{V}^+(x) = \{x[i] : x[i] = 1\}$. We define the compatibility function for disjunction as

$$\chi^{\text{DISJ}}(f, D_X) = \mathbb{E}_{x \sim D_X}\left[\mathbb{1}\{\hat{V}^+(x) \subset V^+(f) \text{ or } \hat{V}^+(x) \cap V^+(f) = \emptyset\}\right] \tag{1}$$

The compatibility imposes a separation between the set of positive indicators and the remaining variables. A disjunction f is compatible with a distribution if the support of the distribution includes no example that contains active variables from both $V^+(f)$ and $[d] \setminus V^+(f)$.

Our first result shows the sample complexity of private semi-supervised learning (Definition 3) of disjunctions DISJ_d^k. In particular, we show this for compatible distributions (Definition 2) as a function of a lower dimensional structure. We refer to this low dimensional structure as the component graph of the distribution. For a distribution D, the component graph is defined as $G_D = (V, E_D)$ where each node corresponds to one of the d variables i.e. $V = \{1, \ldots, d\}$ and E_D contains an edge $(x[i], x[j])$ if and only if $x[i] = x[j] = 1$ for some x in the support of the distribution D. Similarly, let $\hat{G}_S = (V, \hat{E}(S))$ be the empirical component graph where $\hat{E}$ contains an edge $(x[i], x[j])$ if and only if $x[i] = x[j] = 1$ for some example $x \in S$. We denote the probability $\mathbb{P}_{x \sim D}[x[i] = x[j] = 1]$ as $p_{i,j}^D$. Note that the empirical component graph $\hat{G}_S$ is a random object where the edge (i, j) exists with probability $1 - (1 - p_{i,j}^D)^{|S|}$. For a family of distributions $\mathcal{D}$, the minimum positive edge probability $p_{min}^{\mathcal{D}}$ is defined as $(p_{min}^{\mathcal{D}} = \min_{D \in \mathcal{D}, i \neq j, p_{i,j}^D > 0} p_{i,j}^D$.

Theorem 1. *Let $\mathcal{D}$ be the set of compatible distributions (defined in Definition 2) with respect to the hypothesis class DISJ_d^k and the compatibility function χ^{DISJ}. For $\alpha, \beta \in (0,1)$, $\epsilon, \delta > 0$, DISJ_d^k can be $(\alpha, \beta, \epsilon, \delta)$-private semi-supervised PAC learned on the family of distributions $\mathcal{D}$ with*

$$n_{unl} = O\left(\frac{\log \frac{2d^2}{\beta}}{-\log\left(1 - p_{\min}^{\mathcal{D}}\right)}\right), n_{lab} = O\left(\frac{1}{\alpha\epsilon}\left(Z_{\max} + polylog(\frac{1}{\beta}, \frac{1}{\delta})\right)\right).$$

Here, $Z_{\max}$ is the maximum number of components in the component graph for distributions in $\mathcal{D}$ and $p_{\min}^{\mathcal{D}}$ is as defined above.

We provide a detailed proof in Appendix B.1 but provide a short sketch below. In addition, we provide a tighter distribution specific result in Theorem 2 where we can bound $Z_{\max}$.

Proof sketch Our algorithm first uses the unlabelled data to construct the empirical component graph. For any distribution $D \in \mathcal{D}$ with marginal distribution D_X, every edge (i, j) in the component graph of distribution exists in the empirical component graph with edge probability $1 - (1 - p_{i,j}^{D_X})^{n_{\text{unl}}} \leq 1 - \frac{\beta}{d^2}$, where the last inequality follows from substituting the value of n_{unl} and $p_{\min}^{\mathcal{D}}$. By applying the union bound, we show that the empirical component graph is the same as the distributional component graph with high probability thus reducing the effective size of DISJ_d^k from $\binom{d}{k}$ to $2^{Z_{\max}}$. The bound on n_{lab} then follows from the generic algorithm in Kasiviswanathan et al. [29].

Next, make we propose a family of distributions with two components in the component graph for disjunctions. This allows us to both provide a tighter analysis for unlabelled sample complexity using results from Karger [28], and also provides a concrete instantiation of Theorem 1. Let χ^{DISJ} and DISJ_d^k be as defined above. Then, for $p > 0$, we define a new family of compatible distributions $\mathcal{D}_p$.

Definition 4. *For $p \in (0,1), f \in \text{DISJ}_d^k$, define the joint probability $D_{f,p}(x,y) = \mathbb{P}(x|y)\mathbb{P}(y)$. Further, let $\mathbb{P}(y=1) = \mathbb{P}(y=-1) = \frac{1}{2}$ and assume that the variables $V^+(f)$ and $V^-(f)$ are conditionally independent, i.e. $\mathbb{P}[x|y] = \mathbb{P}[V^+(f)|y]\,\mathbb{P}[V^-(f)|y]$. Next, we define $\mathbb{P}[V^+(f)|y=1]$ and $\mathbb{P}[V^-(f)|y=1]$ through the following sampling algorithm after sampling y from $\mathbb{P}(y)$.*

- *If $y=1$, sample an Erdos-Renyi random graph G^+ on vertices $V^+(f)$ with edge probability p.*
- *Label the isolated vertices in G^+ as 0 and all other variables as 1*[1].
- *Given $y=-1$, sample a similar random graph G^-, as above, with vertices $V^-(f)$.*

Finally, we ensure compatibility by setting $\mathbb{P}[V^-(f)=0|y=1] = \mathbb{P}[V^+(f)=0|y=-1] = 1$.

Another way to understand the sampling process in Definition 4 is that each random graph corresponds to one instance $x \sim D$ with the label of the vertices deciding whether the corresponding variable belongs to the set of active variables. In Theorem 2, we show that for this family of distributions, with sufficient unlabelled examples, we can reduce the labelled sample complexity to $O\left(\frac{1}{\epsilon\alpha}\log\frac{1}{\beta}\right)$.

Theorem 2. *For $p \in (0,1)$, $k,d > 35$, $\alpha \in (0,1), \beta \in \left(4\exp\left(-\frac{d-5}{9}\right), \frac{4}{d}\right)$, $\epsilon, \delta > 0$, DISJ_d^k is $(\alpha,\beta,\epsilon,\delta)$-private semi-supervised PAC learnable with compatible distributions $\mathcal{D}_p$ with*

$$n_{unl} \geq \max\left\{\frac{\log\left(1-\frac{9\log\frac{16}{\beta}+4}{d-1}\right)}{\log(1-p)}, 8\log\frac{16}{\beta}\right\}, n_{lab} \geq O\left(\frac{1}{\alpha\epsilon}\left(1+polylog(\frac{1}{\beta},\frac{1}{\delta})\right)\right).$$

The full proof is provided in Appendix B.2. Note that the second term in the unlabelled sample complexity (n_{unl}) in Theorem 2 is always smaller than n_{unl} in Theorem 1. Also, for a fixed edge probability, the first term in n_{unl} in Theorem 2 decreases at the rate of $O\left(1/\sqrt{d}\right)$, while n_{unl} in Theorem 1 increases at the rate of $O(\log d)$. Thus, for sufficiently large d, Theorem 2 provides a much tighter unlabelled sample complexity for the family of compatible distributions defined in Definition 4.

Theorem 2 implies a seemingly unexpected inverse relationship between the unlabelled sample complexity and the dimension d. However, this could be explained by the properties of the random graph in the data generation process for distributions defined in Definition 4. For a connected random graph with moderate dimention d and edge probability $\tilde{p}$, the probability that an extra node disconnects the graph decreases exponentially in d. This implies an inverse relationship between sufficient $\tilde{p}$ and d to keep a graph connected upon addition of nodes. As the edge probability $\tilde{p}$ in the empirical component graph increases with n_{unl}, the unlabelled sample complexity decreases with d.

3.2 Linear Halfspace

Let the instance space $\mathcal{X} = B_2^d = \{x \in \mathbb{R}^d : ||x||_2 = 1\}$ be the d-dimensional unit ball [2]. We define the class of d-dimensional linear halfspaces $\mathcal{H}_L^d = \left\{f_w \mid f_w(x) = \text{sign}(w^T x), w \in B_2^d\right\}$ where $x \in B_2^d$. For $\gamma \in (0,1)$, we define the compatibility function with parameter γ as

$$\chi_\gamma(w^*, D_X) := \mathbb{E}_{x \in D_X}\left[\mathbb{1}\left\{|\langle w^*, x\rangle| \geq \gamma\right\}\right] \tag{2}$$

If f_w obtains zero classification error on D, then the compatibility function $\chi_\gamma(w,D)$ is the probability that the function f_w has a margin of γ for the distribution D. In Theorem 3, we present the sample complexity for private semi-supervised learning of linear halfspace $\mathcal{H}_L^d$ with a family of compatible distributions, given the data lies approximately in a low-dimensional space. Note that the definition of compatible distributions implies that f obtains zero classification error. Theorem 3 shows a labelled sample complexity that is independent of the dimension d.

Theorem 3. *Let $\mathcal{D}_d$ be the family of distributions such that any $D \in \mathcal{D}_d$ over $\mathcal{X} \times \mathcal{Y}$ satisfies*

- *There exists $w^* \in B_2^d$ such that $f_{w^*} = sign(\langle w^*, x\rangle)$ is compatible with D and accurate,* i.e.$\chi_\gamma(w^*, D) = 1$ *and* $\mathbb{P}_{(x,y)\sim D}[f_{w^*}(x) \neq y] = 0$.

[1] If G^+ is empty, uniformly randomly label one variable from $V^+(f)$ as 1 and all others as 0.

[2] However, we can always assume that $\mathcal{X} = \mathbb{R}^d$ and normalize every example to the unit ball

- *Let* $\Sigma_X = \mathbb{E}_{x\sim D_X}\left[xx^\top\right]$ *be the covariance matrix and* $\lambda_1(\Sigma_X), \ldots, \lambda_d(\Sigma_X)$ *its eigenvalues in descending order. Then, there exists a* $k \ll d$*-dimensional approximation of* Σ_X*,* i.e.$\sum_{i=k+1}^{d} \lambda_i(\Sigma_X) \leq \eta$*.*

Then, the hypothesis class $\mathcal{H}_L^d$ *of linear halfspaces of dimension* d *is* $(\alpha, \beta, \epsilon, \delta)$*-private semi-supervised learnable on* $\mathcal{D}_d$ *with sample complexity*

$$n_{unl} = O\left(\frac{kd}{\beta\gamma^2}\right), \quad n_{lab} = O\left(\frac{\sqrt{k}}{\alpha\epsilon\left(\gamma - \sqrt{\frac{\eta}{\beta}}\right)}\right) \tag{3}$$

Proof sketch We consider a very simple and intuitive algorithm that applies Principal Component Analysis (PCA) on the unlabelled dataset and then projects the labelled data into the low-dimensional space, identified with PCA. Finally, we learn the linear halfspace in the low-dimensional space with private-SGD using ramp loss([7]). The approximately low rank property and the compatibility of $\mathcal{D}_d$ ensures that, with sufficient unlabelled data, the transformation by PCA preserves a margin of $O(\gamma - \sqrt{\eta/\beta})$ in the low-dimensional space with high probability. Finally, the sample complexity bound for private-SGD gives the desired dimension-independent labelled sample complexity.

3.3 Comparison with existing works

Comparison with generic algorithms [2, 7] Directly applying private-SGD by Bassily et al. [7] on the original feature space to achieve (ϵ, δ)-DP requires $n_{\text{lab}} = O\left(\sqrt{d}/\alpha\epsilon\right)$. The generic algorithm for SPL proposed by Alon et al. [2] reduces the infinite hypothesis class to a finite α-net of the hypothesis space using unlabeled data. It enforces $(\epsilon, 0)$-DP and achieves a labeled sample complexity $O\left(d/\alpha\epsilon\right)$. Our algorithm significantly outperforms the above generic algorithms when $d \gg k$.

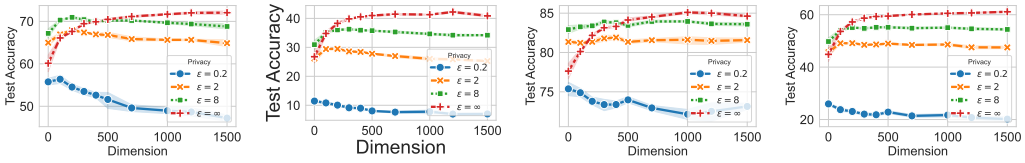
Comparison with other dimension reduction techniques [31] Lê Nguyên et al. [31] introduced an efficient private algorithm for learning large-margin linear halfspaces that avoids the dependence of the labelled sample complexity on the data dimension d. The algorithm first applies the Johnson-Lindenstrauss transformation to reduce the dimension of the feature space from d to $O\left(1/\gamma\right)$ while preserving the margin in the transformed space with high probability. Private learning the reduced hypothesis class with margin $O(\gamma)$ requires labelled sample complexity $O\left(1/\alpha\epsilon\gamma^2\right)$, which degrades rapidly with a smaller margin. Our algorithm removes the quadratic dependence on the margin but pays the price of requiring the data to lie approximately in a low-dimensional space.

Comparison with non-private learning It is interesting to note that our algorithm may not lead to a similar improvement in the non-private case. We present a rough argument here. Denote the best hypothesis in the k-dimensional space by $\hat{h}^* \in \mathcal{H}_L^k$, and the empirical risk minimizer in the k-dimensional space by $\hat{h}$. We can decompose the error of $\hat{k}$ into three parts, i.e.

$$R(\hat{h}) = \underbrace{[R(\hat{h}) - R_n(\hat{h})]}_{\text{Generalisation gap}} + \underbrace{[R_n(\hat{h}) - R_n(\hat{h}^*)]}_{\text{Empirical excess risk}} + \underbrace{R_n(\hat{h}^*)}_{\text{Approximation Error}} .$$

The first and the second part are the standard *generalisation gap* and the *empirical excess risk* usually bounded by uniform convergence and optimisation analysis, whereas the third part $R(\hat{h}^*)$ correspond to the *approximation error* incurred due to projecting the data onto a low dimensional space. A simple use of Rademacher complexity can yield dimension-independent bound for the first term though there remains some hidden dependence. For the second term, Bassily et al. [7] implies a $O\left(\sqrt{k}\right)$ dependence on dimension. The third term necessarily increases with decreases k.

For private learning, the second term decreases but the third term increases as k decreases. This implies that there is an optimal k where we hope to get the smallest error.For non-private learning, the empirical excess risk is essentially independent of k while the first term has a small dependence on k. Thus, the approximation error dominates and the error decrease with the k. We confirm both of these hypotheses in our experimental results in Section 4.



(a) SSL (CIFAR10) (b) SSL (CIFAR100) (c) SL (CIFAR10) (d) SL (CIFAR100)

Figure 1: Figure 1a and 1b shows the change in test accuracy with the PCA projection dimension for SSL representations. Figure 1c and 1d shows the same for representations obtained using SL.

4 Experimental Results

The previous sections showed specific examples of compatible functions and low dimensional structures for two important hypothesis classes. In this section, we investigate whether the concepts carry over to real world Machine Learning (ML) algorithms and in particular deep neural networks. The low dimensional structure we use in our experiments lie in the space of representations obtained with a forward pass through the network before the application of the last linear layer. Existing works [35, 36] have already suggested that these representations are approximately low rank.

Experimental Setting We will refer to the neural networks used to generate representations as the feature extractor and the linear classifier trained on top as the linear probe. We consider two ResNet50 [26] feature extractors for our experiments — one SSL feature extractor trained using MoCov3 (a contrastrative learning framework) on ImageNet-100, a subset of ImageNet [17] and a Supervised Learning (SL) feature extractor trained on ImageNet using the cross-entropy classification loss. Note that the SSL feature extractor does not have access to the ImageNet labels whereas the SL one does. We fix the feature extractor and perform linear probing on CIFAR10 and CIFAR100. Further details of experiments are provided in Appendix D.

4.1 Exploiting the low rank structure of SSL representations

In Section 3.3, we discussed that if the data satisfies a large margin in a low dimensional space, it gives a larger boost to private training than it does for non-private training. In this section, we verify this in practice. Using the SSL pre-trained ResNet50 feature extractor, we first extract 2048 dimensional features for the entire CIFAR10 and CIFAR100 datasets. Then, we randomly partition the training sets into 45,000 private training examples and 5,000 public training examples and leave the test set untouched. We apply PCA on the public train set to obtain the top k principal components, for both CIFAR10 and CIFAR100 and use it to project the private train set and the test set along the top k respective components for both datasets. Finally, we train a linear classifier using DP-SGD [1] on this reduced dimensionality private dataset and evaluate it on the reduced dimensionality test set. Our results for CIFAR10 and CIFAR100 are reported in Figure 1a and 1b respectively.

We observe an interesting phenomenon: under strict privacy constraints *i.e.* small ϵ, as the dimensionality (*i.e.* k) decreases, the private test accuracy increases. For CIFAR10, the maximum private test accuracy at $\epsilon = 0.2, k = 100$ is greater than 0.55. while it is only 0.46 at a dimension of 1500, which is nearly a increase of 20%. For CIFAR100, there is almost an 90% increase in test accuracy by reducing the dimensionality. This is consistent with our theory that suggests that the private labelled sample complexity can be decreased *i.e.*private test accuracy can be increased, by projecting the data to smaller dimensions. While this phenomenon is especially prominent for very small $\epsilon = 0.2$, we observe a more nuanced phenomenon for moderate $\epsilon \in \{2, 8\}$. Reducing the dimensionality helps up to a point ($k \approx 200$), after which further reduction in the dimension deteriorates the private test accuracy. This can be understood as the gain (due to small k) in the numerator in n_{lab} in Equation (3) is offset by a large decrease in the margin (*i.e.*increase in η) in the denominator. In the non-private case, we can see that reducing the dimensionality hurts the test accuracy consistently, which is due to the fact that the margin is gradually degraded by decreasing dimension. We discussed this in Section 3.3. The practical takeaway is that when very small ϵ is desired, which is the case if we want to protect from privacy attacks, there is a benefit in considering this training pipeline: SSL+PCA.

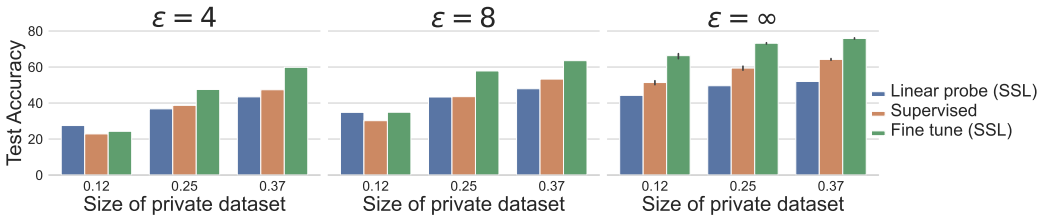
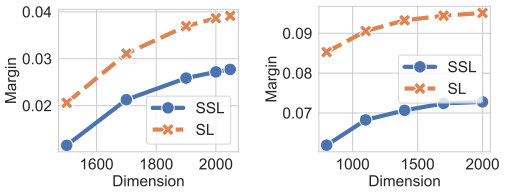


Figure 2: Plots the private CIFAR100 (coarse) test accuracy of a WideResNet 16-4 against varying sizes (fraction of the full dataset) of private labelled dataset. Linear Probe (SSL) and Finetune (SSL) refers to retraining linear classifier and finetuning the entire model, including the SSL-trained feature extractor, on the private CIFAR100 dataset respectively. Supervised indicates private training from scratch. Interesting for small ϵ and small size (split=0.12), linear probing performs best while for larger datasets and larger ϵ, fine-tuning performs best.

4.2 Exploiting the low rank structure of SL pretrained representations

We follow a similar procedure in this section except that the feature extractor is instead trained using classical supervised pre-training on ImageNet. The private test accuracies for CIFAR10 and CIFAR100, reported in Figure 1c and 1d respectively, shows that SL pre-training significantly outperforms SSL pre-training for all values of ϵ (Figure 1a and 1b). This is explained by the fact that the margins of SL representations are larger than SSL representations as shown in Figure 3 for both CIFAR10 and CIFAR100. This better performance of SL pre-training, compared to SSL pre-training, was also observed in De et al. [19] for private learning. While this might suggest that SL pre-training is perhaps always superior to SSL pre-training, experiments in Shi et al. [38] suggest that this is the case only when the pre-training and the downstream tasks are aligned.

SSL fine-tuning Another observation in literature is that privately fine-tuning the whole SSL model (backbone and linear classifier) performs better than just training the linear head. While the main focus of our work is on understanding the latter, we note that SSL fine-tuning has certain drawbacks. First, it is often computationally intensive to be able to train/fine-tune large NN architectures privately due to various training peculiarities of DP-SGD *e.g.*requirement of large batch sizes. Second, as seen in Figure 2 for strong privacy requirements with severe lack of data, SSL linear probing outperforms SSL fine-tuning.



CIFAR10 CIFAR100

Figure 3: Plot margins of low dimensional SSL and SL projections of CIFAR10 and CIFAR100 respectively.

5 Future work and Conclusion

Our results show that a general rule-of-thumb is to use SL pre-training when we know that downstream task and the available labelled pre-training dataset is aligned. However, it would be interesting to be able to characterise the maximum divergence between the pre-training and downstream distribution notwithstanding which SL pre-training outperforms SSL pre-training. Second, our experiments show there is a optimal dimensionality of projection for achieving the best private test accuracy. However, this depends on the inherent margin, the spectral distribution and the amount of available data, as well as the privacy parameters. Providing an easy-to-use technique for computing this optimal dimension would be useful. Finally, in our experiments, we performed a PCA on a left out "public" portion of the training dataset. However, other possible approaches including computing the principal components on the pre-training dataset or executing private PCA on the downstream task might be practically more relevant.

To conclude, we show that unlabelled data can provably benefit private learning for a large class of ML problems. The driving principle behind this is that SSL algorithms can use the unlabelled data to uncover some hidden low dimensional structures. We provide experimental results to argue that this might be the reason why SSL approaches have been so successful, in practice, for private learning.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[2] Noga Alon, Raef Bassily, and Shay Moran. Limits of private learning with access to public data. In *Conference on Neural Information Processing Systems*, 2019.

[3] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite Littlestone dimension. In *ACM Symposium on Theory of Computing*, 2019.

[4] Maria-Florina Balcan and Avrim Blum. A discriminative model for semi-supervised learning. *J. ACM*, 2010.

[5] Maria-Florina Balcan, Avrim Blum, and Ke Yang. Co-training and expansion: Towards bridging theory and practice. *Conference on Neural Information Processing Systems*, 17, 2004.

[6] Nina Balcan, Christopher Berlind, Steven Ehrlich, and Yingyu Liang. Efficient semi-supervised and active learning of disjunctions. In *International Conference on Machine Learning*, 2013.

[7] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Annual Symposium on Foundations of Computer Science*, 2014.

[8] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In *Innovations in Theoretical Computer Science*, 2013.

[9] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. 2013.

[10] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Conference on Neural Information Processing Systems*, 2019.

[11] Avrim Blum and Yishay Mansour. Efficient co-training of linear separators under weak dependence. In *Conference on Learning Theory*, 2017.

[12] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Conference on Learning Theory*, 1998.

[13] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005.

[14] Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *ACM Symposium on Theory of Computing*, 2014.

[15] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. 2006. *The MIT Press*, 2006.

[16] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 2011.

[17] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. 2021.

[18] Sanjoy Dasgupta, Michael Littman, and David McAllester. Pac generalization bounds for co-training. *Conference on Neural Information Processing Systems*, 2001.

[19] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

[20] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology - EUROCRYPT*. 2006.

[21] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, 2006.

[22] Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *Conference on Learning Theory*, 2014.

[23] Spencer Frei, Difan Zou, Zixiang Chen, and Quanquan Gu. Self-training converts weak learners to strong learners in mixture models. 2022.

[24] Christina Göpfert, Shai Ben-David, Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, and Ruth Urner. When can unlabeled data improve the learning rate? In *Conference on Learning Theory*, 2019.

[25] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Conference on Neural Information Processing Systems*, 2021.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[27] Prateek Jain and Abhradeep Guha Thakurta. (near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, 2014.

[28] David R. Karger. Random sampling in cut, flow, and network design problems. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, 1994.

[29] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 2011.

[30] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2016.

[31] Huy Lê Nguyên, Jonathan Ullman, and Lydia Zakynthinou. Efficient private algorithms for learning large-margin halfspaces. In *Algorithmic Learning Theory*, 2020.

[32] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2021.

[33] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Annual Symposium on Foundations of Computer Science*, 2007.

[34] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[35] Edouard Oyallon, Stéphane Mallat, and Laurent Sifre. Generic deep networks with wavelet scattering. *arXiv preprint arXiv:1312.5940*, 2013.

[36] Amartya Sanyal, Varun Kanade, Philip HS Torr, and Puneet K Dokania. Robustness via deep low-rank representations. *arXiv preprint arXiv:1804.07090*, 2018.

[37] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *International Conference on Machine Learning*, 2012.

[38] Yuge Shi, Imant Daunhawer, Julia E Vogt, Philip HS Torr, and Amartya Sanyal. How robust are pre-trained models to distribution shift? *arXiv preprint arXiv:2206.08871*, 2022.

[39] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.

[40] Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2020.

[41] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 1984.

[42] Santosh S Vempala. *The random projection method*, volume 65. American Mathematical Soc., 2005.

[43] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. 2019.

[44] Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*, 2020.

[45] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*, 2022.

A Notations

Definition 5 (Compatibility)**.** *Let $\mathcal{X}$ be an instance space, $\mathcal{H}$ be a hypothesis class, and D_X be the marginal data distribution over $\mathcal{X}$. A compatibility score of a classifier at a point $x \in \mathcal{X}$ is defined as $\chi : \mathcal{C} \times \mathcal{X} \to [0, 1]$. Then, the compatibility between a classifier and a data distribution is defined as*

$$\chi(h, D_X) = \mathbb{E}_{x \sim D_X}[\chi(h, x)]$$

We estimate the empirical compatibility score of a classifier with a sample $S \sim D_X^n$ of size n with

$$\hat{\chi}(h, S) = \frac{1}{n} \sum_{x \in S} \chi(h, x).$$

B Proofs for disjunctions

Recall the definition for the component graph. For a distribution D, the component graph is defined as $G_D = (V, E_D)$ where each node corresponds to one of the d variables i.e. $V = \{1, \ldots, d\}$ and E_D contains an edge $(x[i], x[j])$ if and only if $x[i] = x[j] = 1$ for some x in the support of the distribution D. Similarly, let $\hat{G}_S = (V, \hat{E}(S))$ be the empirical component graph where $\hat{E}$ contains an edge $(x[i], x[j])$ if and only if $x[i] = x[j] = 1$ for some example $x \in S$. We denote the probability $\mathbb{P}_{x \sim D}[x[i] = x[j] = 1]$ as $p_{i,j}^D$. Note that the empirical component graph $\hat{G}_S$ is a random object where the edge (i, j) exists with probability $1 - (1 - p_{i,j}^D)^{|S|}$. For a family of distributions $\mathcal{D}$, the minimum positive edge probability $p_{min}^{\mathcal{D}}$ is defined as $\min_{D \in \mathcal{D}, i \neq j, p_{i,j}^D > 0} p_{i,j}^D$.

B.1 Proof of Theorem 1

Theorem 1. *Let $\mathcal{D}$ be the set of compatible distributions (defined in Definition 2) with respect to the hypothesis class DISJ_d^k and the compatibility function χ^{DISJ}. For $\alpha, \beta \in (0, 1)$, $\epsilon, \delta > 0$, DISJ_d^k can be $(\alpha, \beta, \epsilon, \delta)$-private semi-supervised PAC learned on the family of distributions $\mathcal{D}$ with*

$$n_{unl} = O\left(\frac{\log \frac{2d^2}{\beta}}{-\log\left(1 - p_{\min}^{\mathcal{D}}\right)}\right), n_{lab} = O\left(\frac{1}{\alpha\epsilon}\left(Z_{\max} + polylog(\frac{1}{\beta}, \frac{1}{\delta})\right)\right).$$

Here, $Z_{\max}$ is the maximum number of components in the component graph for distributions in $\mathcal{D}$ and $p_{\min}^{\mathcal{D}}$ is as defined above.

Proof. Consider the algorithm $\mathcal{A}$ that outputs a hypothesis $h \in \mathrm{DISJ}_d^k$ given as input a labelled and unlabelled dataset S_{lab} and S_{unl} of size n_{lab} and n_{unl} respectively.

Step 1 Generate the empirical component graph $\hat{G}_{S_{\text{unl}}} = (V, \hat{E}_{S_{\text{unl}}})$ with S_{unl}, where $V = \{1, \ldots, d\}$ and $\hat{E}_{S_{\text{unl}}}$ incldues an edge (i, j) if and only if $x[i] = x[j] = 1$ for some $x \in S_{\text{unl}}$. Denote the number of components in $\hat{G}_{S_{\text{unl}}}$ as $\hat{Z}$. Then, we define the reduced hypothesis class as $\widetilde{\mathrm{DISJ}}_d^k = \cup_{k=1}^{\hat{Z}} \mathrm{DISJ}_{\hat{Z}}^k$.

Step 2 Let the score function over a function f and a labelled dataset S_{lab} be defined as $Q(f, S_{\text{lab}}) = -\sum_{x \in S_{\text{lab}}} \mathbb{1}\{f(x) \neq y\}$. Output a hypothesis $f \in \widetilde{\mathrm{DISJ}}_d^k$ with probability proportional to $\exp\left(\frac{\epsilon q(f, S_{\text{lab}})}{2}\right)$

First, we show that $\mathcal{A}$ is $(\epsilon, 0)$-DP on the labelled dataset S_{lab}. The graph generation step (Step 1) only uses the public unlabelled data and has no effect on privacy of the labelled dataset. Then, the privacy guarantee of exponential mechanism ([33]) ensures the Step 2 of $\mathcal{A}$ is $(\epsilon, 0)$-private on the labelled dataset.

We then show that $\mathcal{A}$ is an accurate algorithm. In particular, for any $D \in \mathcal{D}$ with marginal distribution D_X, for any $\alpha, \beta \in (0, 1)$, given the unalbelled and labelled datasets S_{unl} and S_{lab} from D_X and D

respectively, the output distribution of $\mathcal{A}$ satisfies $\mathbb{P}_{h\sim\mathcal{A}(S_{\text{unl}},S_{\text{lab}})}\left[\mathbb{P}_{(x,y)}[h(x)\neq y]\geq\alpha\right]\leq\beta$ if the size of S_{lab} and S_{unl} are at least the sample complexity in the theorem.

We first show that the empirical component graph $\hat{G}_{S_{\text{unl}}}$ contains more than $Z_{\max}$ components with probability less than $\frac{\beta}{2}$. Let $\xi_{i,j}$ be the event that the edge (i,j) is in the component graph but not in the the empirical component graph. For any $i\neq j\in V$ such that $\mathbb{P}_{x\sim D}\left[x[i]=x[j]\right]>0$, the probability of $\xi_{i,j}$ is upper bounded by $\frac{\beta}{d^2}$, *i.e.*

$$\mathbb{P}_{S_{\text{unl}}\sim D_X^{n_{\text{unl}}}}\left[\xi_{i,j}\right]=(1-p_{ij}^{D_X})^{n_{\text{unl}}}\leq(1-p_{\min}^{D_X})^{n_{\text{unl}}}\leq\frac{\beta}{2d^2}\tag{4}$$

where the last inequality follows by the sample complexity of the unlabelled dataset $n_{\text{unl}}=O\left(\frac{\log\frac{2d^2}{\beta}}{-\log\left(1-p_{\min}^{D_X}\right)}\right)$.

Applying the union bound over $\xi_{i,j}$ for all pairs of $(i,j)\in\{1,\ldots,d\}\times\{1,\ldots,d\}$, we can show that the empirical component graph contains all edges in the component graph with probability at least $1-\beta/2$. This implies that the empirical component graph, a subgraph of the component graph by construction, is exactly the same as the component graph. Thus, the number of components in $\hat{G}_{S_{\text{unl}}}$ is upper bounded by $Z_{\max}$ with probability at least $1-\beta/2$.

Note that for any distribution $D\in\mathcal{D}$, there exists a compatible labelling function f, i.e. $\chi^{\text{DISJ}}(f,D)=1$ and $\mathbb{P}_{(x,y)\sim D}\left[f(x)\neq y\right]=0$. This ensures that each component in the empirical component graph $\hat{G}_{S_{\text{unl}}}$ consists of variables of the same type. Thus, f remains in the reduced hypothesis class $\widetilde{\text{DISJ}}_d^k$ of size $2^{Z_{\max}}$. The second step of $\mathcal{A}$ is equivalent to the generic private learner in [29] for the reduced hypothesis class $\widetilde{\text{DISJ}}_d^k$. Thus, the labelled sample complexity is

$$n_{\text{lab}}=O\left(\frac{1}{\alpha\epsilon}\left(Z_{\max}+\mathit{polylog}(\frac{1}{\beta},\frac{1}{\delta})\right)\right).$$

□

B.2 Proof of Theorem 2

Theorem 2. *For $p\in(0,1)$, $k,d>35$, $\alpha\in(0,1),\beta\in\left(4\exp\left(-\frac{d-5}{9}\right),\frac{4}{d}\right)$, $\epsilon,\delta>0$, DISJ_d^k is $(\alpha,\beta,\epsilon,\delta)$-private semi-supervised PAC learnable with compatible distributions $\mathcal{D}_p$ with*

$$n_{unl}\geq\max\left\{\frac{\log\left(1-\frac{9\log\frac{16}{\beta}+4}{d-1}\right)}{\log(1-p)},8\log\frac{16}{\beta}\right\},n_{lab}\geq O\left(\frac{1}{\alpha\epsilon}\left(1+\mathit{polylog}(\frac{1}{\beta},\frac{1}{\delta})\right)\right).$$

Before proving theorem 2, we state the following two lemmas.

Lemma 1 ([28])**.** *For a graph $G=(V,E)$ with $V=\{1,\ldots,d\}$. Let $G(p)=(V,\tilde{E})$ be a random subgraph of G with the same vertex set and an edge set $\tilde{E}\subset E$ where each edge in E is included in $\tilde{E}$ with probability p. Let $\hat{c}$ be the minimum expected value of any cut in $G(p)$. For $\ell\geq1$, let $\epsilon=\sqrt{3(\ell+2)\log d/\hat{c}}$. If $\epsilon\leq1$, then with probability at least $1-\frac{4}{\ell d^\ell}$, every cut in $G(p)$ has value between $1-\epsilon$ and $1+\epsilon$ times its expected value.*

Lemma 2. *For $d>35$, $\beta\in\left(4\exp\left(-\frac{d-5}{9}\right),\frac{4}{d}\right)$, an Erdos-Renyi random graph $G(d,p)$ is connected with probability at least $1-\beta$ if*

$$p\geq\frac{9\left(\log\frac{4}{\beta}+4\right)}{d-1}$$

Proof. Let G_0 be a complete graph with d vertices, then an Erdos-Renyi random graph $G(d,p)$ can be viewed as a random subgraph $G(p)$ of G_0 with the same vertex set and an edge set that contains each edge in G_0 with probability p.

Note that the minimum cut of G_0 is $d-1$. A graph is connected if its minimum cut is greater than 1.

For $\beta \in \left(4\exp\left(4-\frac{d-1}{3}\right), \frac{4}{d}\right)$, let $\ell = \frac{\log\frac{4}{\beta}}{\log d} \geq 1$. Then,

$$1 - \frac{4}{\ell d^{\ell}} \geq 1 - \frac{4}{d^{\ell}} = 1 - \beta$$

By lemma 1, with probability at least $1 - \frac{4}{\ell d^{\ell}} \geq 1 - \beta$, every cut in the random graph satisfies

$$\begin{aligned}(1-\epsilon)\, pc &= \left(1 - \sqrt{\frac{3\,(\ell+2)\log d}{p(d-1)}}\right) p\,(d-1) \\ &= (d-1)\,p - \left(\sqrt{3p\,(\ell+2)\,(d-1)\log d}\right) \\ &\overset{(a)}{=} (d-1)\,p - \left(\sqrt{3\,(d-1)\left(\log\frac{4}{\beta} + 2\log d\right)}\right)\sqrt{p} \geq 1\end{aligned} \tag{5}$$

where (a) follows by setting $\ell = \log\frac{4}{\beta}/\log d$.

By solving the quadratic inequality in $\sqrt{p}$, we find that the last inequality in Equation (5) is satisfied for

$$\sqrt{p} \geq \sqrt{\frac{9\log\frac{4}{\beta}+4}{d-1}} \geq \frac{1}{2}\left(\sqrt{\frac{3\left(\log\frac{4}{\beta}+2\log d\right)}{d-1}} + \sqrt{\frac{3\left(\log\frac{4}{\beta}+2\log d\right)+4}{d-1}}\right) \tag{6}$$

as $\log\frac{4}{\beta} \geq \log d$. That is

$$p \geq \frac{9\log\frac{4}{\beta}+4}{d-1}$$

Thus, the random graph is connected with probability at least $1-\beta$. □

Proof for Theorem 2. Let $\mathcal{A}$ be the algorithm as defined in the proof for Theorem 1. For any distribution $D \in \mathcal{D}_p$, given the unlabelled and labelled datasets S_{unl} and S_{lab} of size n_{unl} and n_{lab} from D_X and D, denote the empirical component graph obtained in the first step of $\mathcal{A}$ by $\hat{G}\left(S_{\text{unl}}\right)$. Denote the component graph of the distribution by G_{D_X}.

Note that for distribution $D \in \mathcal{D}_p$, the component graph can be viewed as the combination of the two random graphs G^+ and G^- in the data generation process as defined in Definition 4, where G^+ is the component graph for positive examples, and G^- is the component graph for negative examples.

Similarly, we define $S^+_{\text{unl}} = \{x \in S_{\text{unl}} : f(x) = 1\}$ and $S^-_{\text{unl}} = \{x \in S_{\text{unl}} : f(x) = -1\}$ to be the positive unlablled dataset and negative unlabelled dataset of size n^+_{unl} and n^-_{unl} respectively. Then, the empirical component graph is the composition of the positive and negative empirical component graphs $\hat{G}^+\left(S^+_{\text{unl}}\right)$ and $\hat{G}^-(S^-_{\text{unl}})$.

Then, we show that the positive and negative empirical componnent graphs $\hat{G}^+\left(S^+_{\text{unl}}\right)$ and $\hat{G}^-(S^-_{\text{unl}})$ are connected with probability at least $1-\frac{\beta}{2}$. By Lemma 2, the positive and negative empirical componnent graphs $\hat{G}^+\left(S^+_{\text{unl}}\right)$ and $\hat{G}^-(S^-_{\text{unl}})$ are connected with probability at least $1-\frac{\beta}{2}$ if

$$\tilde{p}^+ \geq \frac{9\left(\log\frac{16}{\beta}+4\right)}{k-1}, \tilde{p}^- \geq \frac{9\left(\log\frac{16}{\beta}+4\right)}{d-k-1} \tag{7}$$

where $\tilde{p}^+ = 1-(1-p)^{n^+_{\text{unl}}}$ and $\tilde{p}^- = 1-(1-p)^{n^-_{\text{unl}}}$.

As the label $\mathbb{P}(y)$ is a Bernoulli random variable with probability $\frac{1}{2}$, by Hoeffding's inequality, with probability at least $1-\frac{\beta}{4}$,

$$\frac{1}{4}n_{\text{unl}} \leq n^+_{\text{unl}}, n^-_{\text{unl}} \leq \frac{3}{4}n_{\text{unl}} \tag{8}$$

for $n_{\text{unl}} \geq 8 \log \frac{16}{\beta}$. Condition on (8), the equations in (7) are satisfied as

$$n_{\text{unl}} \geq \max \left\{ \frac{\log\left(1 - \frac{9 \log \frac{16}{\beta} + 4}{k-1}\right)}{\log(1-p)}, \frac{\log\left(1 - \frac{9 \log \frac{16}{\beta} + 4}{d-k-1}\right)}{\log(1-p)} \right\} \geq \frac{\log\left(1 - \frac{9 \log \frac{16}{\beta} + 4}{d-1}\right)}{\log(1-p)}$$

Thus, the empirical component graph that combines $\hat{G}^{+}(S_{\text{unl}})$ and $\hat{G}^{-}(S_{\text{unl}})$ has exactly two components, which implies that the reduced hypothesis class includes two hypotheses. By [29], the labelled sample complexity for learning the reduced hypothesis class is upper bounded by

$$n_{\text{lab}} = O\left(\frac{1}{\alpha\epsilon}\left(1 + polylog(\frac{1}{\beta}, \frac{1}{\delta})\right)\right).$$

□

C Proofs for linear halfspaces

Theorem 3. *Let $\mathcal{D}_d$ be the family of distributions such that any $D \in \mathcal{D}_d$ over $\mathcal{X} \times \mathcal{Y}$ satisfies*

- *There exists $w^* \in B_2^d$ such that $f_{w^*} = sign(\langle w^*, x \rangle)$ is compatible with D and accurate,* i.e.$\chi_\gamma(w^*, D) = 1$ *and* $\mathbb{P}_{(x,y)\sim D}[f_{w^*}(x) \neq y] = 0$.
- *Let $\Sigma_X = \mathbb{E}_{x \sim D_X}\left[xx^\top\right]$ be the covariance matrix and $\lambda_1(\Sigma_X), \ldots, \lambda_d(\Sigma_X)$ its eigenvalues in descending order. Then, there exists a $k \ll d$-dimensional approximation of Σ_X,* i.e.$\sum_{i=k+1}^{d} \lambda_i(\Sigma_X) \leq \eta$.

Then, the hypothesis class $\mathcal{H}_L^d$ of linear halfspaces of dimension d is $(\alpha, \beta, \epsilon, \delta)$-private semi-supervised learnable on $\mathcal{D}_d$ with sample complexity

$$n_{unl} = O\left(\frac{kd}{\beta\gamma^2}\right), \quad n_{lab} = O\left(\frac{\sqrt{k}}{\alpha\epsilon\left(\gamma - \sqrt{\frac{\eta}{\beta}}\right)}\right) \tag{3}$$

Proof. Let $\mathcal{A}$ be an algorithm that outputs a hypothesis $h \in \mathcal{H}_L^d$ given as input the privacy parameters ϵ, δ and the unlabelled and labelled datasets S_{unl} and S_{lab} of size n_{unl} and n_{lab}. Define $\mathcal{A}$ as follows.

Step 1 Calculate the empirical covariance matrix of the unlabelled dataset $\hat{\Sigma}_{S_{\text{unl}}} := \frac{1}{n}\sum_{x \in S}(x - \bar{x})(x - \bar{x})^T$, where $\bar{x} = \frac{1}{n}\sum_{x \in S} x$. Let $\hat{A}_{S_{\text{unl}}}$ be the matrix consisting of eigenvectors of $\hat{\Sigma}_S$ corresponding to the top k eigenvalues of $\lambda_1(\hat{\Sigma}_{S_{\text{unl}}}), ..., \lambda_k(\hat{\Sigma}_{S_{\text{unl}}})$.

Step 2 Use (ϵ, δ)-private-SGD ([7]) with ramp loss on the low-dimensional mapping of the labelled dataset $S_{\text{lab}}^k = \{(\hat{A}_{S_{\text{unl}}}^T x, y) : (x, y) \in S_{\text{lab}}\}$ to obtain a hypothesis $\hat{w} \in B_2^k$ and output $h(x) = sign\left((\hat{A}_{S_{\text{unl}}}\hat{w})^T x\right)$, where $\hat{A}_{S_{\text{unl}}}\hat{w} \in B_2^d$.

First, the algorithm $\mathcal{A}$ is (ϵ, δ)-DP on the labelled dataset S_{lab} because only the second step operates on the labelled dataset and preserves (ϵ, δ)-DP by the privacy guarantee of private-SGD ([7]).

Then, we show that the algorithm $\mathcal{A}$ is accurate. In particular, for any $\alpha, \beta > 0$, for any $D \in \mathcal{D}_d$ with marginal distribution D_X, given an unlabelled and labelled dataset S_{unl} and S_{lab} of size n_{unl} and n_{lab} from D_X and D respectively, the output of the algorithm $\mathcal{A}$ satisfies $\mathbb{P}_{h \sim \mathcal{A}(S_{\text{unl}}, S_{\text{lab}})}\left[\mathbb{P}_{(x,y)}[h(x) \neq y] \leq \alpha\right] \geq 1 - \beta$.

For any $D \in \mathcal{D}_d$, let $f_{w^*} = sign((w^*)^T x)$ be the function in $\mathcal{H}_L^d$ such that $\chi_\gamma(f_{w^*}, D) = 1$ and $\mathbb{P}_{(x,y)\sim D}[f_{w^*}(x) \neq y] = 0$. First, we show that Step 1 of the algorithm $\mathcal{A}$ preserves a margin of $\gamma - O\left(\sqrt{\frac{\eta}{\beta}}\right)$ in the low-dimensional data space with probability at least $1 - \frac{\beta}{2}$.

For a matrix $A \in \mathbb{R}^{d\times k}$ and a dataset $S = (x_1, ..., x_n)$ sampled from D_X, define the distributional and empirical reconstruction error as $R(A) = \mathbb{E}_{x\in D_X}\left[||x - AA^Tx||^2\right]$ and $\hat{R}(A) = \frac{1}{n}\sum_{i=1}^{n}||x_i - AA^T||^2$.

Define the following two bad events ξ_1 and ξ_2,

$$\xi_1 = \left\{ \left(\langle w^*, x\rangle - \langle \hat{A}_{S_{\text{unl}}}^T w^*, \hat{A}_{S_{\text{unl}}}^T x\rangle\right)^2 \geq \frac{4R(\hat{A}_{S_{\text{unl}}})}{\beta} \right\} \tag{9}$$

$$\xi_2 = \left\{ \left| R(\hat{A}_{S_{\text{unl}}}) - \hat{R}(\hat{A}_{S_{\text{unl}}}) \right| \geq 4\sqrt{\frac{kd}{\beta n}} \right\} \tag{10}$$

Note that the probability of the bad event ξ_1 is upper bounded by $\frac{\beta}{4}$, which follows by Markov inequality as we can upper bound the expectation of $\left(\langle w^*, x\rangle - \langle \hat{A}_{S_{\text{unl}}}^T w^*, \hat{A}_{S_{\text{unl}}}^T x\rangle\right)^2$ by the distributional reconstruction error $R(\hat{A}_{S_{\text{unl}}})$.

$$\begin{aligned}
\mathbb{E}\left[|\langle w^*, x\rangle - \langle \hat{A}_{S_{\text{unl}}}^T w^*, \hat{A}_{S_{\text{unl}}}^T x\rangle|^2\right] &= \mathbb{E}\left[|\langle w^*, x - \hat{A}_{S_{\text{unl}}}\hat{A}_{S_{\text{unl}}}^T x\rangle|^2\right] \\
&\leq \mathbb{E}\left[||w^*||_2^2||x - \hat{A}_{S_{\text{unl}}}\hat{A}_{S_{\text{unl}}}^T x||_2^2\right] \\
&\leq \mathbb{E}\left[||x - \hat{A}_{S_{\text{unl}}}\hat{A}_{S_{\text{unl}}}^T x||_2^2\right] = R(\hat{A}_{S_{\text{unl}}})
\end{aligned}$$

Then, we will show that the probability of the bad event ξ_2 is smaller than $\frac{\beta}{4}$. Consider $\mathscr{A}$ as the set of all matrices whose columns consist of the k eigenvectors of some positive semidefinite matrix corresponding to its top k eigenvalues. Thus, any matrix $A \in \mathscr{A}$ satisfies $\gamma_1(AA^T) = 1$. Also, note that $|\mathscr{A}| \leq dk$.

We upper bound the variance of the empirical reconstruction error of any matrix $A \in \mathscr{A}$ by 2.

$$\begin{aligned}
Var(\hat{R}(A)) &= \mathbb{E}\left[||x - AA^Tx||^4\right] - \mathbb{E}\left[||x - AA^Tx||\right]^2 \\
&\leq \mathbb{E}\left[||x - AA^Tx||^4\right] \\
&= \mathbb{E}\left[\left(x^Tx - x^TAA^Tx\right)^T\left(x^Tx - x^TAA^Tx\right)\right] \\
&\leq \mathbb{E}\left[(x^Tx)^2 + (xAA^Tx)^2\right] \\
&\overset{(a)}{\leq} \mathbb{E}\left[(x^Tx)^2 + (x^Tx)^2\right] \leq 2
\end{aligned} \tag{11}$$

where (a) is due to $\gamma_1(AA^T) = 1$ and $x^TAA^Tx \leq x^T\gamma_1(AA^T)x$.

Thus, applying union bound over all $A \in \mathscr{A}$ and Chebyshev's inequality with $\mathbb{E}\left[\hat{R}(A)\right] = R(A)$ and $Var(\hat{R}(A)) \leq 2$ gives

$$\mathbb{P}\left[\text{sup}_{A\in\mathscr{A}}|R(A) - \hat{R}(A)| \geq \frac{2\ell}{\sqrt{n}}\right] \leq \sum_{A\in\mathscr{A}}\mathbb{P}\left(|R(A) - \hat{R}(A)| \geq \frac{2\ell}{\sqrt{n}}\right) \leq \frac{kd}{\ell^2}$$

Choosing $\ell = 2\sqrt{\frac{kd}{\beta}}$ implies that the probability of ξ_2 is upper bounded by $\frac{\beta}{4}$.

As the bad events ξ_1 and ξ_2 occur with probaiblity less than $\frac{\beta}{4}$, the union bound implies that the probability that none of them occurs is at least $1 - \frac{\beta}{2}$. Then, we show that if none of the bad events occurs, the low-dimensional data space transformed by $\hat{A}_{S_{\text{unl}}}$ preserves a margin of $\gamma - O\left(\sqrt{\frac{\eta}{\beta}}\right)$.

If the event ξ_1 does not occur, we have the following inequalities

$$\langle w^*, x\rangle - 2\sqrt{\frac{R\left(\hat{A}_{S_{\text{unl}}}\right)}{\beta}} \leq \langle \hat{A}_{S_{\text{unl}}}^T w^*, \hat{A}_{S_{\text{unl}}}^T x\rangle \leq \langle w^*, x\rangle + 2\sqrt{\frac{R\left(\hat{A}_{S_{\text{unl}}}\right)}{\beta}} \tag{12}$$

If the event ξ_2 does not occur, we can derive an upper bound on the distributional reconstruction error of $\hat{A}_{S_{\text{unl}}}$ as follows.

$$\begin{aligned} R(\hat{A}_{S_{\text{unl}}}) &\leq 2\sqrt{\frac{dk}{\beta n}} + \hat{R}(\hat{A}_{S_{\text{unl}}}) \\ &\leq 2\sqrt{\frac{dk}{\beta n}} + \sum_{j=k+1}^{d} \hat{\lambda}_j \\ &\leq 2\sqrt{\frac{dk}{\beta n}} + \sum_{j=k+1}^{d} \lambda_j + \sum_{j=k+1}^{d} (\hat{\lambda_j} - \lambda_j) \\ &\leq O\left(\sqrt{\frac{dk}{\beta n}}\right) + \eta \end{aligned} \tag{13}$$

where the last inequality follows by the assumption on the approximate low-dimensional data space and $\sum_{j=k+1}^{d} ||\lambda_j - \hat{\lambda}_j||^2 = O\left(\frac{1}{\sqrt{n}}\right)$ by [43].

Substitute Equation (13) into the left side of Equation (12), we have for $y = 1$,

$$y\langle \hat{A}_{S_{\text{unl}}}^T w^*, \hat{A}_{S_{\text{unl}}}^T x\rangle \geq y\langle w^*, x\rangle - \sqrt{\frac{c_1}{\beta}\sqrt{\frac{dk}{\beta n}} + \eta} \geq \gamma - O\left(\sqrt{\frac{\eta}{\beta}}\right) \tag{14}$$

where c_1 is a positive constant and the last inequality is by $n \geq O\left(\frac{dk}{\beta\eta^2}\right)$.

Similarly, by Equation (13) and the right side of Equation (13), we have for $y = -1$,

$$y\langle \hat{A}_{S_{\text{unl}}}^T w^*, \hat{A}_{S_{\text{unl}}}^T x\rangle \geq y\langle w^*, x\rangle - \sqrt{\frac{c_2}{\beta}\sqrt{\frac{dk}{\beta n}} + \eta} \geq \gamma - O\left(\sqrt{\frac{\eta}{\beta}}\right) \tag{15}$$

where $c_2 > 0$ is a constant.

This implies that with probability at least $1 - \frac{\beta}{2}$, the low-dimensional data space transformed by $\hat{A}_{S_{\text{unl}}}$ is linearly separable by the target function f_{w^*} with a margin of $O\left(\gamma - \sqrt{\frac{\eta}{\beta}}\right)$. Applying the sample complexity result of private-SGD by [7] in the low-dimensional space, we get the labelled sample complexity

$$n_{\text{lab}} = O\left(\frac{\sqrt{k}}{\alpha\epsilon\left(\gamma - \sqrt{\frac{\eta}{\beta}}\right)}\right)$$

□

D Experimental details

For our linear-probing experiments, we use off the shelf ResNet50 models trained on ImageNet-100 using Mocov3 and vanilla SL training on ImageNet.

We obtain our models from `https://github.com/vturrisi/solo-learn` and `https://pytorch.org/hub/` respectively. To prevent the resolution discrepancy between CIFAR and ImageNet from negatively impacting the performance, we apply standard ImageNet preprocessing (rescaling to 256, cropping to 224x224, and normalising) to CIFAR images. For DP training, we search two main hyper-parameters— number of steps in $[500, 1000, 2000]$ and the learning rate in $[0.01, 0.1]$, and we fix the clipping value to 1. For DP linear probing, we use a batch size of 4096 as it is the largest power-of-two batch size we could meaningfully use for the experiments in the low-data regime. We do not use any form of data augmentation. For DP full body fine-tuning, we use a batch size of 1024, as it was the largest batch size we could fit in memory. We also apply Augmentation

Multiplicity with multiplicity 8 and and Exponential Moving Average partially following De et al. [19], but without using Weight Normalization, as it would not be meaningful for a model that has not been trained with it from scratch. For non-DP linear probing we select the number of steps by searching in $[500, 1000, 2000, 3000]$ iterations and the learning rate in $[0.01, 0.1, 1]$, and use a batch size of 256. All results are averaged over 5 seeds.

To obtain the margins in Figure 3, we first perform PCA on the 2048-dimensional embeddings from the SSL and SL pre-trained feature extractor respectively. Then, for each class, we train a linear SVM in a one-versus-rest manner. A margin of a data point is the distance of the data point to a separating hyperplane. For the embeddings on CIFAR100 from both SL and SSL feature extractors, we report the margin value as the minimum margin over all classes satisfied by at least 99% of the data. For the embeddings on CIFAR10, we report the margin value as the minimum margin over all classes satisfied by at least 99% of the data for the SL feature extractor and 96% of the data for the SSL feature extractor. Normally, a margin is computed to be the minimum distance to the halfspace satisfied by all data points. However, here, the representations are not linearly separable and hence we use this relaxation.