Private Multi-Winner Voting for Machine Learning

Adam Dziedzic
adam.dziedzic@utoronto.ca
University of Toronto and Vector Institute
Canada

Christopher A. Choquette-Choo[*]
choquette.christopher@gmail.com
Google Research, Brain Team
USA

Natalie Dullerud
natalie.dullerud@mail.utoronto.ca
University of Toronto and Vector Institute
Canada

Vinith Suriyakumar[*]
suriyaku@gmail.com
MIT
USA

Ali Shahin Shamsabadi[*]
a.shahinshamsabadi@turing.ac.uk
The Alan Turing Institute
UK

Muhammad Ahmad Kaleem
ahmad.kaleem@mail.utoronto.ca
University of Toronto and Vector Institute
Canada

Somesh Jha
jha@cs.wisc.edu
University of Wisconsin-Madison
USA

Nicolas Papernot
nicolas.papernot@utoronto.ca
University of Toronto and Vector Institute
Canada

Xiao Wang
wangxiao@cs.northwestern.edu
Northwestern University
USA

ABSTRACT

Private multi-winner voting is the task of revealing k-hot binary vectors satisfying a bounded differential privacy (DP) guarantee. This task has been understudied in machine learning literature despite its prevalence in many domains such as healthcare. We propose three new DP multi-winner mechanisms: Binary, τ, and Powerset voting. Binary voting operates independently per label through composition. τ voting bounds votes optimally in their ℓ_2 norm for tight data-independent guarantees. Powerset voting operates over the entire binary vector by viewing the possible outcomes as a power set. Our theoretical and empirical analysis shows that Binary voting can be a competitive mechanism on many tasks unless there are strong correlations between labels, in which case Powerset voting outperforms it. We use our mechanisms to enable privacy-preserving multi-label learning in the *central setting* by extending the canonical single-label technique: PATE. We find that our techniques outperform current state-of-the-art approaches on large, real-world healthcare data and standard multi-label benchmarks. We further enable multi-label confidential and private collaborative (CaPC) learning and show that model performance can be significantly improved in the multi-site setting.

KEYWORDS

privacy, machine learning, multi-label classification, differential privacy

1 INTRODUCTION

Differential privacy techniques for machine learning have predominantly focused on two techniques: differentially private stochastic gradient descent (DPSGD) [1] and private aggregation of teacher ensembles (PATE) [35]. At the core of these techniques are two mechanisms. The Gaussian mechanism enables arbitrary private queries of data (e.g., gradients in DPSGD [1]). Instead, the noisy arg max can only reveal the max count. This is used to reveal the predicted label in *Private Aggregation of Teacher Ensembles* (PATE) [35]. These two mechanisms appeal well to the canonical *single-label* classification per input (a.k.a. multi-class classification).

In contrast, more real-world tasks, such as *multi-label* classification [44], can be modeled using multi-winner elections [21, 23]. These are settings where more than one candidate can win, i.e., each input can have > 1 class present. Outside of elections, other multi-winner election scenarios include canonical computer visions tasks like object recognition, where models must recognize all objects present in an image [7]. Another scenario is the task of inferring which topics were written about in a corpus of text (document): e.g., a news article may discuss politics, finance, and/or education. The principal setting we consider is healthcare, in which patient data (*e.g.,* symptom reports or X-rays) may be indicative of multiple conditions [28, 29].

We thus focus on creating private mechanisms for releasing the outcome of a multi-winner election. We first formalize the multi-winner election. Then, we propose Binary voting, a simple yet powerful solution that answers the k single-winner (for each class) elections independently. We prove that it is optimal when there is a lack of correlation among the outcomes of particular candidates and show empirically that this mechanism can outperform state-of-the-art baselines based on DP-SGD. Recognizing that correlations can often exist in data, we then propose τ and Powerset voting for obtaining tighter guarantees. τ voting obtains a tight data-independent privacy bound by limiting the number of votes from each voter, obtained via an ℓ_2 bound on their ballot. Powerset

[*]The work was done while the author was at the University of Toronto and Vector Institute.

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license visit https://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.
Proceedings on Privacy Enhancing Technologies 2023(1), 527–555
© 2023 Copyright held by the owner/author(s).
https://doi.org/10.56553/popets-2023-0031

voting is created by casting the multi-winner election to an analogous single-winner election—thus, Powerset voting reveals the result for all k candidates jointly.

By replacing the noisy arg max mechanism in (single-label) PATE, we create multi-label PATE for DP multi-label semi-supervised machine learning. We conduct extensive empirical evaluation on large datasets, including Pascal VOC, a common multi-label benchmark, as well as CheXpert [28], MIMIC [29], and PadChest [9] which are healthcare datasets. Despite different public data assumptions, we compare against DPSGD (including an improved adaptive variant [48]) because it is the only private baseline in multi-label classification thus far. We find that with the modest assumption of public unlabeled data (required for PATE), we achieve new state-of-the-art DP multi-label models with 85% AUC (17% better than DPSGD) on Pascal VOC. Because many multi-label settings may benefit from distributed, multi-site learning (e.g., healthcare), we further integrate our methods with the *CaPC* protocol for *Confidential and Private Collaborative* learning [10].

We find that training with our mechanisms can improve model performance significantly on large, real-world healthcare data with sensitive attributes. Our main contributions are as follows:

(1) We create three new DP aggregation mechanisms for private multi-winner voting: Binary voting, τ voting, and Powerset voting. We show theoretically and empirically that Binary voting performs better unless there is high correlation among labels.
(2) We enable private multi-label semi-supervised learning that achieves SOTA performance on large real-world tasks: Pascal VOC and 3 healthcare datasets.
(3) We enable multi-label collaborative learning in the multi-site scenario and show empirically that this significantly improves model performance.

2 BACKGROUND AND RELATED WORK

2.1 DP for Machine Learning

Differential Privacy (DP) is the canonical framework for measuring the privacy leakage of a randomized algorithm [19]. It requires the mechanism (in our work, the training algorithm), to produce statistically indistinguishable outputs on any pair of *adjacent* datasets: those differing by any but only one data point. This bounds the probability of an adversary inferring properties of the training data from the mechanism's outputs.

Definition 1 (Differential Privacy). *A randomized mechanism $\mathcal{M}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if for any subset $\mathcal{S} \subseteq \mathcal{R}$ and any adjacent datasets $X, X' \in \mathcal{D}$, i.e. $\|X - X'\|_1 \leq 1$, the following inequality holds:* $\Pr[\mathcal{M}(X) \in \mathcal{S}] \leq e^{\varepsilon} \Pr[\mathcal{M}(X') \in \mathcal{S}] + \delta$.

We use Rényi Differential Privacy (RDP) [33]. Because RDP bounds the privacy loss with the Rényi-divergence, it enables tighter accounting for our mechanisms that use Gaussian noise. RDP is a generalization of $(\varepsilon, \delta = 0) - DP$. We use standard conversions to report $(\varepsilon, \delta) - DP$.

Definition 2 (Rényi Differential Privacy). *A randomized mechanism $\mathcal{M}$ is said to satisfy ε-Rényi differential privacy of order λ, or (λ, ε)-RDP for short, if for any adjacent datasets $X, X' \in \mathcal{D}$:*

$$D_{\lambda}(\mathcal{M}(X) \,||\, \mathcal{M}(X')) =$$
$$= \frac{1}{\lambda - 1} \log \mathbb{E}_{\theta \sim \mathcal{M}(X)} \left[\left(\frac{\Pr[\mathcal{M}(X) = \theta]}{\Pr[\mathcal{M}(X') = \theta]} \right)^{\lambda - 1} \right] \leq \varepsilon,$$

where $D_{\lambda}(P \,||\, Q)$ is the Rényi Divergence between distributions P and Q defined over a range $\mathcal{R}$. It is convenient to consider RDP in its functional form as $\varepsilon_{\mathcal{M}}(\lambda)$, which is the RDP ε of mechanism $\mathcal{M}$ at order λ. Our privacy analysis builds on the following result.

Lemma 2.1. ***RDP-Gaussian mechanism (for a single-label setting)*** *[33]. Let $f : \mathcal{X} \to \mathcal{R}$ have bounded ℓ_2 sensitivity for any two neighboring datasets X, X', i.e., $\|f(X) - f(X')\|_2 \leq \Delta_2$. The Gaussian mechanism $\mathcal{M}(X) = f(X) + \mathcal{N}(0, \sigma^2)$ obeys RDP with $\varepsilon_{\mathcal{M}}(\lambda) = \frac{\lambda \Delta_2^2}{2\sigma^2}$.*

Another notable advantage of RDP over (ε, δ)-DP is that it composes naturally. This will be helpful when analyzing our approach which repeatedly applies the same private mechanism to a dataset.

Lemma 2.2. ***RDP Adaptive Sequential Composition*** *[33]. Given a sequence of k adaptively chosen mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$, each saatisfying (λ, ε_i)-RDP, then their union satisfies $(\lambda, \sum_{i=1}^{k} \varepsilon_i)$-RDP.*

2.2 PATE and CaPC

PATE (Private Aggregation of Teacher Ensembles) [35] is a single-label semi-supervised DP learning approach. First, an ensemble of models is trained on (disjoint) partitions of the training data. Each (teacher) model is then asked to predict (label) a test input by voting for one class. To provide the DP guarantee, only the noisy arg max of the teacher votes are released—instead of each vote directly. To do so, teacher votes are organized into a histogram where $n_i(x)$ indicates the number of teachers that voted for class i. Then, $\text{argmax}\{n_i(x) + \mathcal{N}(0, \sigma_G^2)\}$ is released, where the Gaussian variance σ_G^2 controls the prediction's privacy loss. Finally, these DP-labeled test inputs are used to train a student model which can be used by the model owner. Note that unlike DPSGD, this requires the additional but modest assumption of unlabeled data. Here, each noisy label incurs additional privacy loss (rather than gradients in DPSGD). Loose data-independent guarantees are obtained through advanced composition [20]. Tighter data-dependent guarantees, which can be safely released with additional analysis, are possible when many teachers agree on the predicted label [36]. To further reduce privacy loss, *Confident GNMax* only reveals predictions when $\max_i\{n_i(x)\} + \mathcal{N}(0, \sigma_T^2) > T$, *i.e.,* there is high consensus among teachers.

CaPC (Confidential and Private Collaboration) is a distributed collaborative learning framework that extends PATE from the central to the collaborative setting [10]. In addition to bounding the training data DP privacy loss, CaPC also protects the confidentiality of test data and teacher model parameters by introducing new cryptographic primitives. In CaPC, teacher models are known as *answering parties* and are distributed across sites, communicating via these cryptographic primitives—we use *answering parties* when referring specifically to CaPC and *teacher models* otherwise.

We now clarify the two privacy protection terms we use. *Confidentiality* refers to the notion that no other party (or adversary) can view, in plaintext, the data of interest—it protects the inputs to our mechanism. This differs from *DP* which reasons about what can be inferred from the outputs of our mechanism, in our case about the training set for each teacher model. Both protect data privacy in different ways. In CaPC, each answering party is a unique protocol participant with their own (teacher) model. The querying party initiates the protocol by (1) encrypting unlabeled data; other (answering) parties return an encrypted label. Then, (2) the encrypted vote is secretly shared with both the answering party and the Privacy Guardian (PG), such that each has only one share (neither knows the label in plaintext); the PG follows PATE to achieve an $(\varepsilon, \delta) - DP$ bound for all the answering parties. Finally, (3) the PG returns the DP vote to the querying party via secure computation. The goal is to maximize the number and quality of votes that can be returned to the querying party under a chosen (ε, δ)—DP guarantee.

2.3 Prior Work on Multi-label Classification

The closest to our work is that of [51], which provides a privacy-preserving nearest neighbors algorithm formulated using PATE. The neighboring training points are the teachers and their labels the votes. Our mechanisms can similarly be used to extend [51] to the multi-label setting. However, their work assumes the existence of a publicly-available embedding model (used to project the high-dimensional data to a low-dimensional manifold for meaningful nearest-neighbor distance computation). This is a strong assumption that is often not applicable to the multi-site scenario we consider and especially for sensitive tasks like healthcare. Our approach does not make this assumption. Further, we show in Section 3 that their ℓ_1 vote clipping scheme is not optimal for the commonly used Gaussian distribution. Instead, we prove tighter bounds by clipping votes to an ℓ_2 ball.

The multi-label voting problem can also be formulated as simultaneously answering multiple counting queries which has been studied extensively in the theoretical literature, e.g., [15]. These have yet to show promise empirically in settings such as multi-label voting. We explored these approaches in our setup (not shown) finding looser data-independent guarantees compared to our proposed mechanisms. Another alternative multi-label method is differentially private stochastic gradient descent (DPSGD), which is agnostic to the label-encoding because the mechanism releases gradient updates. Abadi et al. [2] proposed and evaluated this approach in the single-label setting. Zhang et al. [48] applied it to the multi-label setting to train multi-label models (for $k = 5$ labels). We compare with [48] and find that our methods perform better. Many works explore using public labeled datasets to reduce privacy costs, e.g., [32]—this is a stronger assumption than ours. Finally, we stress that our evaluation includes large-scale healthcare datasets with a natural need for privacy and the additional challenge of being severely imbalanced.

Private *heavy hitters* is a related problem to private multi-label classification, where a set of n users each has an element from a universe of size d. The goal is to find the elements that occur (approximately) most frequently, but do so under a local differential privacy constraint [4, 50]. However, these methods release the value of the heavy hitter rather than the argmax.

3 DIFFERENTIAL PRIVACY MECHANISMS FOR MULTI-WINNER ELECTIONS

Multi-winner election systems are direct generalizations of popularly used single-winner elections. They are adopted commonly in real-life elections and their comprehensive analysis can be found in [21] and [23]. Here, we formally define a multi-winner system and propose DP mechanisms to facilitate the private release of the outcome of a multi-winner election.

Compared to single-winner elections, multi-winner elections are more challenging for designing a DP mechanism: single-winner elections only output a single scalar per collection of ballots, indicating the sole winner, whereas multi-winner elections output a vector of winners and thus have higher sensitivity (use more ε) per query. DP mechanisms for single-winner voting have been well studied before, where the noisy arg max achieves tight privacy loss resulting from tight sensitivity and *information minimization* [20]. However, to the best of our knowledge, no prior work formally defines and analyzes DP mechanisms for multi-winner election systems; hence, we provide and analyze Definition 3 to then get formal privacy loss bounds in Section 4.

Definition 3. ***θ-multi-winner election.*** *Given a set of n voters, each with a ballot of k candidates (coordinates) $\boldsymbol{b} \in \{0,1\}^k$ such that $\|\boldsymbol{b}\|_2 \leq \theta$. The outcome of a θ-multi-winner election with threshold T is defined as*

$$f(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \triangleq \left\{ i : \sum_{j=1}^{n} \boldsymbol{b}_j[i] > T \right\},$$

where $T = \frac{n}{2}$ and with the binary decision per label i, $\sum_{j=1}^{n} \boldsymbol{b}_j[i]$ represents the number of positive votes. Note that this definition is deterministic since no noise is added and setting the threshold $T = \frac{n}{2}$ is equivalent to using the number of negative votes $(n - \sum_{j=1}^{n} \boldsymbol{b}_j[i])$ as T.

3.1 DP ∞-Multi-Winner Elections

Observe that in Definition 3, we only require the release of a binary vector. Intuitively, this makes other related private mechanisms that release the entire privatized histogram sub-optimal, *e.g.,* multiple counting queries and the Gaussian mechanism—they should require no less (and often much more) budget than mechanisms that only release the binary vector because of information minimization [20].

Thus, we instead focus on private release of the ∞-multi-winner-election[1]. Because the noisy arg max is a well-studied mechanism for single-winner elections, our first multi-winner approach is to decompose it into k separate single-winner elections. Then, we can obtain a tight final guarantee on the multi-winner ballot via RDP composition over the k independent applications (Lemma 2.2). We call this Binary voting and state it formally in Definition 4 below. We again use RDP composition to obtain a total privacy budget across separate queries (instances of the multi-winner elections).

Definition 4 (DP Binary Voting Mechanism). *For an ∞-multi-winner election, the binary voting mechanism based on noisy* arg max

[1] We use ∞ to emphasize that any vote $\boldsymbol{b} \in \{0,1\}^k$ is allowed but a tighter bound exists: $\|\boldsymbol{b}\|_2 \leq \sqrt{k} < \infty$.

is:

$$\mathcal{M}_\sigma(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n) \triangleq \left\{ i : \left(\sum_{j=1}^{n} \boldsymbol{b}_j[i] + \mathcal{N}(0,\sigma_G^2) \right) > T_i \right\},$$

where $T_i = n - \sum_{j=1}^{n} \boldsymbol{b}_j[i]$ is the number of negative votes. We use this per-candidate threshold T_i instead of a fixed $T = n/2$ because this reduces the privacy cost. However, to preserve privacy, we must add noise to both the positive and negative vote counts (each bin of the histogram). Though we show only the noise added for positive votes, this is equivalent to adding less noise for each label. Note that this definition follows directly from PATE by viewing the i-th label release as a binary task of the label being present or not.

The distinguishing feature of the above Binary mechanism is that it releases a binary result independently, per candidate. Thus, there are many possible Binary voting mechanisms satisfying this goal. We base ours on the noisy arg max mechanism because of its advantageous privacy guarantees stemming from *information minimization* [20]: changing the training set to any adjacent one can only modify a single ballot. Thus, this mechanism has advantageous properties both theoretically, and as well empirically as presented in [35, 36]. Next, we show that an independent, per-candidate voting mechanism (e.g., our Binary voting) is optimal in the coordinate-independent setting (Definition 5 below).

Definition 5. *A function $f(X) : \mathcal{R}^d \rightarrow \mathcal{R}^k$ is coordinate-independent if the i'th output coordinate of f, f_i, is determined only by a unique subset $P_i(X)$ of the input X. P_i is a function that partitions (selects) columns of its input X and uniqueness means that that $\forall i, j$ if $i \neq j$ then $P_i \cap P_j = \emptyset$ so no columns are shared by any P_i.*

We first build an intuition for our guarantee and then provide the formal statement (in Proposition 3.1). Recall from Section 2 that the crux of DP analysis is to bound the sensitivity of the random mechanism $\mathcal{M}$, which is the noisy output of the function[2] f.

Definition 6. *Sensitivity is the maximum deviation*

$$\Delta_p f = \max_{(X,X'):||X-X'||_1=1} ||f(X) - f(X')||_p$$

of f's output for any adjacent pair of inputs X and X′, i.e., have a Hamming distance $||X - X'||_1$ of 1, on their rows [20].

Recall $\varepsilon_{\mathcal{M}}(\lambda) \propto \Delta_p \mathcal{M}$; thus, we seek to decompose a multi-label function (with range k) in terms of the sensitivities of its corresponding k single-label functions, i.e., the scenario of Binary voting. Observe that each output $f_i : i \in [k]$ of a Binary voting mechanism depends only on a unique subset of columns $P_i(X)$ of the input ballots X. Without loss of generality, let each output coordinate be binary, i.e., $f_i(X) \in \{0, 1\}$, and let each output depend on only one input column, i.e., let $d = k$ and $P_i(X) = X_i$ such that $f_i(X) = f_i(X_i)$. Then, $\Delta_p f$ is upper bounded by $k^{1/p}$. To see this, recall sensitivity is defined as the function's maximum p-norm deviation between any (but only) two datasets X and X'. Thus, pick X and X' so that each binary output $f_i(\cdot)$ is flipped between predicted (1) and not predicted (0). If f is coordinate-independent, then its sensitivity will always be $k^{1/p}$ because $f_i(X) = f_i(X_i)$ and thus X (X') are the horizontal stacks of X_i (X_i') such that $f_i(X_i) - f_i(X_i') = 1$. In other words, the sensitivity of the multi-label function is the sum of sensitivities of the single-label functions that comprise it. Because Binary voting is applied independently on each single-label function, it thus achieves the optimal sensitivity in the coordinate-independent setting. Proposition 3.1 below formally states our result. Appendix A contains supporting lemmas and the proofs.

Proposition 3.1. *For a coordinate-independent multi-label function $f(X) : \mathcal{R}^d \rightarrow \mathcal{R}^k$, the ℓ_1 sensitivity $\Delta_1 f$ is equal to that of Binary voting applied per label and thus simultaneous private release of f has no tighter privacy loss than k applications of Binary voting.*

Proposition 3.1 shows that Binary voting has optimal sensitivity if f is coordinate (candidate) independent. However, if the function were coordinate-dependent, Binary voting may be sub-optimal. To see this, consider the case where output coordinates i and j both solely depend on the input coordinate l. Then it may not be possible to find a pair of databases that simultaneously flips both outputs i and j. For instance, if max f_i occurs at $X_l = a$ and min f_i at $X_l = b$, but one of max f_j or min f_j does not occur at these values, then now $\Delta_p f_j$ cannot be maximized while simultaneously maximizing $\Delta_p f_i$. Thus, $\Delta_p f < \sum_i^k \Delta_p f_i$. Since the sensitivity may be less, the privacy loss may also be lower. This implies that if the underlying function is coordinate-dependent, Binary voting may now have larger sensitivity and thus a sub-optimal privacy loss. We will discuss one method for leveraging such correlations in Definition 9 and this overall approach in Section B.

We also find that Binary voting may be sub-optimal when the sensitivity is calculated via $p > 1$ norms. This is a direct result of Hölder's Inequality [26]. We explore this in Section 3.2 by clipping in the ℓ_2 norm.

3.2 DP Mechanisms for τ-Multi-Winner Election

Many real-world ballots may not have many chosen candidates, e.g., in Pascal VOC there are on average fewer than 2 positive labels out of 20 , enabling $\tau < \sqrt{k}$. Thus, instead of allowing arbitrarily many votes per ballot, the τ-multi-winner election restricts the votes in their ℓ_2 norm, in turn limiting the sensitivity of the mechanism. In doing so, tighter data-independent privacy guarantees are obtained. However, this can come at a cost in utility if ballots were cast with more candidates selected than the chosen τ allows for, because the ballot must now be 'clipped'. This is analogous to *norm clipping* in DPSGD [1], which is commonly studied in DP literature. To be meaningful, we require $\tau < \sqrt{k}$; otherwise, this is equivalent to Binary voting (Definition 4).

Definition 7 (DP τ Voting Mechanism). ***Mechanism for τ-Multi-Winner Election.*** *For a τ-multi-winner election the τ Voting mechanism is*

$$\mathcal{M}_\sigma(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n) \triangleq \left\{ i : \left(\sum_{j=1}^{n} \boldsymbol{v}_j[i] + \mathcal{N}(0,\sigma_G^2) \right) > T_i \right\},$$

$$\boldsymbol{v}_j \triangleq \min(1, \frac{\tau}{\left\|\boldsymbol{b}_j\right\|_2})\boldsymbol{b}_j,$$

where we choose $T_i = n - \sum_{j=1}^{n} \boldsymbol{v}_j[i]$, which represents the number of *clipped* negative votes. The clipping itself was done on the

[2] Because the random mechanism differs minimally from the function, we use the two interchangeably.

positive votes (through the v_j's). To obtain the number of negative votes, we subtract the number of clipped positive votes from the total number of votes (in this case n) and thus call these *clipped* negative votes. To analyze the privacy loss of this new mechanism, we propose an extension of Lemma 2.1 for this multi-voting task.

Lemma 3.2. ***RDP-Gaussian mechanism for a multi-label setting**. Let $f : \mathcal{X} \rightarrow \mathcal{R}^k$ obey $\|f(X) - f(X')\|_2 \leq \Delta_2$ for neighboring datasets X, X'. The Gaussian mechanism $\mathcal{M}(X) = f(X) + \mathcal{N}_k(0, \sigma^2 I)$ obeys RDP with $\varepsilon_{\mathcal{M}}(\lambda) = \frac{\lambda \Delta_2^2}{2\sigma^2}$.*

The following proof is similar to the one for Proposition 7 in [33] (the crucial steps are 6 and 7 to transform the integral and obtain the Gauss error function):

Proof.

$$D_\lambda(\mathcal{M}(X)||\mathcal{M}(X')) \tag{1}$$

$$= D_\lambda(\mathcal{N}_k(f(X), \sigma^2 I)||\mathcal{N}_k(f(X'), \sigma^2 I)) \tag{2}$$

$$= \frac{1}{\lambda - 1} \log \left\{ \frac{1}{\sigma^k \sqrt{(2\pi)^k}} \int_{-\infty}^{\infty} \exp\left(\frac{-\lambda}{2\sigma^2} \|\theta - f(X)\|_2^2 \right) \right. \tag{3}$$

$$\left. \cdot \exp\left(\frac{-(1-\lambda)}{2\sigma^2} \left\|\theta - f(X')\right\|_2^2 \right) d\theta \right\} \tag{4}$$

$$= \frac{1}{\lambda - 1} \log \left\{ \frac{1}{\sigma^k \sqrt{(2\pi)^k}} \int_{-\infty}^{\infty} \right. \tag{5}$$

$$\left. \exp\left(\frac{-\lambda \|\theta - f(X)\|_2^2 - (1-\lambda) \|\theta - f(X')\|_2^2}{2\sigma^2} \right) d\theta \right\} \tag{6}$$

$$= \frac{1}{\lambda - 1} \log \left\{ \frac{\sigma^k \sqrt{(2\pi)^k}}{\sigma^k \sqrt{(2\pi)^k}} \exp \frac{(\lambda - 1)\lambda}{2\sigma^2} \left\|f(X) - f(X')\right\|_2^2 \right\} \tag{7}$$

$$= \frac{\lambda \|f(X) - f(X')\|_2^2}{2\sigma^2} \leq \frac{\lambda \Delta_2^2}{2\sigma^2} \tag{8}$$

■

3.2.1 Norms for Clipping. Recall that RDP provides tight privacy analysis for the Gaussian mechanism and that it analyzes sensitivity using the ℓ_2 norm. This motivates our choice of ℓ_2 vote clipping, because it exactly bounds the mechanism's sensitivity. Compared with the ℓ_1 clipping proposed in [51] our bound is no larger, and often much *smaller*, than theirs. We can see this by comparing the privacy loss in terms of the mechanism's sensitivity (see Definition 6).

$$\epsilon_{\mathcal{M}}(\lambda) = \frac{\lambda \tau_2^2}{\sigma^2} = \frac{\lambda \Delta_2^2}{2\sigma^2} \leq \frac{\lambda \Delta_1^2}{2\sigma^2} = \frac{\lambda (2\tau_1)^2}{2\sigma^2} = \frac{\lambda 2 \tau_1^2}{\sigma^2} \tag{9}$$

where in the first inequality we use the fact that $\|x\|_2 \leq \|x\|_1$. Empirically, we confirm that τ voting achieves a much tighter bound on many multi-label datasets. Take τ_1 to represent clipping in the ℓ_1 norm and similarly τ_2 to represent clipping in the ℓ_2 norm. On Pascal VOC with $k = 20$ labels, we find that we can set a much smaller $\tau_2 = 1.8$ but only a $\tau_1 = 3.4$ without deteriorating the mechanism's balanced accuracy. This leads to a >6x tighter (data-independent) privacy loss with the ℓ_2 norm clipping compared to [51] and Binary voting. In practice, we select τ to be marginally larger than the average p-norm of labels in the training data.

3.3 Casting ∞-Multi-Winner Elections as Single-Winner Elections

Similar to Section 3.1, we again explore how single-winner elections can be used to reveal outcomes of multi-winner elections. Instead, here we remove the independence assumption and cast the multi-winner election to one single-winner election (not k independent ones). This improves on one of the major shortcomings of Binary voting: by simultaneously revealing all k candidate's results in one outcome, we can leverage correlations in their results to reduce the privacy loss.

We achieve this new mechanism by encoding each of the 2^k different outcomes as a separate candidate for a single-winner voting mechanism. We call this mechanism the Powerset voting mechanism.

Definition 8 (DP Powerset Mechanism). *Denote the Powerset operator as $\mathcal{P}(\cdot)$. For a ∞-multi-winner election with ordered outcomes $\mathcal{P}(\{0,1\}^k)$, where $|\mathcal{P}(\{0,1\}^k)| = 2^k$,*

$$\mathcal{M}_\sigma(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \{0,1\}^k$$

$$\triangleq \arg\max_i \left\{ \mathcal{P}(\{0,1\}^k)_i : \sum_{j=1}^{n} \mathbb{1}_{\boldsymbol{b}_j = \mathcal{P}(\{0,1\}^k)_i} + \mathcal{N}(0, \sigma_G^2) \right\}$$

where $\mathbb{1}$ represents the indicator function. Importantly, this mechanism only approximates the true Multi-Winner solution because it can only reveal an outcome that was cast as a ballot. If the set of true plurality candidates had no associated ballot, then this result cannot be returned. However, with enough ballots, the resulting utility degradation is reduced.

4 FROM PRIVATE ELECTIONS TO PRIVATE MULTI-LABEL CLASSIFICATION

Single-winner elections, and their DP mechanisms, have played an important role in privacy-preserving machine learning. Notably, PATE adopted the noisy arg max mechanism to enable training of single-label models via semi-supervised knowledge transfer. By replacing the noisy arg max mechanism with our proposed DP multi-winner mechanisms, we thus create new multi-label PATE systems for private multi-label semi-supervised learning.

Binary PATE, τ PATE, and Powerset PATE: We refer readers to Section 2.2 for a detailed description of single-label PATE. Multi-label PATE is nearly unchanged from a systems perspective because we only replace the DP mechanisms used. Notably, both forms of PATE leverage a bank of unlabeled data which is used for the semi-supervised knowledge transfer; this data is often widely available since it is unlabeled [35]. Each query uses some $\varepsilon - DP$ which, once the total composed RDP budget is exhausted, terminates the labelling process. Our mechanisms are used to label and calculate the per-query ε. These m privately labeled data points are then used to train the student model which can be released. All of our mechanisms are compatible with propose-test-release [18] (used in Confident GNMax) which we leverage to reduce the privacy budget usage. We show the full τ-PATE in Algorithm 1.

5 ANALYSIS OF MULTI-LABEL PATE

We now analyze the privacy guarantees of our mechanisms within multi-label PATE. Importantly, we have designed our mechanisms

Algorithm 1 τ **PATE**: Multi-label classification with τ-clipping in ℓ_2-norm and with the Confident GNMax.

```
Input: Data point $x$, clipping threshold $\tau_2$, Gaussian noise scale $\sigma_G$, Gaussian noise scale for Confident GNMax $\sigma_T$, $n$ teachers, each with model $f_j(x) \in \{0,1\}^k$, where $j \in [n]$.
Output: Aggregated vector $V$ with $V_i = 1$ if returned label/feature present, otherwise $V_i = 0$.
for all teachers $j \in [n]$ do
    $v_j \longleftarrow \min(1, \frac{\tau_2}{\|f_j(x)\|_2}) f_j(x)$        ▷ $\tau$-clipping in $\ell_2$ norm
$V^1 = \sum_{j=1}^{n} v_j$        ▷ Number of positive votes per label
$V^0 = n - V^1$        ▷ Number of negative votes per label
for all labels $i \in [k]$ do
    if $\max\{V_i^0, V_i^1\} + \mathcal{N}(0, \sigma_T) < T$ then
        $V_i = \perp$        ▷ Confident-GNMax
    else
        $V_i^0 \longleftarrow V_i^0 + \mathcal{N}(0, 0.5\sigma_G^2)$        ▷ Add Gaussian noise
        $V_i^1 \longleftarrow V_i^1 + \mathcal{N}(0, 0.5\sigma_G^2)$        ▷ for privacy protection
        if $V_i^1 > V_i^0$ then        ▷ Decide on the output vote
            $V_i = 1$
        else
            $V_i = 0$
```

to be compatible with the data-dependent analysis of [36] (by making them extensions of the noisy *argmax*) because it is often tighter than the data-independent guarantee which we discuss below.

5.1 Data-Independent and Data-Dependent Analysis

Data Independent Analysis: We first compare our data-independent guarantees with respect to the noisy arg max mechanism in a single-label setting, for one query. When it satisfies a $\varepsilon_{\mathcal{M}}(\lambda)$-RDP, our Binary voting achieves a data-independent bound of $\sum_{i=1}^{k} \varepsilon_{\mathcal{M}}(\lambda)_i$, τ-voting of $\frac{\lambda\tau^2}{\sigma^2}$, and Powerset voting of $\varepsilon_{\mathcal{M}}(\lambda)$.

Data-Dependent Analysis: However, tighter privacy budgets may be reported when there is high empirically observed consensus on the binary vector, *i.e.,* there is a high likelihood for the returned outcome. The tight data-dependent analysis of [36] bounds the privacy loss in terms of the probability of the returned single-label outcome, calculated from the histogram of teacher votes. This data-dependent privacy loss is often much lower than the data-independent bounds above—even after the required sanitation before release. This sanitation is achieved by smooth sensitivity analysis of the mechanism to protect against information leakage resulting from the release of the budget—because its particular depends on the data itself. We follow the steps and additionally report the expected sanitized privacy budgets.

Because each of our multi-label mechanisms view the votes in different manners, the resulting data-dependent privacy budgets can vary significantly. Below, we compare the data-dependent guarantees for Binary and Powerset PATE, where we empirically observe that τ PATE only marginally differs from Binary PATE resulting from how ℓ_2 clipping impacts the vote distributions.

5.2 Comparing Binary and Powerset PATE

We begin with preliminaries on our Binary and Powerset mechanisms, and the data-dependent DP analysis of [36].

Recall that Binary PATE has $2 \cdot k$ events: the presence or absence independently for each of the k labels. Instead, Powerset PATE operates on the entire binary vector, where each configuration (subset) of this vector is an event—thus, there are 2^k events. This leads to three concrete differences of Powerset PATE: (1) it has exponentially more possible events, (2) when any labels $i \in [k]$ occur simultaneously, it rapidly improves the privacy loss, and (3) it only adds noise once to the entire binary vector. These directly impact the calculated data-dependent bounds.

The data-dependent privacy bound is a function of the probability distribution between events, estimated using the observed votes as follows. Calculate the difference between the two most likely events (label presence/absence in Binary voting, binary vector subsets in Powerset voting). The gap is this difference scaled by the inverse of the noise standard deviation, which bounds the probability of any non-plurality event occurring. To provide a final guarantee for the mechanism, as in single-label PATE, we upper bound the Renyi divergence using Theorem 5.2 below. The main random variable influencing this privacy loss is the gap, $q(\bar{n})$, which we calculate using Proposition 5.1 below. Theorem 5.2 shows how to select ideal higher order moments μ_1, μ_2 and $\varepsilon_1, \varepsilon_2$, using the data-dependent value for $q(\bar{n})$ so that the RDP bound is a nonlinear function of $q(\bar{n})$ solely.

Proposition 5.1 (From [36]). *For a GNMax aggregator $\mathcal{M}_\sigma$, the teachers' votes histogram $\bar{n} = (n_1, \cdots, n_k)$, dataset X, and for any $i^* \in \mathcal{P}(\{0,1\}^k)$, where $\mathcal{P}(\cdot)$ denotes the Powerset, we have*

$$\mathbf{Pr}[\mathcal{M}_\sigma(X) \neq i^*] \leq q(\bar{n}),$$

and

$$q(\bar{n}) \triangleq \frac{1}{2} \sum_{i \neq i^*} erfc\left(\frac{n_i^* - n_i}{2\sigma}\right)$$

Where this is a minimal modification of [36] to go from single-label PATE to multi-label Powerset PATE.

Theorem 5.2 (From [36]). *Let $\mathcal{M}$ be a randomized algorithm with (μ_1, ε_1)-RDP and (μ_2, ε_2)-RDP guarantees and suppose that there exists a likely outcome i given a dataset X and bound $\tilde{q} \leq 1$ such that $\tilde{q} \geq \Pr[\mathcal{M}(X) \neq i]$. Additionally suppose the $\lambda \leq \mu_1$ and $\tilde{q} \leq exp^{(\mu_2-1)\varepsilon_2}(\frac{\mu_1}{\mu_1-1} \cdot \frac{\mu_2}{\mu_2-1})^{\mu_2}$. Then for any neighboring dataset X' of X, we have:*

$$D_\lambda(\mathcal{M}(X) || \mathcal{M}(X')) \leq$$

$$\frac{1}{\lambda - 1} \log\left((1-\tilde{q}) \cdot \mathbf{A}(\tilde{q}, \mu_2, \varepsilon_2)^{\lambda-1} + \tilde{q} \cdot \mathbf{B}(\tilde{q}, \mu_1, \varepsilon_1)^{\lambda-1}\right)$$

where $\mathbf{A}(\tilde{q}, \mu_2, \varepsilon_2) \triangleq (1-\tilde{q}) / \left(1 - (\tilde{q}e^{\varepsilon_2})^{\frac{\mu_2-1}{\mu_2}}\right)$ *and* $\mathbf{B}(\tilde{q}, \mu_1, \varepsilon_1) \triangleq exp^{\varepsilon_1} / \tilde{q}^{\frac{1}{\mu_1-1}}$.

When the top 3 vote counts, $n_1 > n_2 > n_3$ satisfy $n_1 - n_2, n_2 - n_3 \gg \sigma$ (the gap is sufficiently large), the RDP bound can be well approximated as

$$\varepsilon_{\mathcal{M}_\sigma}^{powerset}(\lambda) \leq exp(-2\lambda/\sigma^2)/\lambda, \text{ where } \lambda = (n_1 - n_2)/4, \tag{10}$$

which is from [36, Corollary 11]. Though $q(\bar{n})$ is the formal gap, we will often refer to $\tilde{q}(\bar{n}) = n_1 - n_2$ as the gap too when these assumptions are satisfied. We plot the cdf of the gaps across votes in CheXpert and Pascal VOC in Figure 8 of Appendix F, finding that Binary PATE is indeed in this regime. Assuming Powerset PATE is too, we analyze under what circumstances it outperforms Binary PATE. To do this, we will analyze the respective data-dependent bounds under the expected gap $\mathrm{E}[\tilde{q}(\bar{n})]$.

5.3 Data-Dependent Binary PATE Analysis

We aim to specify $\mathrm{E}[\varepsilon_{binary}]$ for Binary PATE in terms of the probability of the most likely outcome, denoted p. By using the same p for Powerset PATE, we can compare how the privacy budgets vary with respect to p.

We model each teacher's independent label prediction as a Bernoulli trial of probability p_i, $i \in [k]$. With t independent trials, we can model each gap as a binomial distribution of probability p_i. To simplify our exposition we assume $\forall i, p_i = p$ and will later discuss how this impacts our analysis. Using composition, we get an expected privacy loss of

$$\mathrm{E}[\varepsilon_{binary}] = k \cdot \sum_{\tilde{q}=\tilde{q}'}^{\tilde{q}=t} \left(P(\tilde{q} = \tilde{q}') \frac{4}{\tilde{q}} exp\left(\frac{-\tilde{q}}{2 \cdot \sigma^2} \right) \right)$$

$$= k \cdot \sum_{\tilde{q}=\tilde{q}'}^{\tilde{q}=t} \left(\binom{t}{\tilde{q}} p^{\tilde{q}} (1-p)^{1-\tilde{q}} \frac{4}{\tilde{q}} exp\left(\frac{-\tilde{q}}{2 \cdot \sigma^2} \right) \right)$$

Due to the assumptions of Equation 10, we require that low gap events ($\tilde{q}' \rightarrow 0$) have sufficiently low probability so that they can be ignored. Figure 8 of Appendix F provides justification for this.

5.4 Data-Dependent Powerset PATE

We now estimate how $\mathrm{E}[\tilde{q}(\bar{n})]$ changes with changes in p for Powerset PATE—this directly impacts $\mathrm{E}[\varepsilon^{powerset}_{\mathcal{M}_\sigma}]$. This is more involved because it operates over the entire binary vector. If each label has probability $p_i = p$ of being present, then we can express the probability (denoted P, note the upper case) of an outcome (binary vector subset) as the union of these probabilities. Though exact, this is infeasible to calculate and analyze for all 2^k subsets when k is large. This is one reason we estimate only $\tilde{q}(\bar{n})$ rather than $q(\bar{n})$— we need only estimate n_1 and n_2.

We define the Powerset mechanism in Definition 9 of Appendix A, which is similar to Powerset Voting (Definition 8) except that we use standard PATE notation. Here, our histogram n counts subsets with $n_i(x)$ representing the vote count for the i−th subset.

We sort the votes $n_i(x)$ such that $n_i \geq n_j \forall i > j$. We seek to upper bound our best-case expected privacy loss for Powerset PATE. Using Equation 10, we get

$$\mathrm{E}[\varepsilon^{powerset}_{\mathcal{M}_\sigma}(\tilde{q})] = \int \frac{4}{\tilde{q}} exp(\frac{-\tilde{q}}{2 \cdot \sigma^2}) PDF(\tilde{q}) d\tilde{q},$$

where $PDF(\tilde{q})$ is the probability density function of $\tilde{q}$ and we again require that low gap events $\tilde{q} < 1$ have no density.

$\mathrm{E}[\varepsilon^{powerset}_{\mathcal{M}_\sigma}(\lambda)] \rightarrow 0$ as the density of $\tilde{q} \rightarrow \infty$ as expected. The base case gap with $O(1)$ probability can be found by viewing the binary vectors of Powerset PATE as a non-uniform balls and bins problem. We have 2^k subsets, i.e., $|\mathcal{P}(\{0,1\}^k)| = 2^k$. Each subset $i \in [2^k]$ is represented by a bin with probability P_i of having a ball land in it (be voted on by a teacher). The t teachers ($\Sigma_i \bar{n_i} = t$) each vote independently for a bin (subset).

We estimate $\tilde{q}$ by calculating the load of the maximally loaded bin (most voted outcome), denoted as τ, when Powerset PATE has the highest gap, i.e., where each other bin has at most 1 ball. Then, $\tilde{q} = \tau - 1$; by upper-bounding τ, we upper bound the gap.

To upper-bound τ, we use an indicator random variable C_c representing the event of c collisions occurring in any bin, *i.e.,* c votes for that bin. Using Markov's inequality followed by Stirlings approximation, we get (see also Appendix I):

$$Pr[C_c \geq 1] \leq \mathrm{E}[C_c] = \binom{t}{c} \sum_{i}^{2^k} P_i^c \leq (\frac{t \cdot e}{c})^c \sum_{i}^{2^k} P_i^c.$$

As with $\bar{n}$, we sort our probability $[P_1, \cdots, P_{2^k}]$ such that $P_1 \geq P_2 \cdots \geq P_{2^k}$. Because $n_1 \gg n_2$, we have that $\forall i \neq 1,\ P_1 \gg P_i$. As the gap increases, $P_1 \rightarrow 1$, and $P_i^c \rightarrow 0$, we get that $\sum_i^{2^k} P_i^c \leq P_1^c$ and thus $Pr[C_c \geq 1] \leq (\frac{t \cdot e}{c})^c P_1^c$. In particular, we care about the regime where the max load occurs with high-probability ($Pr[C_c] \rightarrow 1$):

$$\begin{aligned} Pr[C_c \geq 1] \leq (\frac{t \cdot e \cdot P_1}{c})^c &\rightarrow 1 \\ \exp(\ln(\frac{t \cdot e \cdot P_1}{c})^c) &\rightarrow 1 \\ \implies c \cdot \ln(t \cdot P_1) + c - c \cdot \ln(c) &\rightarrow 0 \\ \implies c \cdot \ln(t \cdot P_1) + c &= c \cdot \ln(c). \end{aligned} \tag{11}$$

Specifying t and P_1, we can then directly calculate the max collisions c, then $\tilde{q}$, and finally the best-case expected privacy loss. In the next Section 5.5, we analyze how the max load changes with respect to P_1.

5.5 Privacy Analysis Comparison

From the assumptions above, we can directly calculate each P_i, for Powerset PATE. We do this by recognizing that a single teacher's binary vector can be modeled as a binomial distribution of k (the number of labels) trials and probability p. To calculate the probability of any one of the 2^k possible binary vectors from the Powerset vector, we express each one as a l−hot binary vector (l labels present, independent of their coordinate position). Each possible subset satisfying l−hot will have equal probability of occurring (because we fixed p for all labels). Then, the maximum probability binary vector, when k sufficiently large ($k > 5$, approximately) and $p \rightarrow 0$ or $p \rightarrow 1$ is when $l = 0$ or $l = k$ as otherwise the combinatorics leads to a probability split amongst too many choices. This implication may appear unnatural at first; however, it well models our output label distribution: when there are a set of label coordinates that share similarly high (or low) probabilities of predicting presence (or absence) of that label, then their most likely binary vector is all 1's or 0's. We will next analyze the three major contributors to changes in the gap.

Impact of Random Labels. To model more complex distributions where in general $p_i \neq p$, we can bucket the ranges of p present and analyze these buckets separately. Let us operate under the assumption of coordinate-independence. In this case, we have the maximal variance for the binomial distribution and thus largest gaps for Powerset and Binary PATE. In this case, we expect both methods to perform poorly: however, because the data-independent bound for Powerset PATE is tighter than for Binary PATE, Powerset PATE performs better, as shown in Figure **??**. This gives us our first two observations: (1) *Powerset PATE has a better worst-case privacy loss (data-independent bound)* and (2) *both Powerset and Binary PATE degrade to their data-independent bounds as* $p \longrightarrow 0.5$.

Impact of Correlation (Coordinate Dependence). Even when a bucket of labels has $p \approx 1$ or $p \approx 0$ (but not equal), we can still have a varying distribution of outcomes. In particular, with coordinate-independence, we can directly use our binomial analysis above and find that there is a uniform probability for all outcomes in each $l-$hot vector. In particular, there are factorially many outcomes each with equal probability. This drastically degrades the gap by reducing P_1. However, when there is coordinate-dependence, e.g., the best-case when each label's value directly implies the rest, then we have only two outcomes for each $l-$hot vector. This improves the gap by reducing the possible subsets and improving P_1. We formalize this as follows: if we have some subset of d vectors that are dependent, then we can reduce k in our above analyses to $(k-d)$. This gives us our third observation: (3) *coordinate dependence has a combinatoric improvement for Powerset PATE and at best a linear improvement for Binary PATE.*

Impact of Higher Noise Multipliers. Appealing back to the data-dependent $q(\bar{n})$ calculation of Proposition 5.1, we see that Binary PATE has a gap calculated as $2P - t/2\sigma$ for t teachers. This gives $q^{binary}(\bar{n}) = erfc(2P - t/2\sigma)$ However, for a similar gap, which is bounded for both to a $\max(\tilde{q}(\bar{n})) = t$, we see that Powerset PATE will have $q^{powerset}(\bar{n}) = \sum_i^{2^k} erfc(\tilde{q}(\bar{n})/2\sigma)$, which is always worse when the *gaps* are equal. We can see this effect in Figure **??**. Further, because we union bound across the 2^k classes, a higher noise multiplier has a larger impact on Powerset PATE, as shown in Figure 2. These give our fourth and fifth observations: (4) *in the best case, and for any equal gaps, Powerset PATE has a higher privacy loss*, and (5) *higher noise multipliers will impact Powerset PATE more for similar gaps.*

Overall, we find that there are two regimes where Powerset voting (PATE) outperforms Binary voting (PATE). *The first regime is when there are high correlations between the multi-winner candidates*, i.e., there is strong coordinate dependence. We found empirical evidence of this case, as we will describe later, in Figure 3. *The second regime is when neither mechanism can leverage stronger data-dependent privacy guarantees*, the data-independent guarantees of Powerset PATE are tighter (see Section 5.1). We find that though Powerset PATE may not always be capable of returning the plurality vote, the utility of the votes remains high (see Figure 2).

5.6 Choosing the PATE Mechanism

As we will see from the empirical results in the next section, we recommend choosing τ PATE in all cases where a reasonable $\tau < \sqrt{k}$ can be chosen based on the average ℓ_2 norm of the training data. Otherwise, Binary PATE is generally preferred, unless there is a significant correlation between labels detected (we show one example of this analysis in Appendix B).

6 EMPIRICAL EVALUATION

We compare our proposed mechanisms for multi-label classification on real-world datasets. Our goal is to answer the following questions: (1) How do the multi-label PATE methods work in practice and which of them should be selected by practitioners? (2) In the centralized learning setting, how does our proposed multi-label PATE compare with the other method of choice - DPSGD? Finally, (3) how our private multi-label classification performs in the framework of a distributed collaborative learning system, such as CaPC, which preserves privacy and confidentiality? We observe that the τ PATE is the preferred multi-label method that outperforms other PATE-based methods as well as DPSGD, and can provide benefits when used by collaborating parties.

6.1 Experimental Setup

We carry out the evaluation on three medical datasets CheXpert [28], MIMIC-CXR [29], PadChest [9], and on a vision dataset Pascal VOC 2012 [22]. We use DenseNet-121 [27] for the medical datasets and ResNet-50 [25] for the vision dataset. DenseNet has been shown to be the best architecture for X-ray data [13]. See Appendix C for dataset and model descriptions. Our main exposition chooses safe $\delta < 1/N$ where N is the number of records in the dataset. Only one experiment uses $\delta = 1e - 4$ to compare directly with [48]. We experiment with $(\varepsilon = 8, \delta = 1E - 4)$ for 5 labels on CheXpert (see Appendix F.2), $(\varepsilon = 10, \delta = 1E - 5)$ for predictions on PascalVOC, $(\varepsilon = 20, \delta = 1E - 6)$ for 11 labels on CheXpert or MIMIC, and also $(\varepsilon = 20, \delta = 1E - 6)$ for 15 labels on PadChest (see also Section 6.5). As an example, we show how the number of answered queries decreases gradually with orders of magnitude lower δ values for the PascalVOC dataset in Table 14. We note that though τ should use DP protection [37], it is commonplace to tune hyperparameters non-privately. Our mechanisms add a negligible run-time performance overhead on top of PATE, where the main bottleneck is the training of the teacher models. [3]

Evaluation Metrics. We use the following metrics: (1) accuracy (ACC) - a proportion of correct predictions per label, (2) balanced accuracy (BAC) - mean recall and specificity per class [8], (3) area under the receiver operating characteristic curve (AUC) [24], and (4) mean average precision (MAP), where average precision per class is the area under the precision-recall curve. We compute each metric per label using methods from *sklearn.metrics* and then average them across all the labels. The definitions per label of the metrics are as follows: (1) accuracy per label (ACC): $\frac{TP+TN}{TP+TN+FP+FN}$, (2) Balanced Accuracy per label (BAC) = $\frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP})$, where TP = True Positive; FP = False Positive; TN = True Negative; FN = False Negative, (3) Area-Under-the-Curve (AUC) = $\int_0^1 t(f)\ df$, where $t(f)$ is the TP rate against the FP rate, (4) mean Average Precision (MAP) = $\frac{1}{n}\sum_{i=1}^{n} P_i$, where P_i is the precision for the i-th label equal to $\frac{TP}{TP+FP}$ for that label and n is the number of labels per example. More details on the evaluation metrics can be found

[3] Our code can be found at this link: https://github.com/anonymous-user-commits/private-multi-winner-voting.

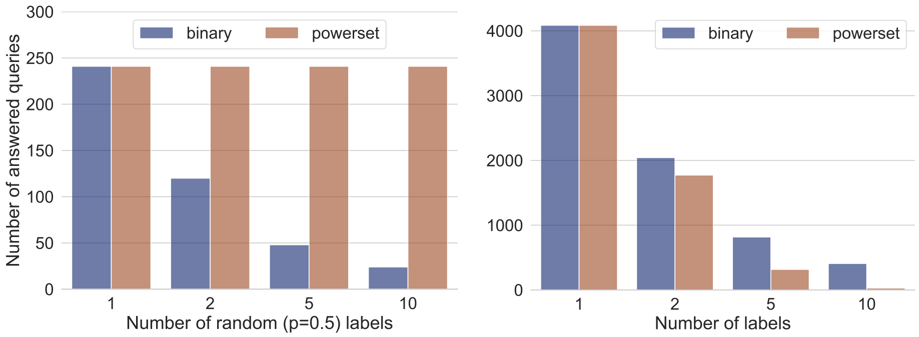

Figure 1: LEFT: Powerset PATE outperforms Binary PATE as more labels are generated randomly, i.e., p=0.5 with $\varepsilon = 20$. If all votes are random, then both Binary and Powerset PATE fall back on the data independent bound. RIGHT: Binary PATE outperforms Powerset PATE when they both have a similarly high gap.. Here, all teachers always output 0 for all labels. We use $\varepsilon = 2$. BOTH: we use 50 teachers and privacy noise $\sigma_{GNMax} = 7$.

in Appendix D and further information on the experimental setup in Appendix E.

6.2 Queries Answered vs Label Count

We analyze how our methods scale with a different number of labels in a dataset. We set the privacy budget $\varepsilon = 20$ throughout the whole experiment. The scale of the Gaussian noise σ_{GNMax} is adjusted per label count so that we can preserve high values of the metrics (ACC, BAC, AUC, MAP), with maximum performance drop by a few percentage points. We find the values of σ_{GNMax} separately for the Binary and Powerset methods so that their performance metrics are the same for a given label count. For example, in Figure 2 for 11 labels, we select $\sigma_{GNMax} = 11$ for Binary PATE and $\sigma_{GNMax} = 4$ for Powerset PATE, where the values of the metrics for both methods are ACC=.95, AUC=.66, and MAP = .37 on Pascal VOC. For the CheXpert dataset, we set $\sigma_{GNMax} = 20$ for Binary PATE and $\sigma_{GNMax} = 5$ for Powerset PATE, where the values of the metrics for both methods are ACC=.71, AUC=.68, and MAP = .45. Here, we observe that the private multi-label classification with Binary PATE performs better and preserves higher values of the metrics with more noise added, compared to the Powerset PATE method.

Next, we compare the Binary and Powerset methods across many labels and in terms of how many queries can be answered privately for a given number of labels in the dataset. We run the experiment on the Pascal VOC and CheXpert datasets and present results in Figure 3 (as well as in Tables 11-12 in the Appendix). Our analysis shows that the private multi-label classification with Binary PATE performs better (answers more queries) than Powerset PATE for Pascal VOC. The CheXpert dataset is more noisy in terms of labeling, the pathologies are sparse and difficult to detect; this dataset does not give us a clear preference of the multi-label PATE method in terms of the number of answered queries.

6.3 Query-Utility Tradeoff in PATE

In comparing how the privacy parameter σ_G impacts the query-utility tradeoff of each mechanism, under a fixed $\varepsilon = 20$ across all datasets, we find that *under high consensus Binary voting performs best and under lower consensus τ voting performs best.* In Figure 4, we compare each multi-winner election mechanism used in τ PATE and find that each mechanism has a range of σ_G where it performs best. In particular, as $\sigma_G \to 0$, no queries can be answered by any mechanism because doing so would require more than the allotted privacy budget ε. For sufficiently small σ_G and with suitable consensus amongst voters, which is the case for Pascal VOC, CheXPert, and MIMIC-CXR, we find the data-dependent analysis for Binary voting outperforms all others while remaining in a regime of high-performance metrics. As $\sigma_G \to \infty$, the τ voting mechanism outperforms all others, though in most cases at a decrease in the performance metrics; however, on PadChest, this is the best bound. This can be explained by the fact that on PadChest we can only train 10 teachers before each individual model's accuracy degrades too much. An important distinction is that τ PATE can perform worse than Binary PATE when the chosen τ bound is too small (because ballot clipping can change the vote distribution). For well-chosen values of τ we observe only marginal decrease in the number of queries answered and the performance metrics (c.f. Figure 4 with Figures 11, 12, and 13 in Appendix F). Inspecting the tradeoff of Powerset PATE (see Figure 2), we find that a much lower noise can be tolerated before a steep decline in performance metrics. Because of this, we find that much fewer queries can be answered (see Tables 11 and 12). Thus, we recommend τ PATE as the de-facto mechanism.

6.4 Private Centralized Learning

We now show that even in the centralized setting, our multi-label PATE methods outperform the competitive baselines. We train models on Pascal VOC and CheXpert which include 20 and 5 labels for this experiment, respectively. We leverage the entire training set to train a single non-private model and a single private model via DPSGD [2]. For our multi-label PATE we instead train 50 teachers each on a separate disjoint partition of the centralized training set (and thus, with $1/50$ number of samples compared with DPSGD and the non-private model). We use these teacher models to privately train a student model using semi-supervised learning with MixMatch [6] (where modifications are made to adapt MixMatch to the multi-label setting). Though the DPSGD algorithm does not assume

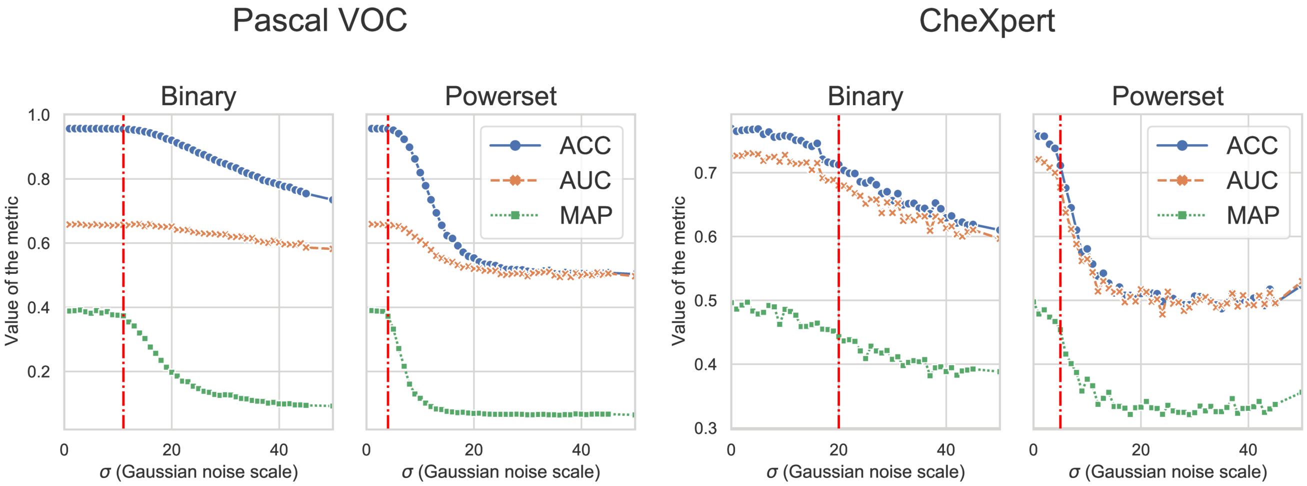

Figure 2: Binary vs Powerset PATE performance. We compare the utility of the private votes returned in terms of ACC, AUC, and MAP, with respect to Gaussian noise scale σ_{GNMax}. We select the first 11 labels in Pascal VOC and all 11 labels from CheXpert. We find that Binary PATE can tolerate a much higher noise scale σ_{GNMax} compared with Powerset PATE before significant degradation in the votes is observed.

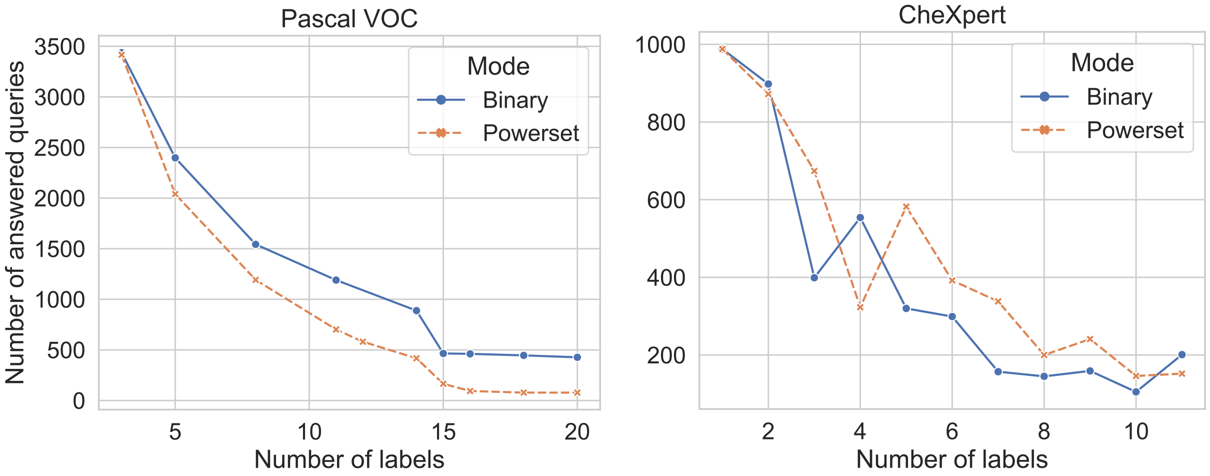

Figure 3: Binary vs Powerset PATE: number of answered queries. We compare the number of answered queries vs the number of k first labels selected from the Pascal VOC and CheXpert datasets. We keep the privacy budget $\varepsilon = 20$.

a public unlabeled dataset, we compare it with our methods because DPSGD is the only baseline for multi-label prediction. DPSGD for multi-label was studied in prior work [48], and cannot directly leverage public unlabeled data. On the other hand, our PATE-based approaches leverage a public pool of unlabeled samples that have noisy labels provided by the ensemble of teachers. The added noise protects the privacy of the centralized training data. We then train the student on the newly labeled samples. Our Binary PATE does not use τ clipping or the confident GNMax improvement so as to fairly compare with Powerset PATE. Binary PATE has the benefit of using non-private learning on public data whereas DPSGD must add noise in training which impedes model learning. Finally, DPSGD incurs a high computational cost (which multi-label PATE does not) due to the expensive per-example gradient computations [42].

Pascal VOC. The student model, non-private baseline, and DPSGD baseline were all pre-trained on ResNet-50 models. Observing Table 1, we see that our Binary PATE algorithm outperforms all other privacy-preserving techniques by a significant margin. The non-private model achieves strong performance across all metrics where the model trained using DPSGD incurs significant degradation across all metrics.

Though Binary PATE outperforms DPSGD and Powerset PATE, it falls short of the non-private model by a wide margin, indicating much room for improvement in multi-label privacy-preserving

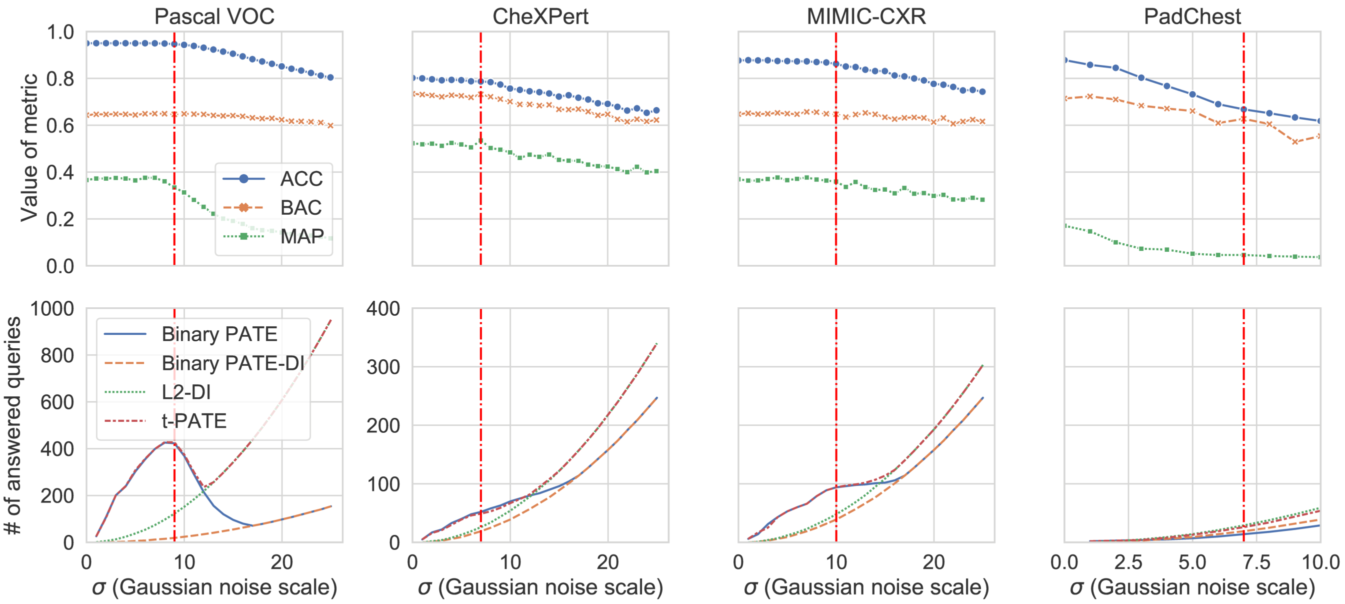

Figure 4: With sufficient consensus, the best query-utility tradeoff obtained lies in a regime of σ_G where the data-dependent bound is used. In the 1st row, we maximize the σ_G of τ PATE while maintaining sufficiently high values of the performance metrics. The chosen values for (σ_G, τ) are $9, 1.8$ for Pascal VOC, $10, 3$ for MIMIC-CXR, and $7, 2.8$ for CheXPert and $7, 2.7$ PadChest. When there is a lack of consensus (on PadChest), we see that the data-independent ℓ_2 (L2-DI) mechanism bound outperforms all others. For a well chosen τ, there is little-to-no impact on the consensus of the data-dependent regime (c.f. Binary PATE and τ PATE which leverage the data-dependent bound when it reduces privacy loss). Because of this, τ PATE achieves a competitive query-utility tradeoff. See Figure 13 in Appendix F.4 for tuning of Binary PATE as an ∞-Winner Election.

Table 1: DPSGD vs PATE on Pascal VOC for all 20 labels. Comparison between standard non-private model, DPSGD, Powerset and Binary multi-label in terms of utility using metrics: Accuracy (ACC), Balanced Accuracy (BAC), Area-Under-the-Curve (AUC), and Mean Average Precision (MAP).

Method	ACC	BAC	AUC	MAP
Non-private	.97	.85	.97	.85
DPSGD	.92	.50	.68	.40
Powerset PATE	.94	.58	.70	.29
Binary PATE	.94	.62	.85	.57

techniques—in particular, in extreme multi-label settings, which we motivate and expand on in Appendix B. We observe that Powerset PATE answers much fewer queries (leading to less training data for the student model) than Binary PATE, at only 78 compared to 427. Though the student model only trains on 427 samples compared to 5717 for DPSGD.

CheXpert. We follow the experimental setup provided for Adaptive DPSGD from [48]. We use the DenseNet-121 model pre-trained on ImageNet and fine-tune only the last fully connected layer while keeping all the other (convolutional) layers fixed. Across all of the experiments, we use $\varepsilon = 8$ as the privacy budget. We use the whole CheXpert test set, and here report results for $\delta = 1E - 4$. We set $\delta = 1E - 4$ only in this case to compare fairly with prior work [48]. In all other cases, we use the standard $\delta = 1/(\text{number of records})$, which is $\delta = 1E - 6$. To evaluate the performance of the models, we use the CheXpert test set (from the valid.csv file in CheXpert-v1.0-small). As shown in the Table 2, similarly to the results for Pascal VOC, we also observe a gap in performance (about 8%) between the non-private model and Binary PATE. For the direct comparison between private methods, in this setting, the Binary PATE also outperforms the state-of-the-art Adaptive DPSGD on four out of five labels, and in the average AUC across all the first 5 labels.

Table 2: DPSGD vs PATE on Chexpert for the first 5 labels. We compute the Area-Under-the-Curve (AUC) metric per label. We denote Adaptive DPSGD as Adaptive.

Method	AT	CA	CO	ED	EF	Average
Non-private	*0.84*	*0.80*	*0.87*	*0.90*	*0.91*	*0.87*
DPSGD	0.56	0.53	0.66	0.56	0.62	0.58
Adaptive	0.75	0.73	**0.84**	**0.79**	0.79	0.78
Binary PATE	**0.78**	**0.75**	**0.84**	0.76	**0.81**	**0.79**

6.5 Varying Privacy Budget ϵ

Because multi-label classification uses $k > 1$ labels, our work significantly outperforms the naive expectation of using k times higher privacy budget from single-label ($k = 1$) classification. We note that since we deal with multi-label scenarios and complicated vision and medical datasets, rather than the typical single-label classification tasks, it is more difficult to attain tighter DP guarantee. Our choice of $\varepsilon <= 20$ falls within the range that is generally considered in prior work [10, 36] and has been found to be robust to privacy attacks [11, 34]. We explore the privacy utility trade-off. The value

Table 3: Pascal VOC with 20 labels: Performance of Binary PATE for different values of the privacy budget ε w.r.t. number of answered queries, ACC, BAC, AUC, and MAP as measured on the test set with the specified $\sigma_{\textbf{GNMax}}$. PB (ε) is the privacy budget. We use 50 teacher models. We set $\sigma_{\textbf{GNMax}} = 7$.

PB (ε)	Queries answered	ACC	BAC	AUC	MAP
1	**0**	-	-	-	-
2	**6**	.86	.62	.62	.44
3	**13**	.93	.67	.67	.53
4	**22**	.93	.64	.64	.44
5	**31**	.95	.63	.63	.39
6	**40**	.95	.67	.67	.45
7	**64**	.95	.64	.64	.35
8	**81**	.95	.66	.66	.40
9	**101**	.95	.60	.60	.28
10	**113**	.96	.63	.63	.30
11	**135**	.96	.64	.64	.33
12	**165**	.96	.65	.65	.35
13	**199**	.96	.63	.63	.32
14	**217**	.96	.64	.64	.35
15	**239**	.96	.63	.63	.32
16	**272**	.96	.63	.63	.31
17	**306**	.96	.63	.63	.30
18	**332**	.96	.63	.63	.31
19	**362**	.96	.63	.63	.30
20	**403**	.96	.63	.63	.30

of ε can be decreased with fewer queries answered by teachers in PATE, which is shown in Table 3, where we vary the privacy budget from a tight guarantee ($\varepsilon = 1$) to looser guarantees (up to $\varepsilon = 20$).

6.6 Multi-Label CaPC

By replacing the single-label PATE in [10] with our multi-label PATE mechanisms, we enable multi-label learning in a multi-site setting: the framework of CaPC allows for distributed collaboration across models located at different sites. We scale the evaluation of multi-label CaPC learning to real-world datasets and models by providing a decentralized (independent) evaluation of each answering party, enabling large models. Our multi-label CaPC experiments replicate the setup of [10] using the source code provided.

We train 1 model per participant on separate distinct portions of the training set and then use multi-label CaPC to improve the performance of 3 participants. Details are in Appendix E. Observing Table 18, we see that multi-label CaPC consistently improves the BAC, with the greatest improvement of a considerable 5 percentage point increase on Pascal VOC and improvement across all other metrics. These improvements are echoed in the larger and more privacy-sensitive CheXpert and MIMIC-CXR datasets; however, we observe some performance degradation on PadChest, likely because of the degraded vote utility compared to the original training data. As one of the main applications for CaPC is the healthcare domain, the on-average performance improvements and associated privacy guarantees demonstrate the utility of multi-label CaPC in a realistic use case.

Inspecting Figure 5, we see that multi-label CaPC leads to significant improvements in low-sample, low-performance labels (e.g., labels EN and FR on CheXpert). Thus, poorer performing models and classes can gain from the noisy aggregation of more performant teacher models through multi-label CaPC. This has potential ramifications for fairness because our experiments consistently demonstrate that private multi-label CaPC can improve model performance on its poorer performing subpopulations—where [43] show that differentially private can hurt performance on these subpopulations. We reiterate that these subpopulations are common in many settings such as healthcare due to imbalanced labels, e.g., rarer diseases (see Figure 7 of Appendix C).

7 CONCLUSIONS

Single-label classification requires us to return one categorical output. Instead, multi-label classification is more complicated since we return $k > 1$ class-labels for each input. Naively, one could apply single-label classification repeatedly for each of the k labels. However, there are correlations between these k labels. We thus show that this is suboptimal. For example, there are fewer than $\tau = 3$ positive labels per query in our medical datasets. Intuitively, a teacher in PATE should only vote for up to τ positive labels in this setting.

We address the need for privacy in the multi-label setting with three new multi-label voting mechanisms. We show and prove that, while simple, our Binary voting cannot be outperformed without strong candidate correlations. When these correlations exist, we prove new data-independent bounds for our τ voting mechanism and theoretically analyze when Powerset voting performs better. We also compare all possible norms used for τ-clipping and analytically demonstrate that the ℓ_2 norm is optimal. These fundamental insights allow us to scale privacy-preserving ML to multi-label tasks. Using these three new mechanisms, we create multi-label PATE which outperforms DPSGD in the centralized private learning setting. We further enable multi-label learning in multi-site scenarios by creating multi-label CaPC. We test our methods on standard vision and medical datasets. Our results show new state-of-the-art for private learning in multi-label settings and demonstrate a need for further exploration to lessen the gap between private methods and non-private baselines.

ACKNOWLEDGMENTS

We would like to acknowledge our sponsors, who support our research with financial and in-kind contributions: CIFAR through the Canada CIFAR AI Chair program, DARPA through the GARD program, Intel, Meta, NFRF through an Exploration grant, and NSERC through the Discovery Grant and COHESA Strategic Alliance. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. We would like to thank members of the CleverHans Lab for their feedback.

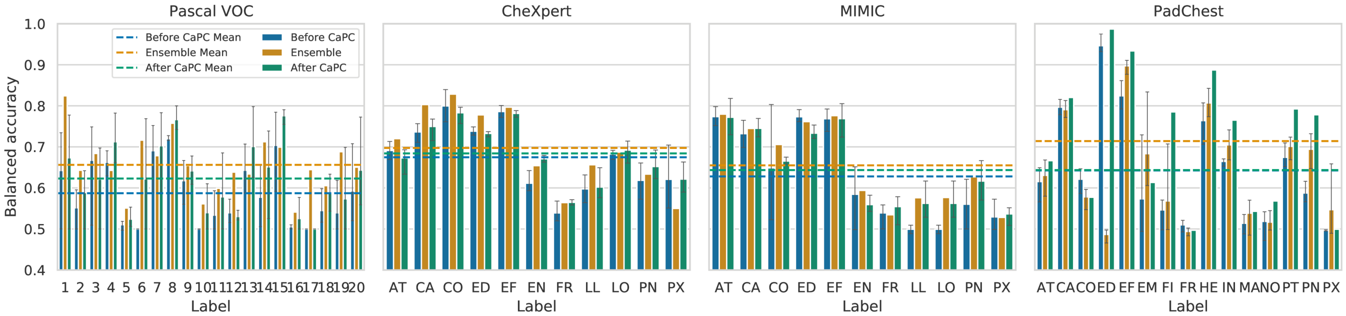

Figure 5: Using CaPC to improve model performance. ***Dashed lines represent mean balanced accuracy (BAC).*** **We retrain a given model using additional data labelled by all other models from the same ensemble. We observe a mean increase of BAC by 2.0 percentage points on CheXpert.**

Table 4: Model improvements through retraining with multi-label CaPC.

Dataset	# of Models	State	PB (ε)	ACC	BAC	AUC	mAP
Pascal VOC	1	Initial	-	.97	.85	.97	.85
	50	Before CaPC	-	.93±.02	.59±.01	.88±.01	.54±.01
	50	After CaPC	10	**.94±.01**	**.62±.01**	.88±.01	.54±.01
	50	After CaPC	20	**.94±.01**	**.64±.01**	**.89±.01**	**.55±.01**
CheXpert	1	Initial	-	.79	.78	.86	.72
	50	Before CaPC	-	.77±.06	.66±.02	.75±.02	.58±.02
	50	After CaPC	20	.76±.07	**.69±.01**	**.77±.01**	**.59±.01**
MIMIC	1	Initial	-	.90	.74	.84	.51
	50	Before CaPC	-	.84±.07	.63±.03	.78±.03	.43±.02
	50	After CaPC	20	**.85±.05**	**.64±.01**	**.79±.01**	**.45±.03**
PadChest	1	Initial	-	.86	.79	.90	.37
	10	Before CaPC	-	.90±.01	.64±.01	.79±.01	.16±.01
	10	After CaPC	20	.88±.01	.64±.01	.75±.01	.14±.01

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 308–318. https://doi.org/10.1145/2976749.2978318

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 308–318.

[3] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. 2013. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd international conference on World Wide Web*. 13–24.

[4] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Guha Thakurta. 2017. Practical Locally Private Heavy Hitters. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3d779cae2d46cf6a8a99a35ba4167977-Paper.pdf

[5] Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. 2021. Asymmetric loss for multi-label classification. In *ICCV*.

[6] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/1cd138d0499a68f4bb72bee04bbec2d7-Paper.pdf

[7] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. 2004. Learning multi-label scene classification. *Pattern recognition* 37, 9 (2004), 1757–1771.

[8] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. 2010. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*. IEEE, 3121–3124.

[9] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vayá. 2020. PadChest: A large chest x-ray image dataset with multi-label annotated reports. *Medical Image Analysis* 66 (Dec 2020), 101797. https://doi.org/10.1016/j.media.2020.101797

[10] Christopher A. Choquette-Choo, Natalie Dullerud, Adam Dziedzic, Yunxiang Zhang, Somesh Jha, Nicolas Papernot, and Xiao Wang. 2021. CaPC Learning: Confidential and Private Collaborative Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=h2EbJ4_wMVq

[11] Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-Only Membership Inference Attacks. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 1964–1974. https://proceedings.mlr.press/v139/choquette-choo21a.html

[12] Amrita Roy Chowdhury, Theodoros Rekatsinas, and Somesh Jha. 2020. Data-Dependent Differentially Private Parameter Learning for Directed Graphical Models. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 1939–1951. https://proceedings.mlr.press/v119/chowdhury20a.html

[13] Joseph Paul Cohen, Mohammad Hashir, Rupert Brooks, and Hadrien Bertrand. 2020. On the limits of cross-domain generalization in automated X-ray prediction. In *Proceedings of the Third Conference on Medical Imaging with Deep Learning (Proceedings of Machine Learning Research, Vol. 121)*, Tal Arbel, Ismail Ben Ayed, Marleen de Bruijne, Maxime Descoteaux, Herve Lombaert, and Christopher Pal (Eds.). PMLR, Montreal, QC, Canada, 136–155. http://proceedings.mlr.press/

v121/cohen20a.html

[14] Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. 2012. Differentially Private Spatial Decompositions. *2012 IEEE 28th International Conference on Data Engineering* (2012), 20–31.

[15] Yuval Dagan and Gil Kur. 2020. A bounded-noise mechanism for differential privacy. *arXiv preprint arXiv:2012.03817* (2020).

[16] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Miami Beach, Florida, USA.

[17] Emily Denton, Jason Weston, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. User conditional hashtag prediction for images. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1731–1740.

[18] Cynthia Dwork and Jing Lei. 2009. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 371–380.

[19] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. Springer, 265–284.

[20] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[21] Edith Elkind, Piotr Faliszewski, Jean-Francois Laslier, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. 2017. What Do Multiwinner Voting Rules Do? An Experiment over the Two-Dimensional Euclidean Domain. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (San Francisco, California, USA) *(AAAI'17)*. AAAI Press, 494–501.

[22] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88, 2 (2010), 303–338.

[23] Piotr Faliszewski, Arkadii M. Slinko, and Nimrod Talmon. 2017. Multiwinner Voting: A New Challenge for Social Choice Theory. In *Trends in computational social choice*.

[24] David J Hand and Robert J Till. 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine learning* 45, 2 (2001), 171–186.

[25] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, Nevada, USA.

[26] Otto Hölder. 1889. Ueber einen mittelwerthabsatz. *Nachrichten von der Königl. Gesellschaft der Wissenschaften und der Georg-Augusts-Universität zu Göttingen* 1889 (1889), 38–47.

[27] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[28] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn L. Ball, Katie S. Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. 2019. CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. *CoRR* abs/1901.07031 (2019). arXiv:1901.07031 http://arxiv.org/abs/1901.07031

[29] Alistair E. W. Johnson, Tom J. Pollard, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih ying Deng, Yifan Peng, Zhiyong Lu, Roger G. Mark, Seth J. Berkowitz, and Steven Horng. 2019. MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042* (2019).

[30] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. The composition theorem for differential privacy. In *International conference on machine learning*. PMLR, 1376–1385.

[31] Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. 2015. Consistent Multilabel Classification. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/85f007f8c50dd25f5a45fca73cad64bd-Paper.pdf

[32] Zelun Luo, Daniel J. Wu, Ehsan Adeli, and Li Fei-Fei. 2021. Scalable Differential Privacy With Sparse Network Finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5059–5068.

[33] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 263–275.

[34] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papemoti, and Nicholas Carlin. 2021. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 866–882.

[35] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[36] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

[37] Nicolas Papernot and Thomas Steinke. 2022. Hyperparameter Tuning with Renyi Differential Privacy. In *International Conference on Learning Representations*. https://openreview.net/forum?id=-70L8lpp9DF

[38] Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artieres, George Paliouras, Eric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Galinari. 2015. Lshtc: A benchmark for large-scale text classification. *arXiv preprint arXiv:1503.08581* (2015).

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[40] Hieu H Pham, Tung T Le, Dat Q Tran, Dat T Ngo, and Ha Q Nguyen. 2021. Interpreting chest X-rays via CNNs that exploit hierarchical disease dependencies and uncertainty labels. *Neurocomputing* 437 (2021), 186–194.

[41] Yanyao Shen, Hsiang-fu Yu, Sujay Sanghavi, and Inderjit Dhillon. 2020. Extreme Multi-label Classification from Aggregated Labels. In *International Conference on Machine Learning*. PMLR, 8752–8762.

[42] Pranav Subramani, Nicholas Vadivelu, and Gautam Kamath. 2020. Enabling fast differentially private sgd via just-in-time compilation and vectorization. *arXiv preprint arXiv:2010.09063* (2020).

[43] Vinith M Suriyakumar, Nicolas Papernot, Anna Goldenberg, and Marzyeh Ghassemi. 2021. Chasing Your Long Tails: Differentially Private Prediction in Health Care Settings. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 723–734.

[44] Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)* 3, 3 (2007), 1–13.

[45] Salil Vadhan. 2017. *The Complexity of Differential Privacy*. Springer International Publishing, Cham, 347–450. https://doi.org/10.1007/978-3-319-57048-8_7

[46] Stephan Van Eeden, Jonathon Leipsic, SF Paul Man, and Don D Sin. 2012. The relationship between lung inflammation and cardiovascular disease. *American journal of respiratory and critical care medicine* 186, 1 (2012), 11–16.

[47] Guoqiang Wu and Jun Zhu. 2020. Multi-label classification: do Hamming loss and subset accuracy really conflict with each other?. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 3130–3140. https://proceedings.neurips.cc/paper/2020/file/20479c788fb27378c2c99eadcf207e7f-Paper.pdf

[48] Xinyue Zhang, Jiahao Ding, Maoqiang Wu, Stephen T.C. Wong, Hien Van Nguyen, and Miao Pan. 2021. Adaptive Privacy Preserving Deep Learning Algorithms for Medical Data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 1169–1178.

[49] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene Parsing through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, Hawaii.

[50] Wennan Zhu, Peter Kairouz, Haicheng Sun, H. B. McMahan, and Wei Li. 2020. Federated Heavy Hitters Discovery with Differential Privacy. In *AISTATS*.

[51] Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. 2020. Private-kNN: Practical Differential Privacy for Computer Vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

A THEORETICAL MECHANISM ANALYSIS

Lemma A.1. *If $f(X) : \mathcal{R}^d \rightarrow \mathcal{R}^k$ is coordinate-independent, there exists a pair of databases (X, X') with $||X - X'||_1 = 1$ (a change in only one row) which achieves the worst-case sensitivity for each of the coordinates f_i.*

Proof. The proof follows by extension of the case where $d = k$ and $f_{i*}(X) = f_{i*}(X_{i*})$, i.e., each output coordinate is determined by a unique input coordinate. Thus, we can independently maximize the sensitivity for each f_{i*} and horizontally stack these to obtain a single X and X' that maximizes the sensitivity of f.

$$\begin{aligned}\Delta_1 f_{i*} = & \max_{\substack{(X,X') \\ ||X-X'||_1=1}} ||f_{i*}(X) - f_{i*}(X')||_1 \\ &= ||f_{i*}([X_{i|i\in P_{i*}}, X_{i|i\notin P_{i*}}]) - f_{i*}([X'_{i|i\in P_{i*}}, X'_{i|i\notin P_{i*}}])||_1 \\ &= ||f_{i*}([X_{i|i\in P_{i*}}, X_{i|i\notin P_{i*}}]) - f_{i*}([X'_{i|i\in P_{i*}}, X_{i|i\notin P_{i*}}])||_1\end{aligned}$$

Therefore we can choose the members in $P_{i*}(X)$ arbitrarily and independently to achieve the worst case sensitivity for f_{i*} while the members not in $P_{i*}(X)$ can take any value because they do not affect the output. Using the fact that the P_i's are disjoint, we can continue this over all values of $i*$ from 1 to k to get the desired result.

■

Proof of Proposition 3.1 From Lemma A.1, because $f = [f_1, \ldots, f_k]$, then the worst-case sensitivity is $\Delta f = (\sum_i^k (\Delta f_i)^p)^{1/p}$ by expansion of sensitivity from Definition 6. Similarly, we can construct an equivalent query by k applications of Binary voting which gives a sensitivity of $\Delta f = \sum_i^k (\Delta f_i)$. These two are equivalent for $p = 1$, i.e., the ℓ_1 norm. Binary voting is suboptimal for norms of $p > 1$ as a result of Hölder's inequality.■

Proposition A.2. *For a multi-label function $f(X) : \mathcal{R}^d \rightarrow \mathcal{R}^k$ to have a lower sensitivity than the sum of the worst-case sensitivities for each coordinate, i.e., $\Delta_1 f < \Sigma_{i=1}^k \Delta_1 f i$, f must be coordinate-dependent.*

Proof. Follows by contradiction since the function cannot be coordinate-independent. ■

Definition 9 (Mechanism for Powerset PATE). *Denote the Powerset operator as $\mathcal{P}(\cdot)$. For a sample x and 2^k subsets (classes), let $f_j(x) \in \{0,1\}^k$ denote the $j-th$ teacher model binary vector prediction. Let $n_i(x)$ be the vote count for the $i-th$ subset (class), i.e., $n_i(x) \triangleq |\{j : f_j(x) = P(\{0,1\}^k)_i\}|$. We define the Powerset PATE mechanism as*

$$\mathcal{M}_\sigma(x) \triangleq \arg\max_i \left\{ n_i(x) + \mathcal{N}(0, \sigma^2) \right\}.$$

B TOWARDS PRIVATE EXTREME MULTI-LABEL CLASSIFICATION

We face extreme multi-label classifications [41] in many real-world applications such as semantic segmentation [49], hash-tag suggestions for user images [17], product categorisation [3] and webpage annotation [38] where both input size and label size are extremely large. In this section, we investigate the privacy-accuracy tradeoffs of private semantic segmentation (a common and underlying example in extreme multi-label settings) that links each image pixel to its corresponding object class (an integer value) with a reasonable accuracy of above 60% but an expensive privacy cost of $\varepsilon \approx 3,000$, for a relative "small" image of size 200 × 200, or $\approx 40,000$ pixels. We conclude this section by proposing future directions to alleviate the privacy-accuracy tradeoffs in extreme multi-label settings.

We consider MIT ADE20K semantic segmentation dataset [49] that contains 150 objects including 35 stuff objects (e.g. sky, building) and 115 discrete objects (e.g. person, car). The label size for each image pixel is fixed (=150). However, the number of predicted labels for each image is the number of pixels, which varies across the dataset. To perform private semantic segmentation, we use PATE to label each image pixel. We split the training set of MIT ADE20K dataset into equally sized partitions for 20 teachers and train a Pyramid Pooling ResNet50-Dilated architecture of the Cascade Segmentation Module. The test accuracy of the ensemble of teachers (using 2000 of the test images) with respect to the PATE noise standard deviation σ_G between 0 and 5 varies from 67% to 47%. We observe that the level of noise must be quite small, $\sigma < 3$, or there is a steep drop in accuracy of more than 10 percentage points. The privacy cost is too high to provide meaningful guarantees ε, due to the small σ and large number of pixels required to be labeled.

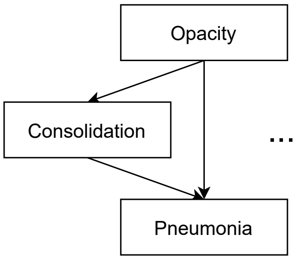

Figure 6: Example hierarchical structure of labels in Chest radiography setting.

We believe that privacy analysis in extreme multi-label settings can be tightened by exploiting the semantics of inputs. For example in the semantic segmentation task, we can reduce the privacy costs by taking advantage of the dependency between pixels so that instead of releasing an answer per pixel, we can release only a single label per semantic region (a grouping of pixels). Exploring label dependence, rather than assuming label independence, may also enable tighter privacy loss analysis and improve accuracy, as our analysis of Proposition 3.1 suggests. Label dependence is prevalent in many tasks, e.g., in healthcare labels are naturally organised into tree-like hierarchies such that domain experts (e.g. doctors) perform observations and diagnoses conditioned upon their parent node [46]. Figure 6 shows an example of the label structure where the root label node corresponds to the most generic disease of Opacity, while the leaf label node represents the most specific disease of Pneumonia [40]. Pneumonia implies the presence of both Consolidation and Opacity diseases. Thus, there exist many possible methods to optimize the answering of queries. It may be possible to tighten the privacy loss due to the implications (or, correlations) between labels; or, to query labels in a specific order such that the all dependent nodes (Consolidation and Opacity) can be inferred by the agreed presence of parent nodes (Pneumonia) by the teacher ensemble.

In addition to exploiting the knowledge of input and label domains, our analysis of the optimal settings for Binary PATE shows that privacy mechanisms can be tailored to the multi-label classifications. For example, k-fold adaptive bounds [30] that draw tighter (≪ sublinear) privacy bounds for homogenous privacy settings can be extended to heterogeneous ε per label and per query settings of multi-label classification. However, it is unclear if and under what scenarios we can achieve a tighter bound. For instance, it is possible to take the maximum ε across all queries, but if there is a large gap k-fold adaptive composition may yield looser bounds. These settings of coordinate dependence, high label correlations, and heterogeneous k-fold adaptive composition are interesting for future work.

We design an experiment where the baseline method obtains answers to all the labels while the new proposed method exploits the semantics and queries labels selectively.

First, we generate positive dependency matrices for labels in each dataset and present results in Tables 6,7,8, and 9. We find that the positive dependencies (e.g., if disease A is present then disease B is present as well) constitute a small fraction of the whole dataset. This is because there are many more negative than positive examples in the CheXpert dataset, which is caused by a class imbalance, a common problem in medical datasets. For instance, we find that if both Pneumonia and Pneumothorax are present then Lung Opacity occurs in 83.3% of the cases. However, both Pneumonia and Pneumothorax are present in only 0.06% of samples of the dataset. Thus, we consider negative instead of positive dependencies. For example, if Atelectasis is absent then Consolidation is absent as well in 98.7% of the cases. After ignoring samples for which at least one of Atelectasis or Consolidation have missing values, the percentage of samples where both labels are negative is 83%. We obtain the negative dependencies using the training set and generate the dependency matrix in Table 5.

We compare multi-label PATE executed for each label vs using the semantics and querying the first label (Atelectasis) only, followed by (1) skipping the remaining labels and setting them as negative if the first label is negative, or (2) querying the other labels if the first label is positive. As expected, leveraging the semantics increases the number of answered queries from 35 to 127 for the same privacy budget $\varepsilon = 8$ of at the cost of lower performance (less accurate answers to the queries). However, increasing the number of answered queries by adding more privacy noise ($\sigma = 67.5$) causes the answered queries to be less accurate than by exploiting the label dependencies. We show a detailed comparison in Table 10.

Note that in the above example we consider the first five labels from the CheXpert dataset. We use the same setup as for the comparison between DPSGD and multi-label PATE F.2. The metrics are computed on the same 127 queries (to obtain 127 answered queries for the *Answer all labels* we increase its privacy budget from 8 to 26.5).

C DATASETS AND MODEL ARCHITECTURES

We experiment on four multi-label datasets. First, we use the common computer vision multi-label dataset, Pascal VOC 2012. Three other of these datasets are privacy sensitive large-scale medical datasets that are commonly used in ML for healthcare: CheXpert [28], MIMIC-CXR [29] and PadChest [9]. These medical datasets present a realistic and large-scale application for multi-label CaPC.

Pascal VOC 2012 contains $11,540$ images that are split into $5,717$ images for training and $5,823$ images for validation [22]. There are 20 classes of object labels (with their index in parentheses) – aeroplane (1), bicycle (2), bird (3), boat (4), bottle (5), bus (6), car (7), cat (8), chair (9), cow (10), dining table (11), dog (12), horse (13), motorbike (14), person (15), potted plant (16), sheep (17), sofa (18), train (19), and tv monitor (20). We use a ResNet-50 model [25] that was pre-trained on ImageNet [16].

Medical Datasets contain chest radiographs (X-ray) images. CheXpert [28] has $224,316$ radiographs. MIMIC-CXR-JPG [29] contains $377,110$, and PadChest [9] has $160,868$. The goal in each dataset is to predict presence of pathologies. However, there are differences between pathology labels across these three datasets. X-ray images of CheXpert are annotated with 11 pathologies– Atelectasis, Cardiomegaly, Consolidation, Edema, Effusion, Enlarged Cardiomediastinum, Fracture, Lung Lesion, Lung Opacity, Pneumonia and Pneumothorax. MIMIC-CXR-JPG includes 11 pathologies– Enlarged Cardiomediastinum, Cardiomegaly, Lung Opacity, Lung Lesion, Edema, Consolidation, Pneumonia, Atelectasis, Pneumothorax, Pleural Effusion, Pleural Other, Fracture, Support Devices. PadChest includes 15 pathologies– Atelectasis, Cardiomegaly, Consolidation, Edema, Effusion, Emphysema, Fibrosis, Fracture, Hernia, Infiltration, Mass, Nodule, Pleural_Thickening, Pneumonia and Pneumothorax.

The pathology and its code (in parenthesis) is as follows: Atelectasis (AT), Cardiomegaly (CA), Consolidation (CO), Edema (ED), Effusion (EF), Emphysema (EM), Enlarged Cardiomediastinum (EN), Fibrosis (FI), Fracture (FR), Hernia (HE), Infiltration (IN), Lung Lesion (LL), Lung Opacity (LO), Mass (MA), Nodule (MO), Pleural_Thickening (PT), Pneumonia (PN), Pneumothorax (PX).

All datasets obtain labels from associated reports. Both CheXpert and MIMIC-CXR-JPG use the CheXpert labelling system, which is a rule based approach. PadChest obtains reports annotated by trained radiologists, then trains an attention-based recurrent neural network to predict on these annotations, and labels the remaining data. On all three datasets, we train a DenseNet-121 [27] model and filter only frontal images (AP and PA). The main difference between our setup and the one from [13] is that we use both frontal views *AP* and *PA*, while the cited work uses only one of the frontal views, thus either *AP* and *PA*.

The label distribution for each multi-label dataset (Pascal VOC, CheXpert, MIMIC-CXR, and PadChest) is presented in Figure 7. The label distribution for medical datasets is more unbalanced than for Pascal VOC, due to prevalence or rarity of certain diseases.

D EVALUATION METRICS

The *accuracy metric (ACC)* used refers to the average accuracy over all labels, i.e., the accuracy is measured for each label individually over the samples in the test set and then averaged to give an overall accuracy for the classifier. This corresponds to micro averaging in [31]. To put the above definition in words, the per label accuracy is the proportion of correct predictions (both true positives and true negatives) among the total number of cases examined. We use the function *sklearn.metrics.accuracy_score* from [39] in the code.

The *balanced accuracy (BAC)* is the macro-average of recall and true-negative-rate scores per class. In the binary case, balanced accuracy is equal to the arithmetic mean of sensitivity (true-positive-rate, recall) and specificity (true-negative-rate). We use the standard method *sklearn.metrics.balanced_accuracy_score* from [39] to compute the balanced accuracy per label and then average the scores across all labels. If the data set is balanced or the classifier performs

Table 5: Negative Dependency Matrix for the first 5 labels form the CheXpert dataset.

	Atelectasis	**Cardiomegaly**	**Consolidation**	**Edema**	**Effusion**
Atelectasis		*0.975*	*0.987*	*0.976*	*0.983*
Cardiomegaly	0.736		0.836	0.784	0.869
Consolidation	0.527	0.591		**0.631**	0.790
Edema	**0.625**	**0.665**	**0.758**		**0.822**
Effusion	0.485	0.567	0.731	0.633	

Table 6: Positive dependency (co-prevalence) matrix for labels in the Pascal-VOC dataset.

	aeroplane	**bicycle**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**	**diningtable**	**dog**	**horse**	**motorbike**	**person**	**pottedplant**	**sheep**	**sofa**	**train**	**tvmonitor**
aeroplane	1.000	0.000	0.001	0.007	0.000	0.004	0.061	0.000	0.001	0.000	0.000	0.000	0.000	0.001	0.133	0.003	0.000	0.000	0.000	0.000
bicycle	0.000	1.000	0.004	0.005	0.060	0.030	0.163	0.005	0.039	0.004	0.009	0.005	0.002	0.035	0.564	0.042	0.000	0.007	0.004	0.009
bird	0.001	0.003	1.000	0.012	0.004	0.001	0.005	0.003	0.008	0.004	0.001	0.006	0.004	0.000	0.052	0.008	0.006	0.000	0.000	0.001
boat	0.010	0.006	0.017	1.000	0.008	0.008	0.039	0.000	0.019	0.006	0.006	0.008	0.000	0.010	0.378	0.012	0.000	0.004	0.002	0.004
bottle	0.000	0.044	0.004	0.005	1.000	0.001	0.029	0.022	0.236	0.000	0.254	0.038	0.003	0.007	0.592	0.065	0.005	0.070	0.000	0.100
bus	0.007	0.040	0.002	0.009	0.002	1.000	0.384	0.000	0.000	0.000	0.000	0.002	0.002	0.026	0.477	0.009	0.002	0.000	0.007	0.000
car	0.033	0.076	0.003	0.016	0.018	0.134	1.000	0.005	0.020	0.007	0.006	0.030	0.018	0.089	0.445	0.020	0.004	0.002	0.018	0.005
cat	0.000	0.003	0.002	0.000	0.016	0.000	0.006	1.000	0.066	0.001	0.016	0.028	0.000	0.000	0.076	0.027	0.001	0.060	0.000	0.020
chair	0.001	0.017	0.005	0.008	0.139	0.000	0.019	0.055	1.000	0.000	0.335	0.065	0.002	0.005	0.428	0.159	0.000	0.156	0.002	0.152
cow	0.000	0.006	0.010	0.010	0.000	0.000	0.026	0.003	0.000	1.000	0.000	0.013	0.006	0.003	0.159	0.000	0.003	0.000	0.000	0.000
diningtable	0.000	0.008	0.002	0.005	0.304	0.000	0.011	0.027	0.679	0.000	1.000	0.017	0.000	0.000	0.551	0.159	0.000	0.090	0.000	0.051
dog	0.000	0.002	0.004	0.003	0.022	0.001	0.029	0.023	0.065	0.003	0.008	1.000	0.003	0.002	0.240	0.024	0.008	0.076	0.000	0.013
horse	0.000	0.002	0.006	0.000	0.004	0.002	0.046	0.000	0.006	0.004	0.000	0.008	1.000	0.002	0.472	0.004	0.004	0.000	0.002	0.002
motorbike	0.002	0.037	0.000	0.009	0.009	0.021	0.203	0.000	0.011	0.002	0.000	0.006	0.002	1.000	0.575	0.032	0.006	0.002	0.000	0.000
person	0.021	0.074	0.009	0.045	0.104	0.047	0.125	0.019	0.127	0.011	0.081	0.071	0.052	0.070	1.000	0.041	0.013	0.069	0.036	0.050
pottedplant	0.004	0.042	0.011	0.011	0.088	0.007	0.042	0.051	0.363	0.000	0.180	0.055	0.004	0.030	0.319	1.000	0.000	0.164	0.014	0.106
sheep	0.000	0.000	0.015	0.000	0.012	0.003	0.015	0.003	0.000	0.003	0.000	0.031	0.006	0.009	0.169	0.000	1.000	0.000	0.000	0.000
sofa	0.000	0.006	0.000	0.003	0.078	0.000	0.003	0.094	0.291	0.000	0.083	0.142	0.000	0.001	0.433	0.134	0.000	1.000	0.001	0.125
train	0.000	0.004	0.000	0.002	0.000	0.005	0.040	0.000	0.004	0.000	0.000	0.000	0.002	0.000	0.287	0.015	0.000	0.002	1.000	0.000
tvmonitor	0.000	0.008	0.002	0.003	0.129	0.000	0.010	0.037	0.331	0.000	0.055	0.029	0.002	0.000	0.365	0.101	0.000	0.146	0.000	1.000

Table 7: Positive dependency (co-prevalence) matrix for labels in the MIMIC-CXR dataset.

	Enlarged Cardiomediastinum	**Cardiomegaly**	**Lung Opacity**	**Lung Lesion**	**Edema**	**Consolidation**	**Pneumonia**	**Atelectasis**	**Pneumothorax**	**Pleural Effusion**	**Pleural Other**	**Fracture**	**Support Devices**
Enlarged Cardiomediastinum	1.000	0.254	0.353	0.056	0.173	0.078	0.076	0.349	0.093	0.391	0.017	0.032	0.527
Cardiomegaly	0.041	1.000	0.263	0.021	0.246	0.059	0.079	0.316	0.044	0.400	0.012	0.022	0.477
Lung Opacity	0.050	0.231	1.000	0.059	0.157	0.057	0.166	0.278	0.047	0.343	0.015	0.019	0.381
Lung Lesion	0.063	0.145	0.466	1.000	0.073	0.082	0.115	0.194	0.049	0.284	0.026	0.017	0.228
Edema	0.046	0.410	0.297	0.017	1.000	0.092	0.116	0.257	0.029	0.531	0.008	0.012	0.440
Consolidation	0.052	0.245	0.271	0.049	0.231	1.000	0.224	0.230	0.057	0.532	0.010	0.016	0.525
Pneumonia	0.033	0.212	0.505	0.044	0.186	0.143	1.000	0.222	0.018	0.307	0.012	0.012	0.285
Atelectasis	0.055	0.311	0.311	0.027	0.152	0.054	0.082	1.000	0.068	0.501	0.007	0.024	0.471
Pneumothorax	0.063	0.186	0.224	0.030	0.073	0.057	0.028	0.291	1.000	0.332	0.011	0.047	0.602
Pleural Effusion	0.052	0.333	0.325	0.034	0.266	0.106	0.096	0.423	0.066	1.000	0.010	0.021	0.481
Pleural Other	0.061	0.267	0.402	0.084	0.108	0.057	0.102	0.159	0.062	0.281	1.000	0.077	0.321
Fracture	0.050	0.211	0.213	0.024	0.073	0.038	0.044	0.239	0.109	0.243	0.033	1.000	0.304
Support Devices	0.066	0.372	0.338	0.025	0.206	0.098	0.083	0.373	0.112	0.450	0.011	0.024	1.000

Table 8: Positive dependency (co-prevalence) matrix for labels in the CheXpert dataset.

	Atelectasis	**Cardiomegaly**	**Consolidation**	**Edema**	**Enlarged Cardiomediastinum**	**Fracture**	**Lung Lesion**	**Lung Opacity**	**Pneumonia**	**Pneumothorax**	**Pleural Effusion**
Atelectasis	1.000	0.107	0.055	0.056	0.048	0.053	0.056	0.414	0.033	0.084	0.471
Cardiomegaly	0.118	1.000	0.049	0.192	0.101	0.039	0.050	0.354	0.030	0.023	0.354
Consolidation	0.118	0.095	1.000	0.047	0.042	0.024	0.100	0.384	0.128	0.036	0.426
Edema	0.105	0.327	0.041	1.000	0.042	0.019	0.042	0.431	0.045	0.024	0.483
Enlarged Cardiomediastinum	0.107	0.205	0.044	0.050	1.000	0.045	0.086	0.343	0.017	0.049	0.258
Fracture	0.119	0.079	0.026	0.023	0.046	1.000	0.068	0.284	0.014	0.054	0.193
Lung Lesion	0.085	0.068	0.071	0.033	0.058	0.045	1.000	0.533	0.063	0.054	0.323
Lung Opacity	0.136	0.106	0.059	0.076	0.051	0.041	0.116	1.000	0.072	0.062	0.384
Pneumonia	0.088	0.073	0.160	0.064	0.020	0.017	0.112	0.585	1.000	0.017	0.235
Pneumothorax	0.149	0.037	0.030	0.023	0.039	0.042	0.064	0.336	0.011	1.000	0.381
Pleural Effusion	0.186	0.127	0.079	0.102	0.046	0.034	0.085	0.463	0.035	0.085	1.000

equally well on either class, the balanced accuracy metric reduces to the conventional accuracy (i.e., the number of correct predictions divided by the total number of predictions).

The *Area-Under-the-Curve (AUC)* refers to the area under the receiver operating characteristic (ROC) curve. The ROC curve plots the TP rate against the FP rate. The AUC can be interpreted as the probability that the model ranks a random positive example more highly than a random negative example. We use the method *sklearn.metrics.roc_auc_score* from [39] to compute the AUC per label which we then average over all of the labels. The AUC metric is useful in that it does not depend on the classification threshold used.

The *Mean Average Precision metric (MAP)* is the mean of the average precision of the model over all classes. The average precision for each class is found as the area under the precision-recall curve. The full code showing the functions for MAP and AP can be found in our code at */datasets/deprecated/coco/helper_functions/helper_functions.py* which was originally written for [5].

Table 9: Positive dependency (co-prevalence) matrix for labels in the PadChest dataset.

	Atelectasis	Cardiomegaly	Consolidation	Edema	Effusion	Emphysema	Fibrosis	Fracture	Hernia	Infiltration	Mass	Nodule	Pleural Thickening	Pneumonia	Pneumothorax
Atelectasis	1.000	0.130	0.018	0.002	0.138	0.013	0.005	0.041	0.023	0.171	0.019	0.045	0.041	0.074	0.005
Cardiomegaly	0.062	1.000	0.010	0.011	0.077	0.002	0.013	0.035	0.031	0.129	0.007	0.028	0.036	0.029	0.001
Consolidation	0.071	0.081	1.000	0.005	0.227	0.013	0.009	0.032	0.009	1.000	0.018	0.065	0.018	0.452	0.004
Edema	0.036	0.462	0.025	1.000	0.416	0.000	0.010	0.015	0.010	0.736	0.015	0.030	0.041	0.137	0.005
Effusion	0.165	0.194	0.068	0.025	1.000	0.013	0.007	0.053	0.008	0.301	0.028	0.064	0.043	0.095	0.007
Emphysema	0.055	0.015	0.014	0.000	0.049	1.000	0.015	0.048	0.007	0.145	0.018	0.077	0.102	0.054	0.015
Fibrosis	0.030	0.159	0.013	0.003	0.037	0.021	1.000	0.037	0.039	0.569	0.009	0.043	0.085	0.036	0.003
Fracture	0.063	0.113	0.012	0.001	0.068	0.017	0.010	1.000	0.030	0.084	0.011	0.058	0.058	0.026	0.007
Hernia	0.062	0.174	0.006	0.001	0.018	0.004	0.018	0.053	1.000	0.073	0.063	0.043	0.049	0.014	0.001
Infiltration	0.088	0.140	0.129	0.019	0.130	0.017	0.049	0.028	0.014	1.000	0.012	0.067	0.043	0.292	0.004
Mass	0.093	0.071	0.022	0.004	0.115	0.020	0.007	0.035	0.115	0.117	1.000	0.125	0.030	0.071	0.015
Nodule	0.049	0.064	0.018	0.002	0.058	0.019	0.008	0.041	0.017	0.142	0.028	1.000	0.063	0.070	0.005
Pleural Thickening	0.051	0.095	0.006	0.003	0.045	0.029	0.018	0.048	0.023	0.104	0.008	0.072	1.000	0.035	0.003
Pneumonia	0.086	0.071	0.133	0.008	0.093	0.014	0.007	0.020	0.006	0.665	0.017	0.075	0.033	1.000	0.000
Pneumothorax	0.081	0.027	0.018	0.004	0.108	0.063	0.009	0.085	0.004	0.130	0.054	0.076	0.036	0.000	1.000

Table 10: Exploit label dependencies for the multi-label classification.

	# of queries answered	ACC	BAC	AUC	MAP
Answer all labels	*35*	*0.84*	*0.82*	*0.82*	*0.64*
Increse privacy noise	127	0.63	0.61	0.61	0.44
Exploit negative dependencies	127	0.68	0.72	0.72	0.48

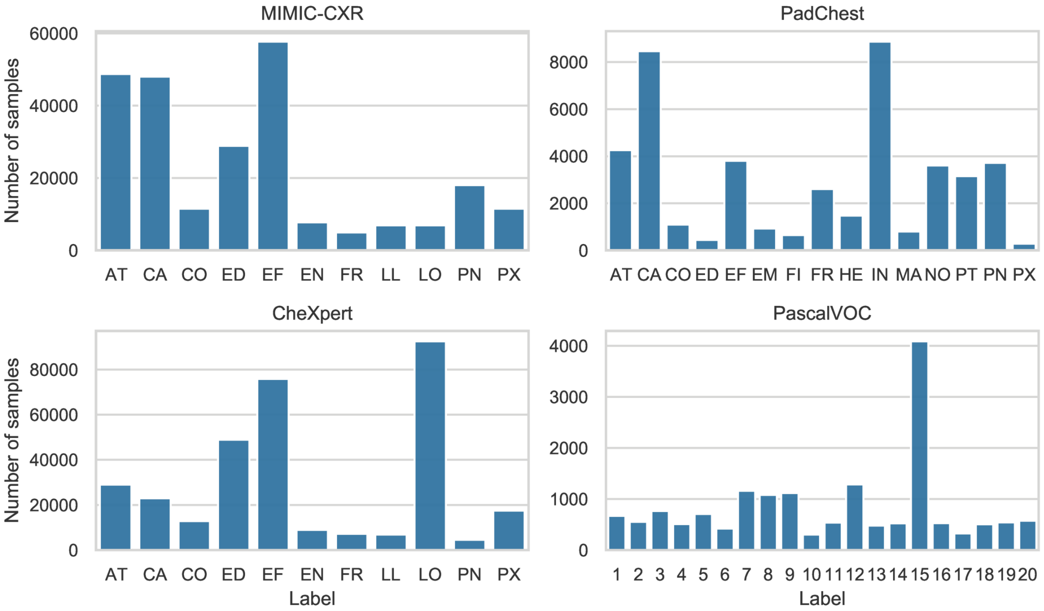

Figure 7: The label distribution for each multi-label dataset (Pascal VOC, CheXpert, MIMIC-CXR, and PadChest).

For a comprehensive evaluation of different algorithms, it is a common practice to test their performance on various measures and a better algorithm is the one that performs well on most of the measures [47].

In the code, we followed the evaluation procedure from [5] and also computed these metrics: true-positives, false-positives, false-negatives, true-negatives, average per-Class precision (CP), recall (CR), F1 (CF1), and the average Overall precision (OP), recall (OR) and F1 (OF1).

E EXPERIMENTAL DETAILS

Our experiments were performed on machines with Intel®Xeon®Silver 4210 processor, 128 GB of RAM, and four NVIDIA GeForce RTX 2080 graphics cards, running Ubuntu 18.04.

We train ResNet-50 [25] models on the Pascal VOC 2012 [22] dataset by minimizing multi-label soft margin loss function in 1000 epochs. As the optimiser, we use SGD with learning rate and weight decay of 0.001 and $1e-4$, respectively. We split the $5,717$ images of the training set to 50 private training sets in order to train 50 Pascal VOC teacher models.

We train the medical datasets CheXpert, MIMIC-CXR, and PadChest using the standard DenseNet-121 architecture trained for 100 epochs, with a weight decay of $1e-5$ and the Adam optimizer (learning rate=0.001) with the adam_amsgrad paremeter set. We reduce the learning rate when it plateaus, batch size is 64. The loss type is BCE (Binary Cross Entropy) with logits, and probability threshold is adjusted per label (where the standard value is 0.5). For 50 CheXpert models, each model is trained on 3750 private train samples. PadChest uses 9223 private samples per each of 10 teacher models.

Regarding the metrics, balanced accuracy (BAC), area under the curve (AUC), and mean average precision are common metrics used to asses performance of the multi-label classification [5]. Note that because we are in a sparse-hot multi-label setting (where a sample has many 0's - negative labels) the accuracy is not a preferred metric because a model can achieve high accuracy by returning all 0's; we include the accuracy metric for completeness.

For our CaPC experiments, we sample uniformly, without replacement, data points from the respective training data distribution to create disjoint partitions D_i of equal size for each party i. We use 50 total parties for Pascal VOC, CheXpert, and MIMIC-CXR, as well as 10 for PadChest; this choice depended on the sizes of the datasets and the performance of the models on the resulting partitioned data (here, we ensured no single model dropped below 60% BAC). We use $Q = 3$ querying parties and sample (without replacement) at least 1000 data points from the test distribution to form the unlabeled set for each querying party. We leave at least 1000 held-out data points for evaluating models. We first train party i's model on their private data D_i, then simulate multi-label CaPC learning by having each querying party complete the protocol with all other parties as answering parties. Using the new labelled data provided by multi-label CaPC, we retrain each querying party's model on their original data D_i plus the new labelled data. We average metrics over at least 3 runs with each using a different random seed.

F ADDITIONAL EXPERIMENTS

F.1 Powerset with τ-voting

The integration of τ-voting into the Powerset methods does not lower the sensitivity of the mechanism because changing one of the teachers potentially decreases the vote count from one class and increase it in another one. Thus, the sensitivity remains 2.

Intuitively, up to τ positive labels could be selected based on the confidence of the positive value for a label, where we would choose the top τ labels with the highest confidence of being positive. However, given that most deep multi-label models use independent predictive heads (i.e., a separate sigmoid activation per output label), naively comparing these values may not lead to the best performance. Thus, no clear notion of what the desired subset is in the case that $> \tau$ candidates are present.

For the Pascal VOC dataset, we set the same threshold of 0.5 probability per label, so the prediction heads for each label are aligned. For the CheXpert dataset, different probability thresholds are set per label, so the problem of deciding which of the labels should remain positive (if there are more positive labels than the max τ of positive labels) is difficult to resolve. This would likely require some form of domain knowledge.

For the original train set of the Pascal VOC dataset (with 5823 samples), the average number of positive labels per example is 1.52, with 20 total labels. More detailed statistics on the train set are presented in Table 13.

When the τ-voting is applied with the fixed privacy budget, we are able to answer more queries, while the performance metrics (e.g., accuracy) remain comparable. When the initial number of labels is set to $k = 5$, for $\tau = 2$ the number of classes is 16 instead of 32 ($\tau = 5$, which is equivalent to no clipping) and we are able to answer 5% more queries, for $\tau = 1$, we have only 6 classes and almost 14% more answered queries. For $\tau = 5$ there is only 1 more class than for $\tau = 4$ and we observe a very small difference in the number of answered queries (namely 8). The number of answered queries using the Binary method is still higher than with Powerset even when $\tau = 1$ (the number of classes is $C = k + 1$).

We present performance of Powerset PATE with τ-clipping w.r.t. number of answered queries in Table 15.

F.2 DPSGD vs PATE on CheXpert: Setup

The Adaptive DPSGD method from [48] does not publish the code, thus the following is our best effort and the results for Adaptive DPSGD come from Figure 5 in [48]. We use the following parameters to obtain our results for the Binary multi-label PATE:

```
sigma gnmax = 7.0
sigma threshold = 0
threshold = 0
method = multilabel
batch size = 20
learning rate = 0.001
epochs = 100
weight decay = 0
X-ray views = ['AP', 'PA']
Multilabel_prob_threshold = [
0.53, 0.5, 0.18, 0.56, 0.56]
```

F.2.1 CDF of Gaps. We show the Cumulative Distribution Function (CDF) for the gaps in Figure 8. We observe that the ensemble of teachers is confident about the answers for most labels for the Binary PATE and there are very small gaps for most queries when using Powerset PATE, which shows much lower confidence of teachers in choosing the same (super) classes, which are created from aggregated label predictions (the values of the labels are collected in a binary vector that constitutes a class).

F.2.2 Gaps and Performance Metrics. With more labels for the Powerset method, we have more possible classes and the votes become more spread out. This can be measured by the gap, which is the difference between the maximum number of votes per class and the runner-up (the number of votes for the next class with the highest number of votes). In Figures 9 and 10, we plot the average gap (on the y-axis) for the test set across all histograms for a given number of labels (presented on the x-axis). We compare the Powerset PATE (denoted as Powerset) vs the Binary PATE per label (denoted as Binary). The average gap between votes is comparable for different number of labels of Binary PATE since this

Table 11: Pascal VOC: Performance of Binary PATE vs Powerset PATE w.r.t. number of answered queries, ACC, BAC, AUC, and MAP as measured on the test set with the specified σ_{GNMax}. PB (ε) is the privacy budget. We use 50 teacher models. When we limit number of labels per dataset, we select the first k labels.

Dataset	Mode	# of Labels	Queries answered	PB (ε)	σ_{GNMax}	ACC	BAC	AUC	MAP
Pascal VOC	Binary	1	**5464**	20	2	.98	.84	.84	.75
Pascal VOC	Powerset	1	**5464**	20	2	.98	.84	.84	.75
Pascal VOC	Binary	2	**5442**	20	5	.97	.74	.74	.54
Pascal VOC	Powerset	2	**5464**	20	5	.97	.74	.74	.55
Pascal VOC	Binary	3	**3437**	20	7	.97	.72	.72	.51
Pascal VOC	Powerset	3	**3416**	20	7	.97	.72	.72	.51
Pascal VOC	Binary	5	**2398**	20	8	.96	.66	.66	.39
Pascal VOC	Powerset	5	**2040**	20	8	.96	.65	.65	.33
Pascal VOC	Binary	8	**1543**	20	7	.96	.69	.69	.46
Pascal VOC	Powerset	8	**1192**	20	7	.95	.68	.68	.37
Pascal VOC	Binary	11	**1190**	20	11	.95	.66	.66	.37
Pascal VOC	Powerset	11	**702**	20	4	.95	.66	.66	.37
Pascal VOC	Binary	14	**888**	20	12	.95	.65	.65	.36
Pascal VOC	Powerset	14	**417**	20	3	.95	.65	.65	.37
Pascal VOC	Binary	15	**465**	20	8	.94	.66	.66	.36
Pascal VOC	Powerset	15	**165**	20	3	.94	.66	.66	.36
Pascal VOC	Binary	16	**461**	20	7	.95	.65	.65	.38
Pascal VOC	Powerset	16	**94**	20	2	.95	.65	.65	.38
Pascal VOC	Binary	18	**446**	20	2	.95	.65	.65	.36
Pascal VOC	Powerset	18	**94**	20	2	.95	.65	.65	.36
Pascal VOC	Binary	20	**427**	20	7	.95	.65	.65	.37
Pascal VOC	Powerset	20	**78**	20	2	.95	.65	.65	.33

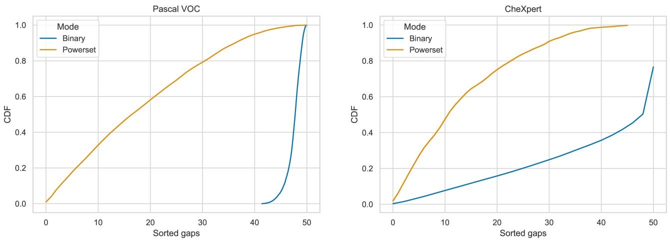

Figure 8: Binary vs Powerset PATE: CDF of gaps (differences between vote counts). We use 50 teacher models trained on the Pascal VOC and CheXpert datasets. These are raw gaps without adding any noise to the vote histograms. Most of the gaps are relatively large (> 40) for the Binary PATE per label, which shows that teachers are confident about the answers to queries. The average gap for the Powerset method is relatively low (only 18 for Pascal VOC and 13 for CheXpert).

method considers each label separately and there are always only two classes. The average gap between votes decreases very fast for Powerset because of the exponential growth of number of classes with more labels considered. Additionally, the performance metrics: acc (accuracy), bac (balanced accuracy), area under the curve (auc), and mean average precision (map), are also higher for the Binary than Powerset method in case of CheXpert dataset and comparable for Pascal VOC.

F.3 General Tuning of Hyper-Parameters

We set the hyper-parameters by considering the utility and privacy of our proposed method. Based on Lemma 3.2, we minimize the privacy budget by maximizing the σ parameter of the Gaussian noise and minimizing the τ parameters. In Figure 1, we tune the value of the σ parameter of the Gaussian noise based on the utility metrics: accuracy (ACC), balanced accuracy (BAC), mean average precision (MAP). We set σ as 9, 10, and 7 for Pascal VOC, MIMIC, and CheXpert datasets, respectively. This allows us to answer many queries with high performance in terms of ACC, BAC, and MAP.

Similarly, we tune the τ values for ℓ_2 norm in Figure 11 and for ℓ_1 norm in Figure 12.

Regarding the values of parameters σ_G, T, and σ_T, we follow the original work on PATE [35] and perform a grid search over a range of plausible values for each of the hyper-parameters.

The detailed analysis of tuning the hyper-parameters are for: τ-s in Section F.4, σ_G in Section F.5, PATE's thresholding (T, and σ_T) in Section F.6, and probability thresholds in Section F.8.

F.4 Tuning τ-Clipping

We determine the minimum value of τ for clipping in ℓ_2 and ℓ_1 norms for each dataset so that the performance as measured by accuracy, BAC, AUC, and MAP, is preserved. We present the analysis in Figures 11 and 12.

F.5 Tuning σ_{GNMax}

We determine how much of the privacy noise, expressed as the scale of Gaussian noise σ, can be added so that the performance

Table 12: CheXpert: Performance of Binary PATE vs Powerset PATE w.r.t. number of answered queries, ACC, BAC, AUC, and MAP (as measured on the test set with the specified σ_{GNMax}. PB (ε) is the privacy budget. We use 50 teacher models. When we limit number of labels per dataset, we select the first k labels.

Dataset	Mode	# of Labels	Queries answered	PB (ε)	σ_{GNMax}	ACC	BAC	AUC	MAP
CheXpert	Binary	1	**988**	20	8	.71	.71	.71	.63
CheXpert	Powerset	1	**988**	20	8	.71	.71	.71	.63
CheXpert	Binary	2	**898**	20	18	.73	.72	.72	.62
CheXpert	Powerset	2	**872**	20	12	.72	.72	.72	.60
CheXpert	Binary	3	**399**	20	13	.75	.77	.77	.55
CheXpert	Powerset	3	**674**	20	10	.74	.76	.76	.55
CheXpert	Binary	4	**554**	20	17	.74	.75	.75	.60
CheXpert	Powerset	4	**323**	20	5	.76	.78	.78	.60
CheXpert	Binary	5	**320**	20	17	.74	.76	.76	.59
CheXpert	Binary	5	**932**	20	29	.70	.70	.70	.54
CheXpert	Powerset	5	**582**	20	10	.74	.76	.76	.54
CheXpert	Binary	6	**299**	20	18	.74	.75	.75	.56
CheXpert	Powerset	6	**392**	20	7	.74	.75	.75	.55
CheXpert	Binary	7	**157**	20	12	.73	.74	.74	.51
CheXpert	Powerset	7	**338**	20	7	.73	.72	.72	.51
CheXpert	Binary	8	**145**	20	13	.73	.73	.73	.51
CheXpert	Powerset	8	**200**	20	5	.73	.71	.71	.50
CheXpert	Binary	9	**159**	20	16	.71	.70	.70	.50
CheXpert	Powerset	9	**241**	20	6	.71	.70	.70	.50
CheXpert	Binary	10	**105**	20	11	.74	.72	.72	.50
CheXpert	Powerset	10	**146**	20	4	.73	.71	.71	.50
CheXpert	Binary	11	**201**	20	5	.71	.68	.68	.45
CheXpert	Powerset	11	**152**	20	20	.71	.68	.68	.44

Table 13: Statistics about the positive labels per data point in the Pascal VOC dataset.

# of positive labels	# of train samples
6	1
5	17
4	105
3	484
2	1677
1	3539
0	0

Table 14: Parameter δ vs the number of answered queries. We show how the number of answered queries decreases gradually with orders of magnitude lower δ values for the PascalVOC dataset with $\varepsilon = 16, \sigma_{GNMax} = 9$.

δ value for (ε, δ)-DP	# of answered queries
10^{-5}	170
10^{-6}	158
10^{-7}	151
10^{-8}	139

as measured by accuracy, BAC, AUC, and MAP, is preserved. The experiment is run using the binary PATE per label and presented in Figure 13.

F.6 Tuning PATE

There are three parameters to tune: the differential privacy Gaussian noise standard deviation σ_G, the count threshold T, and the thresholding Gaussian noise standard deviation σ_T. We first disable the thresholding (σ_T and T) and tune σ_G to achieve a high BAC of the noisy ensemble while maximizing the number of answered queries. In Figure 18 we tune the σ_G parameter from PATE for different configurations of retraining. The goal is to maintain a high accuracy while selecting σ_G with high value so that as many queries as possible are answered. After tuning σ_G, we grid-search σ_T and T. Thresholding is useful if we want to add a relatively small amount of Gaussian noise (σ_G) in the noisy max. For example, for $\sigma_G = 7$ and 50 teacher models trained on CheXpert, the maximum number of labels answered without thresholding is 737 (67 queries) with a BAC above 0.69. With thresholding, e.g., at $T = 50$ and $\sigma_T = 30$, we can answer an average of 756 labels, up to 834. A well tuned threshold can also help reduce σ_G, which is the major influencing factor on the final BA of the noisy ensemble. Note that tuning the threshold benefits from a confident ensemble: here, we find the ensemble is often confident with a positive-negative vote difference of 40 of max 49. Also note that the BAC can be higher on easier queries.

F.7 Compare Different Methods

In Figure 14, we directly compare the Binary PATE (denoted as PATE), τ-PATE, and clipping in ℓ_2 and ℓ_1 norms. The error regions are for the three querying parties.

Table 15: Pascal VOC: Performance of Powerset PATE with τ-clipping w.r.t. number of answered queries with the specified σ_{GNMax}. The ACC, BAC, AUC, and MAP are measured on the answered queries from the test set. PB (ε) is the privacy budget. We use 50 teacher models. When we limit the number of labels per dataset, we select the first k labels. The τ denotes a maximum number of positive labels that a teacher is allowed to return per data sample.

# of Labels	τ	Queries answered	PB (ε)	σ_{GNMax}	ACC	BAC	AUC	MAP
1	1	**5464**	20	2	.98	.84	.84	.75
2	1	**5464**	20	5	.97	.75	.75	.55
2	2	**5464**	20	5	.97	.74	.74	.55
3	1	**3437**	20	7	.97	.71	.71	.48
3	2	**3421**	20	7	.97	.71	.71	.49
3	3	**3416**	20	7	.97	.72	.72	.51
4	1	**2547**	20	7	.97	.68	.68	.43
4	2	**2527**	20	7	.97	.69	.69	.43
4	3	**2485**	20	7	.97	.69	.69	.43
4	4	**2485**	20	7	.97	.69	.69	.43
5	1	**2321**	20	8	.96	.65	.65	.36
5	2	**2151**	20	8	.96	.65	.65	.34
5	3	**2077**	20	8	.96	.65	.65	.34
5	4	**2048**	20	8	.96	.65	.65	.33
5	5	**2040**	20	8	.96	.65	.65	.33
8	8	**1192**	20	7	.95	.68	.68	.37
10	1	**947**	20	5	.97	.65	.65	.36
10	2	**896**	20	5	.96	.65	.65	.36
10	3	**895**	20	5	.96	.65	.65	.33
10	4	**885**	20	5	.96	.65	.65	.33
10	5	**877**	20	5	.96	.65	.65	.33
10	6	**861**	20	5	.96	.64	.64	.30
10	7	**849**	20	5	.96	.64	.64	.29
10	8	**848**	20	5	.96	.66	.66	.34
10	9	**848**	20	5	.96	.64	.64	.32
10	10	**848**	20	5	.96	.64	.64	.30
11	11	**702**	20	4	.95	.66	.66	.37
14	14	**417**	20	3	.95	.65	.65	.37
15	15	**165**	20	3	.94	.66	.66	.36
16	16	**94**	20	2	.95	.65	.65	.38
18	18	**94**	20	2	.95	.65	.65	.36
20	1	**111**	20	2	.95	.63	.63	.33
20	2	**99**	20	2	.96	.67	.67	.41
20	3	**81**	20	2	.96	.61	.61	.32
20	4	**78**	20	2	.96	.63	.63	.33
20	5	**78**	20	2	.96	.63	.63	.34
20	10	**78**	20	2	.95	.65	.65	.34
20	20	**78**	20	2	.95	.65	.65	.33

In Table 17, we show the performance of the retrained models when using CaPC with various privacy budgets ε. In Figure 15 we show the detailed per label change in BAC for the retraining with privacy budget $\varepsilon = 10$. We observe an increase in average BA by around 0.03 after retraining when the privacy budget is set to $\varepsilon = 10$. We present detailed analysis of the Binary PATE performance in Table 3 on the Pascal VOC dataset when selecting the privacy budget ε in the range from 1 to 20. The number of answered queries is proportional to the square of the privacy budget.

We also compare the performance after re-training with and without the (confidence) thresholding mechanism used in PATE (represented by σ_T and threshold T parameters). We observe that in case of the medical datasets, there can be a slightly higher increase of the metrics (accuracy, BAC, AUC, MAP), when we do not perform the thresholding. For example, for the CheXpert dataset, when thresholding is not used, the improvement is higher by about one percentage point across all metrics when compared to the option with thresholding. For PadChest, such increase is for AUC and MAP, for BAC we see a drop by one percentage point, and there is no difference in terms of accuracy (remains at the level of about 0.86). This requires further investigation. Intuitively, we observe that the metrics can increase substantially with thresholding for the labels (pathologies) that are easy to classify. However, the most difficult to predict pathologies are left without answer and no improvement is made on them. On the other hand, without thresholding, we have to answer all labels and correct predictions on the hardest labels can produce substantial improvement in the classification of these hard labels.

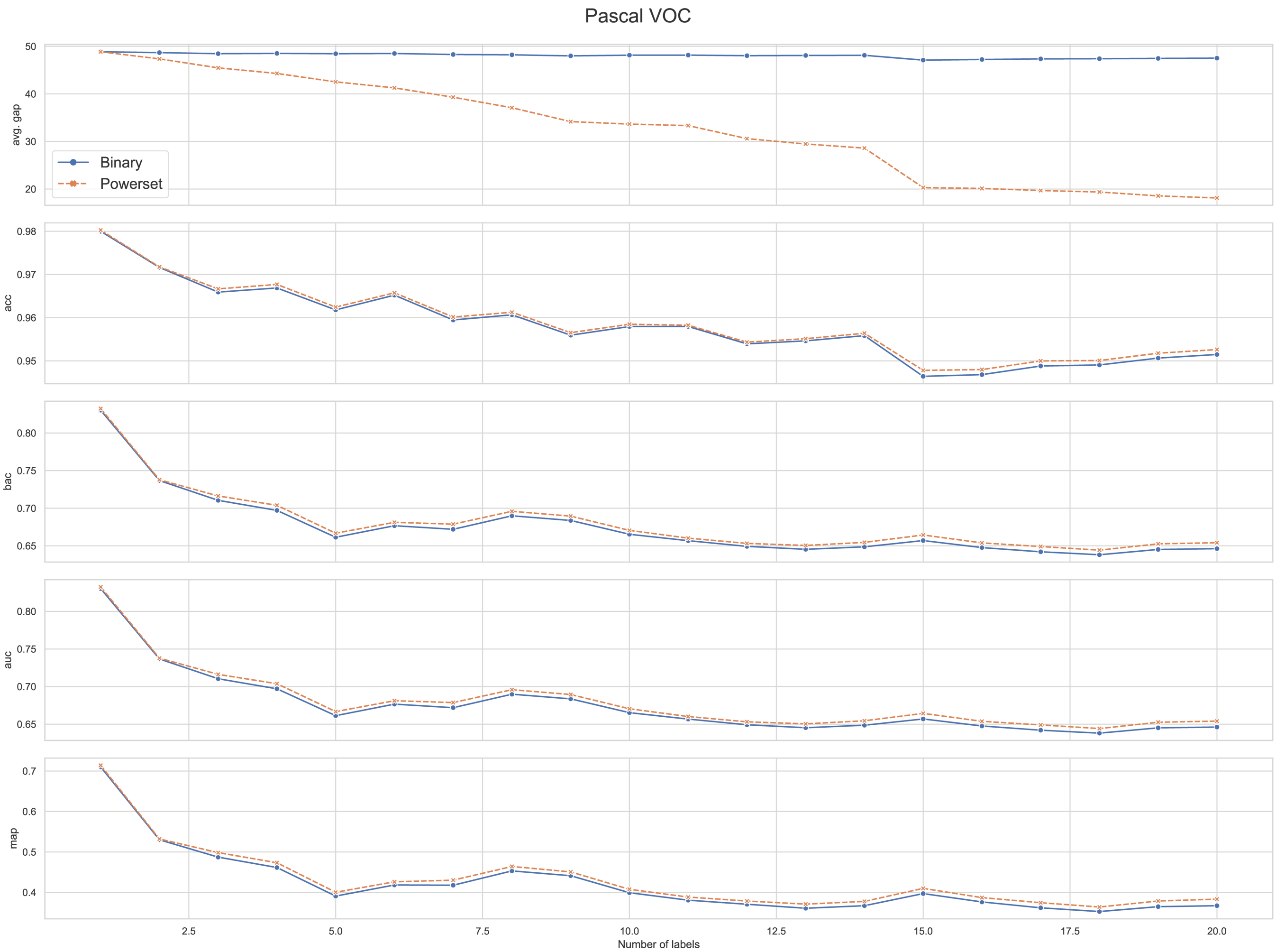

Figure 9: Binary vs Powerset PATE: gaps and metrics for Pascal VOC. We use 50 teacher models. These are raw gaps without adding any noise to the vote histograms.

F.8 Tuning Probability Threshold

We tune the global (applied to all labels) probability threshold γ that determines if a given probability denotes positive ($P > \gamma$), or a negative ($P \leq \gamma$) vote. The best global γ for the 50 models trained on the CheXpert dataset is 0.32 with the balanced accuracy of 0.702 and the μ value of 0.45, where μ is the average probability of predictions after applying the element-wise sigmoid function to the logits.

We also tune the γ threshold per label for CheXpert. We find the following probability thresholds per label using the validation set on the ensemble: 0.53, 0.5, 0.18, 0.56, 0.56, 0.21, 0.23, 0.46, 0.7, 0.2, 0.32. Next, we select the 3 teacher models with the lowest BAC and retrain them with CaPC. The average metrics BAC, AUC, and MAP before retraining are 0.63, 0.70, 0.53, and after retraining, we observe a significant improvement to 0.68, 0.75, 0.58, respectively. Thus, the value for each metric increases by around 0.05. We find the biggest performance improvement for these weakest models after learning from better teachers and retraining. We present detailed results per label and for BA as well as AUC metrics in Figure 17.

F.9 New Data Independent Bound

We also proved a new data-independent bound that achieves tighter guarantees (Lemma 3.2). The results for the method are displayed in Figure 4 under the label *L2-DI*, which denotes ℓ_2-norm τ clipping of the ballots. *L2-DI* can be compared with the Binary PATE data-dependent analysis, which is under the label Binary PATE. For the data sets like Pascal VOC, on which we achieve higher value of the metrics (Accuracy (ACC) 94%, BAC 64%, AUC 89%, MAP 55%), the data-dependent analysis allows us to release around 3.5x more queries (427 queries whereas the data-independent one only 123 queries, with all the parameters, such as amount of Gaussian noise added, being equal). On the other hand, for the data sets like MIMIC

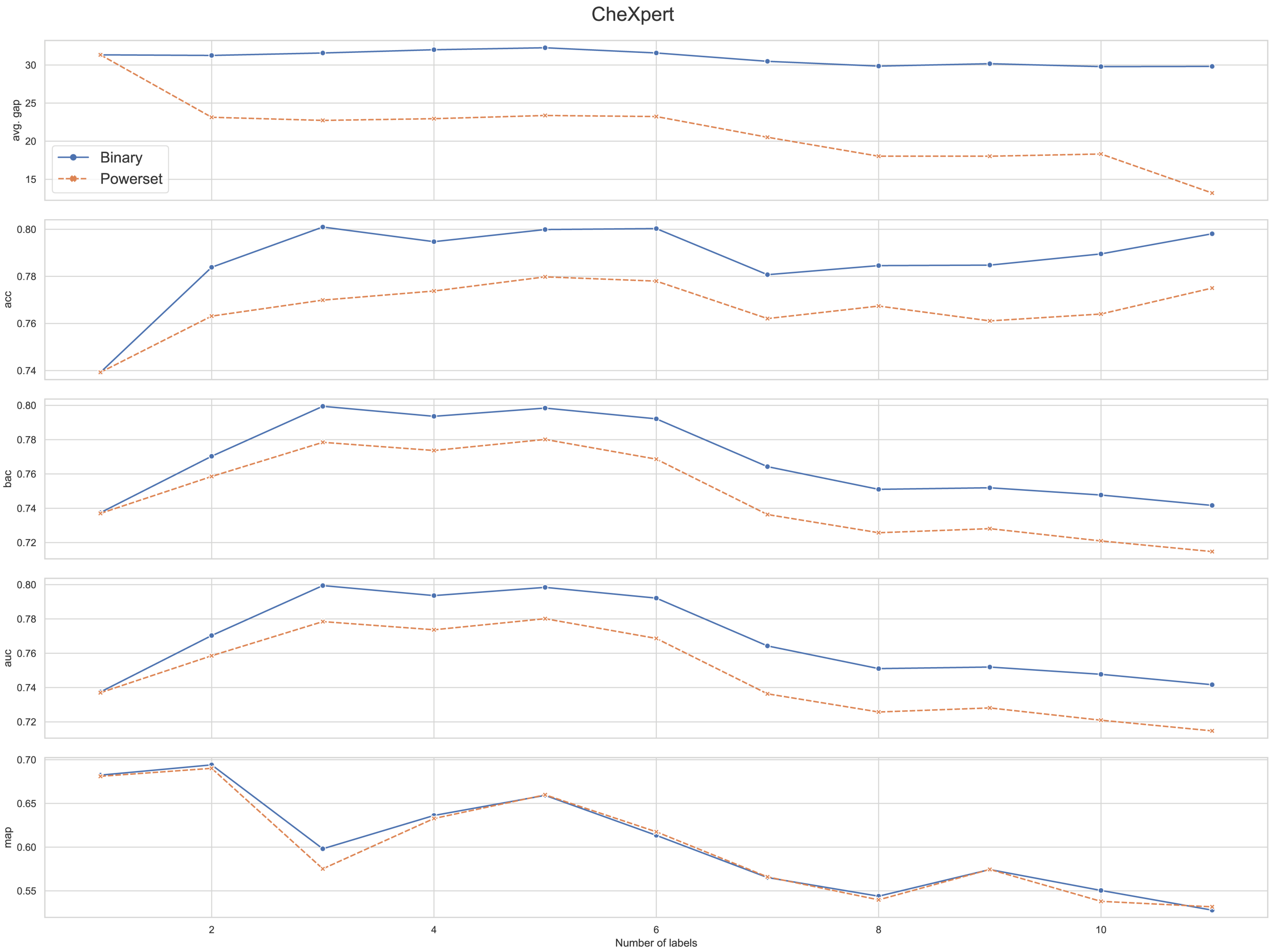

Figure 10: Binary vs Powerset PATE: gaps and metrics for CheXpert. We use 50 teacher models. These are raw gaps without adding any noise to the vote histograms.

with lower performance on the metrics (Accuracy (ACC) 85%, BAC 64%, AUC 79%, MAP 45%), the difference is less pronounced and the data-dependent analysis allows us to release around 1.9x more queries (94 queries whereas the data-independent one releases 48 queries).

F.10 Cross-Domain Retraining

In real-world healthcare scenarios, there are often rare diseases that may be difficult to accurately model with machine learning. Even coalitions of hospitals sharing similar data distributions of the same domain may not see benefits, due to poor aggregate performance. However, cross-domain collaboration through multi-label CaPC can help improve performance in these cases. The hospital annotation discrepancies may pose barriers: here, we simulate this by the different X-ray image labels between PadChest and CheXpert (see Supplement Section C).

To overcome this, we take the union of labels between the medical datasets and follow the experimental setup of [13]. We observe poor performing models on the PadChest dataset, with a low average performance of BAC $= 0.57$. Because of this, the benefits of multi-label CaPC within this coalition of hospitals are limited, since the ensemble is only marginally better. However, if this group of hospitals collaborated with another from a different but related domain, here represented by CheXpert, they may be able to see additional benefits. The models trained on this dataset achieve a higher BA (particularly on the first 5 shared pathologies as presented in Figure 16). Thus, the ensemble of all models engages in multi-label CaPC, and the models trained on CheXpert act as answering parties to provide labels for querying parties from PadChest. Using a $\sigma_G = 9$, $T = 50$, and $\sigma_T = 30$, we observe a higher BAC on all of those 5 pathologies, where we do not see a significant decrease on

Table 16: Performance of multi-label CaPC w.r.t. ACC, BAC, AUC, and MAP, on Pascal VOC (PA), CheXpert (CX), MIMIC (MC), and PadChest (PC). (-) denotes N/A. PB (ε) is the privacy budget.

Dataset	# of Models	State	PB (ε)	ACC	BAC	AUC	MAP
PA	1	Initial	-	.97	.85	.97	.85
	50	Before CaPC	-	.93±.02	.59±.01	.88±.01	.54±.01
	50	After CaPC	10	**.94±.01**	**.62±.01**	.88±.01	.54±.01
	50	After CaPC	20	**.94±.01**	**.64±.01**	**.89±.01**	**.55±.01**
CX	1	Initial	-	.79	.78	.86	.72
	50	Before CaPC	-	.77±.06	.66±.02	.75±.02	.58±.02
	50	After CaPC	20	.76±.07	**.69±.01**	**.77±.01**	**.59±.01**
MC	1	Initial	-	.90	.74	.84	.51
	50	Before CaPC	-	.84±.07	.63±.03	.78±.03	.43±.02
	50	After CaPC	20	**.85±.05**	**.64±.01**	**.79±.01**	**.45±.03**
PC	1	Initial	-	.86	.79	.90	.37
	10	Before CaPC	-	.90±.01	.64±.01	.79±.01	.16±.01
	10	After CaPC	20	.88±.01	.64±.01	.75±.01	.14±.01

Table 17: Performance of multi-label CaPC with τ-PATE w.r.t. ACC, BAC, AUC, and MAP, on Pascal VOC (PA), CheXpert (CX), MIMIC (MC), and PadChest (PC). (-) denotes N/A. PB (ε) is the privacy budget. T (Y/N) in the table refers to the PATE thresholding i.e. corresponding to the parameters T and σ_T being used (Y) or not used (N). Note that the probability thresholding per layer was used in these experiments (as described in Section F.8).

Dataset	# of Models	State	T(Y/N)	PB (ε)	ACC	BAC	AUC	MAP
PA	1	Initial	-	-	.88	.97	.91	
	50	Before CaPC	-	-	.93±.01	.59±.01	.89±.01	.54±.02
	50	After CaPC	N	20	**.94±.01**	**.60±.01**	.89±.01	.54±.02
CX	1	Initial	-	-	.79	.78	.86	.72
	50	Before CaPC	-	-	.75±.02	.69±.01	.77±.01	.59±.01
	50	After CaPC	Y	20	.73±.01	**.69±.01**	.76±.01	**.59±.01**
	50	After CaPC	N	20	.74±.01	**.70±.01**	**.77±.01**	**.59±.01**
MC	1	Initial	-	-	.90	.74	.84	.51
	50	Before CaPC	-	-	.84±.07	.63±.03	.78±.03	.43±.02
	50	After CaPC	Y	20	.84±.02	**.64±.04**	.77±.02	**.44±.01**
PC	1	Initial	-	-	.86	.79	.90	.37
	10	Before CaPC	-	-	.82±.01	.64±.01	.79±.01	.17±.01
	10	After CaPC	Y	20	**.86±.04**	.61±.01	.71±.03	.14±.02
	10	After CaPC	N	20	**.86±.02**	.60±.01	.72±.02	.15±.03

the other pathologies. Thus, multi-label CaPC provided benefits for the PadChest models in this cross-domain scenario.

G PREVIOUS ALGORITHM FOR THE MULTI-LABEL CLASSIFICATION

For completeness, we present the previously proposed algorithm 2 from [51].

H DATA-DEPENDENT PRIVACY ANALYSIS

Our binary voting mechanism does leverage both the smooth sensitivity and propose-test-release methods. For example, the Confident GNMax (proposed by [36]) is a form of the propose-test-release method described in Section 3.2 in [45].

First explored by [14], **differential privacy guarantees can be data-dependent**. These guarantees can lead to better utility with a long history, with several works using them [12, 14, 35, 36]. One main caveat is that the released epsilon score must now also be noised because it is itself a function of the data. [36] provide a way to do this via the smooth sensitivity (Section B in Appendix). Indeed our data-dependent differential privacy guarantees are formally proven and are based on the following intuition. Take the exponential mechanism which gives a uniform privacy guarantee. For instance, when the top score and the second score are very close, applying the exponential mechanism to this data there is a nearly uniform chance of picking either coordinate. However, for some inputs, the utility can be very strong—when the top score is much higher than the second score, then this mechanism is exponentially more likely to pick the top score than the second. This does not require local sensitivity of stability based methods, but is rather derived from the likelihood of picking either coordinate for this mechanism given the gap in the scores.

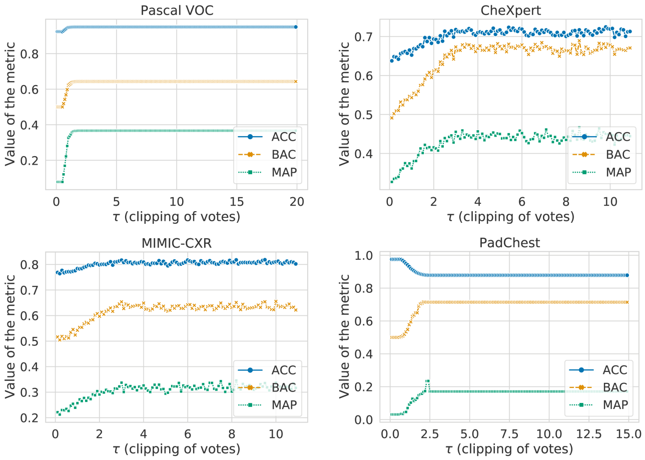

Figure 11: Value of the metric vs τ-clipping of votes in ℓ_2 norm. For a given τ, we plot accuracy (ACC), balanced accuracy (BAC), and mean average precision (MAP).

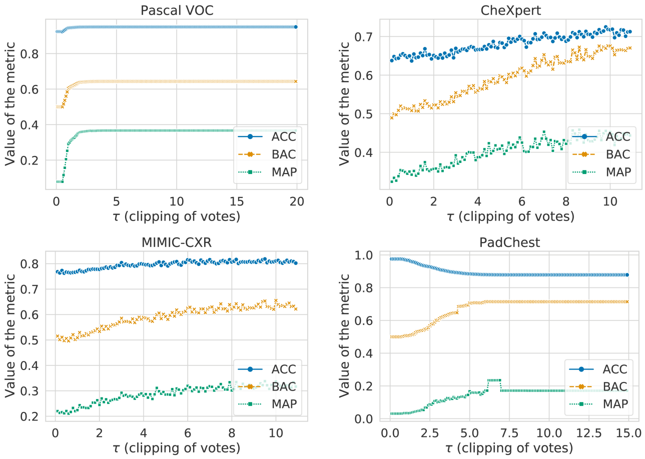

Figure 12: Value of the metric vs τ-clipping of votes in ℓ_1 norm. For a given τ, we plot accuracy (ACC), balanced accuracy (BAC), and mean average precision (MAP).

Algorithm 2 Multi-label classification with τ-clipping in ℓ_1-norm from Private kNN by [51].

```
Input: Data point $x$, clipping threshold $\tau_1$, Gaussian noise scale $\sigma_G$, $n$ teachers, each with model $f_j(x) \in \{0,1\}^k$, where $j \in [n]$.
Output: Aggregated vector $V \in \{0,1\}^k$ with $V_i = 1$ if returned label present, otherwise $V_i = 0$, where $i \in [k]$.
for all teachers $j \in [n]$ do
    $v_j \leftarrow \min(1, \frac{\tau_1}{\|f_j(x)\|_1}) f_j(x)$          ▷ $\tau$-clipping in $\ell_1$ norm
$V^1 = \sum_{j=1}^{n} v_j$          ▷ Number of positive votes per label
$V^0 = n - V^1$
$V^0 \leftarrow V^0 + \mathcal{N}(0, \sigma_G)$          ▷ Add Gaussian noise for privacy protection
$V^1 \leftarrow V^1 + \mathcal{N}(0, \sigma_G)$
for all labels $i \in [k]$ do
    if $V_i^1 > V_i^0$ then          ▷ Decide on the output vote
        $V_i = 1$
    else
        $V_i = 0$
```

I FROM STIRLING'S APPROXIMATION TO UPPER BOUND OF FACTORIAL

Stirling's approximation:

$$\sqrt{2\pi n}(n/e)^n \leq n! \leq e^{1/12n}\sqrt{2\pi n}(n/e)^n$$

Upper bound of factorial:

$$\begin{aligned}
\binom{n}{k} &= \frac{n!}{k!(n-k)!} \\
&\leq \frac{e^{1/12n}\sqrt{2\pi n}(n/e)^n}{\sqrt{2\pi k}(k/e)^k\sqrt{2\pi(n-k)}((n-k)/e)^{n-k}} \\
&= \frac{e^{1/12n}\sqrt{n}}{\sqrt{2\pi k(n-k)}}\left(\frac{n/e}{k/e}\right)^k\left(\frac{n/e}{(n-k)/e}\right)^{n-k} \\
&= \frac{e^{1/12n}\sqrt{n}}{\sqrt{2\pi k(n-k)}}\left(\frac{n}{k}\right)^k\left(\frac{n}{n-k}\right)^{n-k} \\
&\text{(Note: } \left(\frac{(n-k)+k}{n-k}\right)^{n-k} = \left(1+\frac{k}{n-k}\right)^{n-k} \leq e^k\text{)} \\
&\leq \frac{e^{1/12n}\sqrt{n}}{\sqrt{2\pi k(n-k)}}\left(\frac{n}{k}\right)^k e^k \\
&\leq \frac{e^{1/12n}\sqrt{n}}{\sqrt{2\pi(n-1)}}\left(\frac{en}{k}\right)^k \\
&\leq \left(\frac{en}{k}\right)^k
\end{aligned}$$

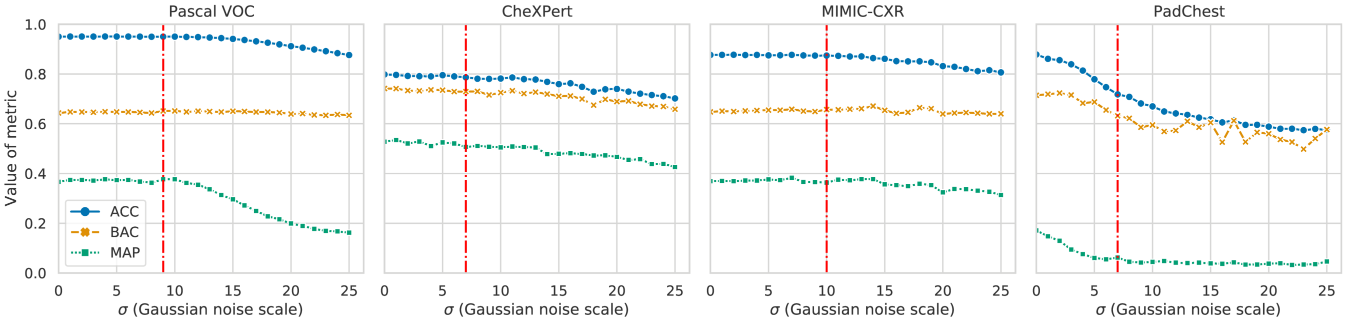

Figure 13: Value of the metric versus noise standard deviation σ. For a given σ, we plot accuracy (ACC), balanced accuracy (BAC), and mean average precision (MAP).

Table 18: Model improvements through retraining with multi-label CaPC. DD - denotes Data Dependent. We also present the results with sanitized epsilon values.

Dataset	# of Models	State	DD ε	Sanitized ε	ACC	BAC	AUC	MAP
Pascal VOC	1	Initial	-	-	.97	.85	.97	.85
	50	Before CaPC	-		.93±.02	.59±.01	.88±.01	.54±.01
	50	After CaPC	10	12.97	**.94±.01**	**.62±.01**	.88±.01	.54±.01
	50	After CaPC	20	26.00	**.94±.01**	**.64±.01**	**.89±.01**	**.55±.01**
CheXpert	1	Initial	-		.79	.78	.86	.72
	50	Before CaPC	-	-	.77±.06	.66±.02	.75±.02	.58±.02
	50	After CaPC	20	25.80	.76±.07	**.69±.01**	**.77±.01**	**.59±.01**
MIMIC	1	Initial	-	-	.90	.74	.84	.51
	50	Before CaPC	-	-	.84±.07	.63±.03	.78±.03	.43±.02
	50	After CaPC	20	25.80	**.85±.05**	**.64±.01**	**.79±.01**	**.45±.03**
PadChest	1	Initial	-	-	.86	.79	.90	.37
	10	Before CaPC	-	-	.90±.01	.64±.01	.79±.01	.16±.01
	10	After CaPC	20	25.80	.88±.01	.64±.01	.75±.01	.14±.01

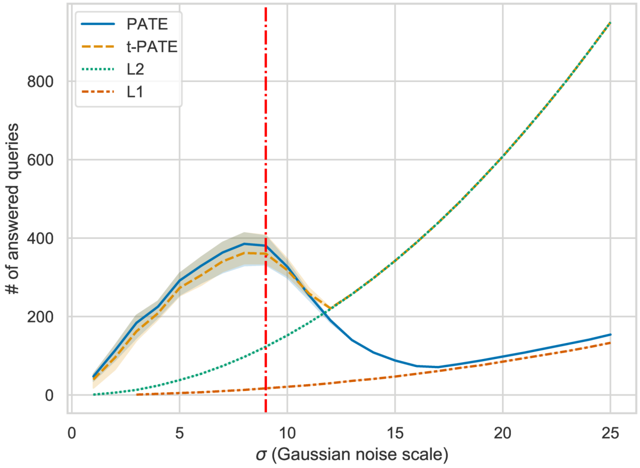

Figure 14: Compare methods: number of answered queries vs σ - using the Pascal VOC dataset, with $\tau_1 = 3.4$ (set for ℓ_1−norm clipping, and $\tau_2 = 1.8$ set for τ-PATE and ℓ_2−norm clipping. The red vertical line denotes the selected value of σ.

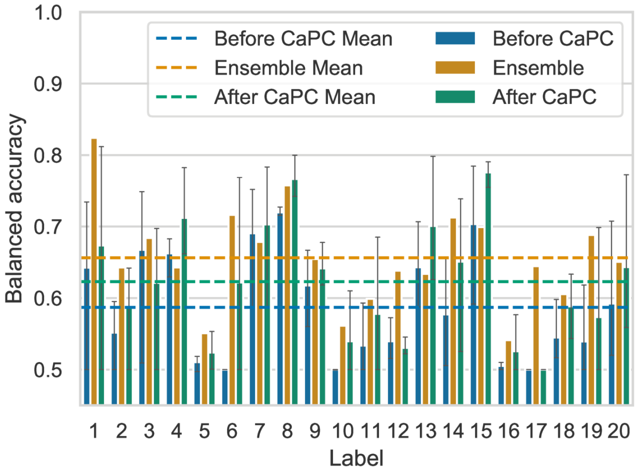

Figure 15: Retraining with privacy budget $\varepsilon = 10$ for the Pascal VOC dataset.

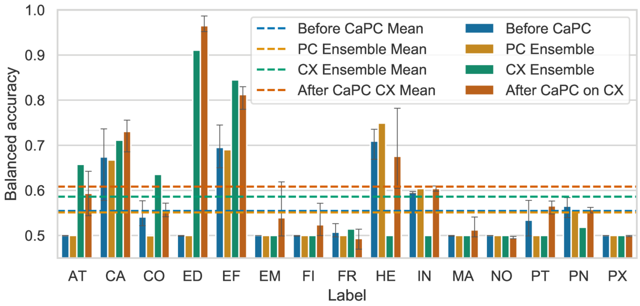

Figure 16: Cross-domain retraining with CaPC. We train 10 models on PadChest (PC) and compare their performance on PadChest test set against the ensemble of these models (PC Ensemble), the ensemble of 50 CheXpert models (CX Ensemble), and finally retrain the 10 PadChest models via binary multi-label PATE using the CheXpert ensemble (after CaPC on CX).

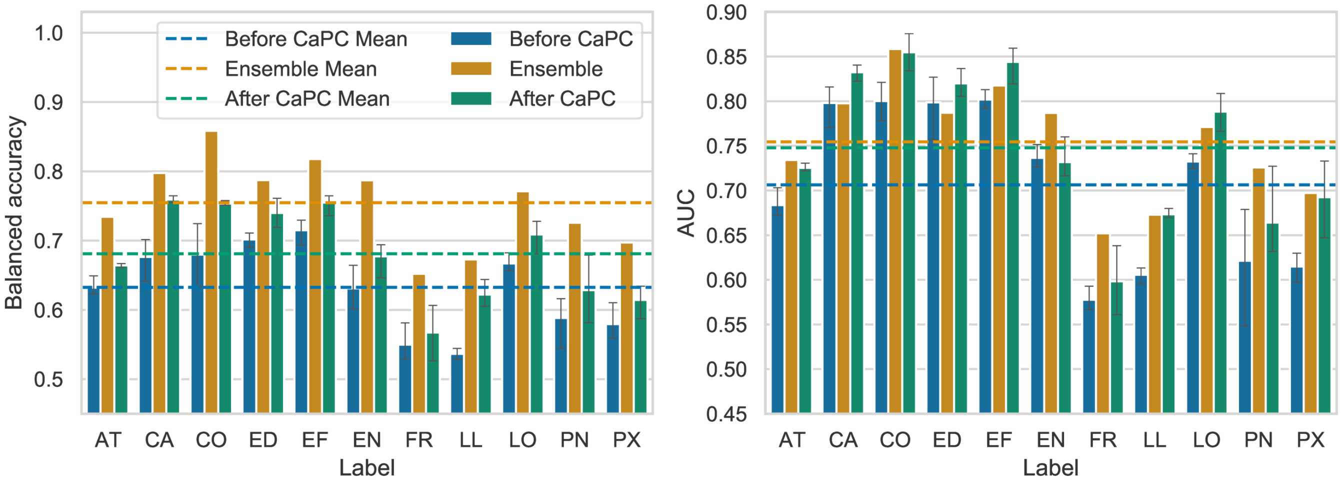

Figure 17: Using CaPC to improve the weakest models. ***Dashed lines represent mean values of the metrics: Balanced Accuracy (BAC) and AUC.*** **We retrain a given model using additional CheXpert data labelled by the ensemble of all the other models trained on CheXpert. All metrics are improved after retraining by around** 0.05 **on average.**

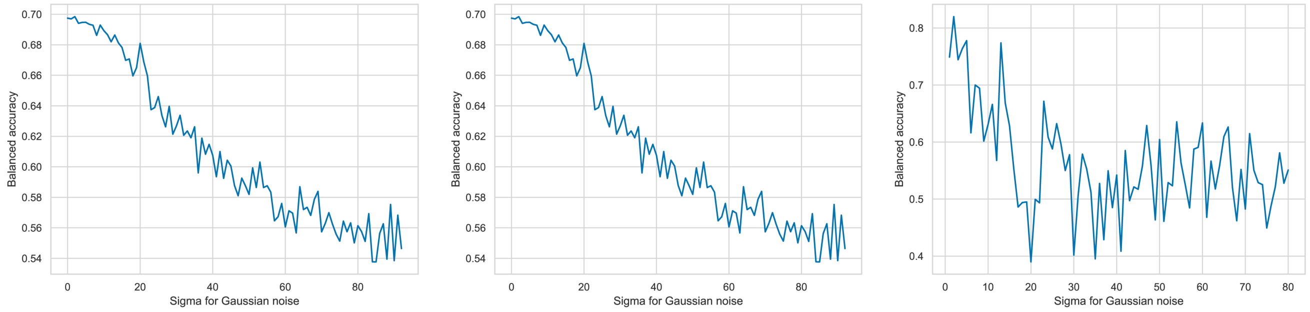

Figure 18: The analysis of the balanced accuracy (y-axis) of votes from the ensemble as we increase the (sigma of the Gaussian) noise (x-axis) from the binary multi-label PATE. We search for variance of the noise σ_G **that can preserve high BAC for the ensemble. Left: 50 CheXpert models with train and test on CheXpert,** $\sigma_G \leq 9.0$ **preserves more than 0.69 of BAC. Middle: 50 CheXpert models with train on CheXpert and test on PadChest test set,** $\sigma_G \leq 9.0$ **preserves more than 0.75 of BAC. Right: 10 PadChest models with train and test on PadChest,** $\sigma_G \leq 5.0$ **preserves more than 0.75 of BAC.**