A Differential Privacy

An algorithm is said to be differentially private if its outputs on adjacent inputs (in our case, datasets) are statistically indistinguishable. Informally, the framework of differential privacy requires that the probabilities of an algorithm making specific outputs be indistinguishible on two adjacent input datasets. Two datasets are said to be adjacent if they only differ by at most one training record. The degree of indistinguishibility is bounded by a parameter denoted ε. The lower ε is, the stronger the privacy guarantee is for the algorithm because it is harder for an adversary to distinguish adjacent datasets given access to the algorithm's predictions on these datasets. In the variant of differential privacy we use, we can also tolerate that the guarantee not hold with probability δ. This allows us to achieve higher utility.

B Shifting Distributions

In Section 3, we explain that we shift our histogram estimate by a constant to account for the number of teachers known to the attacker. The following theorem shows that the number of teachers does not affect the attacker's computation

Theorem 2. *For two histograms,* $H^1 = [h^1_1, \ldots, h^1_m]$ *and* $H^2 = [h^2_1, \ldots, h^2_m]$, $Q^{H^1,\sigma} = Q^{H^2,\sigma}$ *if* $h^1_i - h^2_i = h^1_j - h^2_j$ *for all* $i, j = 1, \ldots, m$.

Proof. let $d = h^1_i - h^2_i = h^1_j - h^2_j$ for all $i, j = 1, \ldots, m$.

$$\begin{aligned}\mathbb{P}(g^2_i > g^2_j) &\sim \mathcal{N}((h^2_i - h^2_j), 2\sigma^2)\\ &= \mathcal{N}((h^1_i + d) - (h^1_j + d), 2\sigma^2)\\ &= \mathcal{N}(h^1_i - h^2_j, 2\sigma^2)\\ &= \mathbb{P}(g^1_i > g^1_j)\end{aligned}$$

for all $i, j = 1, \ldots, m$.

$$\begin{aligned}Q^{H^1,\sigma}_k &= \mathbb{P}([g^1_k > g^1_1, \ldots, g^1_k > g^1_{k-1},\\ &g^1_k > g^1_{k+1}, \ldots, g^1_k > g^1_m])\\ &= \mathbb{P}([g^2_k > g^2_1, \ldots, g^2_k > g^2_{k-1},\\ &g^2_k > g^2_{k+1}, \ldots, g^2_k > g^2_m])\\ &= Q^{H^2,\sigma}_k\end{aligned}$$

□

What Theorem 2 states is, if the difference between two histograms is uniform, then the probability distribution of the outcomes is the same. With the support of Theorem 2, H can be safely shifted by a constant amount to sums up to the number of teachers, N.

C Chosen histograms for evaluation

Table 1 shows the histograms we chose for evaluation in the 3 consensus-level categories.

	MNIST	SVHN
High consensus		
H1	[4, 7, 6, 8, 4, 2, 0, 214, 4, 1]	[0, 0, 0, 0, 250, 0, 0, 0, 0, 0]
H2	[4, 7, 207, 10, 4, 4, 0, 10, 3, 1]	[0, 0, 250, 0, 0, 0, 0, 0, 0, 0]
H3	[5, 205, 7, 8, 4, 3, 0, 11, 6, 1]	[0, 0, 0, 250, 0, 0, 0, 0, 0, 0]
H4	[4, 7, 6, 7, 4, 200, 4, 10, 7, 1]	[0, 250, 0, 0, 0, 0, 0, 0, 0, 0]
H5	[4, 7, 210, 7, 4, 4, 0, 10, 3, 1]	[0, 0, 0, 0, 0, 0, 250, 0, 0, 0]
Median consensus		
H1	[5, 183, 9, 16, 4, 3, 1, 10, 17, 2]	[0, 0, 1, 0, 249, 0, 0, 0, 0, 0]
H2	[6, 7, 6, 30, 4, 181, 0, 10, 5, 1]	[0, 10, 1, 232, 1, 3, 0, 1, 0, 2]
H3	[4, 7, 6, 10, 13, 4, 0, 17, 3, 186]	[0, 0, 0, 6, 0, 243, 0, 0, 0, 1]
H4	[6, 18, 184, 7, 10, 4, 7, 10, 3, 1]	[236, 0, 0, 7, 0, 0, 6, 0, 1, 0]
H5	[7, 7, 8, 7, 4, 9, 193, 10, 4, 1]	[234, 2, 0, 4, 0, 0, 0, 1, 9, 0]
Low consensus		
H1	[12, 7, 6, 30, 4, 161, 0, 10, 19, 1]	[1, 1, 20, 12, 0, 0, 2, 207, 7, 0]
H2	[4, 8, 7, 11, 38, 16, 1, 13, 8, 144]	[0, 158, 1, 6, 4, 38, 0, 40, 1, 2]
H3	[4, 7, 15, 33, 6, 5, 0, 171, 5, 4]	[0, 184, 0, 2, 3, 0, 0, 61, 0, 0]
H4	[4, 7, 117, 99, 4, 4, 0, 10, 4, 1]	[0, 0, 24, 0, 0, 0, 0, 0, 0, 226]
H5	[4, 17, 6, 11, 154, 4, 0, 11, 5, 38]	[10, 1, 2, 19, 7, 109, 73, 0, 19, 10]

Table 1: The 30 MNIST and SVHN vote histograms sampled from the collection of histograms provided by Papernot et al [1] (divided into 3 equally-sized consensus groups). We refer to histograms denoted here by H1-5 in the different consensus groups throughout the presentation of our results.

D Fittig Random Forests

Every one of our teachers in Section 2 fits a random forest classifier using the sklearn package; each teacher performed a grid search over the following hyperparameters, and picked the values that lead to the lowest training loss.

- max_depth : the maximum number of levels that a tree has, an integer chosen between 1 and 11 inclusively;
- max_features : the maximum number of features, while splitting a node, one of sqrt(number of features), log(number of features), 0.1*(number of features), 0.2*(number of features), 0.3*(number of features), 0.4*(number of features), 0.5*(number of features), 0.6*(number of features), 0.7*(number of features), 0.8*(number of features), 0.9*(number of features);
- n_estimators : the number of trees that the forest has, an integer chosen between log(9.5) and log(300.5);
- criterion: the loss function, one of gini impurity and entropy;
- min_samples_split : the minimum number of instance for a node to split, one of 2, 5, 10;
- bootstrap: one of True or False

E End-to-end sensitive-attribute inference

In Section 2, we showed that histograms leak by mounting an attack that classifies histograms to low-consensus and high-consensus groups, which reveals information about minority-group membership. In Section 4, we showed that we can extract histograms by querying PATE instances. Now, we combine these two attacks, to extract minority-group membership information directly from a PATE instance. Our setting mirrors the setting from Section 2, but the attacker does not have direct access to histograms of individuals, and instead they extract them from PATE's answers using our methodology (Section 3). We used the same ensemble from Section 2, but this time, the 250 teachers' vote histogram was noised, again using $\sigma = 40$, $\delta = 0.00001$ and a privacy budget of 1.9

as in [2]. We sampled 10 low-consensus and 10 high-consensus members of the test set, and ran the attack on them: we queried PATE with each member's data record until exhausting the privacy budget, computed the Monte Carlo estimators, ran the optimization to recover the vote histogram, and then classified it to low-consensus/high-consensus as in Section 2. Results are given in Figure 8, and indeed, they mirror the results of the attack in Section 2.

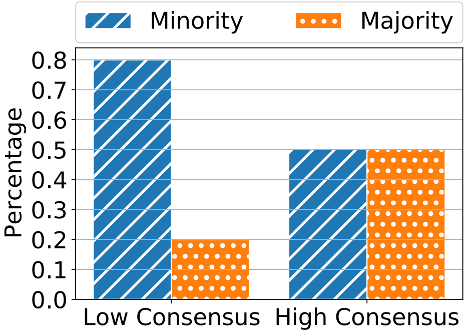

Figure 8: High vs. low-consensus distributions of the PhD-detection attack on PATE: vote histograms of minority-group members present lower consensus, allowing an attacker to identify them.

F Edge values for noise

Here, our purpose is to evaluate our attack given extremely low and extremely high values of σ. We repeated the query-number-limited attack from Section 4.1 where adversaries perform 10^4 queries. This time, we used a σ value approaching 0 and a very high one (400). Figure 9 shows that when noise is close to 0, the error rate is the highest, it then drops and climbs again as we increase the error. This is consistent with what we would expect: we know that when $\sigma = 0$, the attacker cannot learn anything but the argmax class, whereas if σ is infinitely large, PATE's output distribution is uniform regardless of the underlying votes, and the attacker again cannot learn anything.

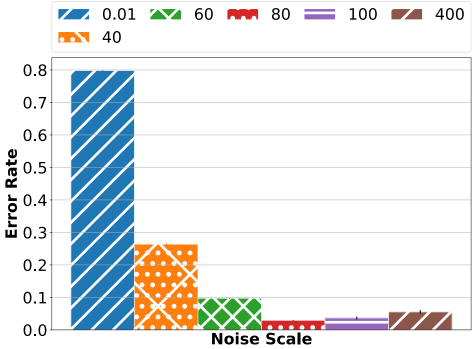

Figure 9: Error rates with baselines of a median-consensus histogram (from H3) in SVHN. When the noise is close to 0, the error is the largest; at some point, the error starts moderately increasing as the noise increases.