Over-the-Air Ensemble Inference with Model Privacy

Selim F. Yilmaz, Burak Hasırcıoğlu, Deniz Gündüz

Abstract

We consider distributed inference at the wireless edge, where multiple clients with an ensemble of models, each trained independently on a local dataset, are queried in parallel to make an accurate decision on a new sample. In addition to maximizing inference accuracy, we also want to maximize the privacy of local models. We exploit the superposition property of the air to implement bandwidth-efficient ensemble inference methods. We introduce different over-the-air ensemble methods and show that these schemes perform significantly better than their orthogonal counterparts, while using less resources and providing privacy guarantees. We also provide experimental results verifying the benefits of the proposed over-the-air inference approach, whose source code is shared publicly on Github.

Index Terms

over-the-air computation, edge inference, differential privacy, ensemble inference, multi-class classification.

I. Introduction

The increasing adoption of Internet-of-Things (IoT) devices results in the collection and processing of massive amounts of mobile data at the wireless edge. Conventional centralized machine learning (ML) methods are impractical for edge applications due to privacy concerns and limited communication resources. Implementing decentralized ML models at the edge solves this issue, and thus, *edge learning* and *edge inference* have attracted significant attention over

The authors are with Department of Electrical and Electronic Engineering, Imperial College London, UK. Deniz Gündüz is also with Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia (UNIMORE), Italy. Email: {s.yilmaz21, b.hasircioglu18, d.gunduz}@imperial.ac.uk

The present work has received funding from the European Union's Horizon 2020 Marie Skłodowska Curie Innovative Training Network Greenedge (GA. No. 953775). This work was partially funded by the European Research Council (ERC) through Starting Grant BEACON (no. 677854) and by the UK EPSRC (grant no. EP/T023600/1) under the CHIST-ERA program.

the recent years [1]–[4]. Edge learning aims to train large ML models in a distributed setting, whereas edge inference aims to make inferences in a distributed manner at the edge.

Although collaborative training at the edge can bring significant advantages, it requires significant coordination and communication across nodes. Moreover, limited wireless resources are a major bottleneck, and noise, interference, and lack of accurate channel state information can prevent or slow down convergence of learning algorithms or results in a reduced accuracy [5]. Therefore, in this paper, we consider collaborative inference using independently trained model at the edge nodes. While a growing body of work studies distributed learning over wireless networks, the literature on distributed wireless inference, particularly using deep learning techniques, is relatively limited [6]–[8].

We treat the resultant problem as an ensemble inference problem, where the individual hypotheses of the nodes need to be conveyed to the querying server, and combined for the most accurate decision. Ensemble learning methods combine multiple hypotheses instead of constructing a single best hypothesis to model the data [9]. In ensemble learning, each hypothesis vote for the final decision, where votes can have weights depending on their confidence. It is generally intractable to find the optimal hypothesis, and choosing a model among a set of equally-good models has the risk of choosing the model that has worse generalization performance; however, averaging these models would reduce this risk [9], [10]. Furthermore, weighted or voting based ensemble methods have theoretical guarantees, e.g., expected error of an averaging ensemble of models is not greater than the average of expected errors of the individual models with a mean square objective [10].

Privacy is an important concern in all ML applications since the data about individuals can reveal sensitive information about them. In the case of ensemble inference, when the models are queried, their outputs may reveal sensitive information about their training sets. For instance, even when an adversary has black-box access to the models, whether or not a data point is used during training can be inferred via membership inference attacks [11], or even the whole model can be reconstructed via model inversion attacks [12]. Hence, even if adversaries can only observe the inference results, we need to introduce some additional mechanisms to protect the sensitive information.

Differential privacy (DP) guarantees can be obtained via introducing additional randomness to the output, such as adding noise at the expense of some accuracy loss. Since DP bounds the amount of information leaked about the individuals, DP mechanisms make black-box attacks less effective. One approach to provide DP guarantees to ML is differentially private training [13]. Typically, Gaussian noise is added to the gradients during training, where the noise variance is determined according to the desired privacy level. This approach is extended to a federated setting in [14].

In this work, we are interested in enabling distributed inference at the edge while limiting the privacy leakage. One straightforward approach is to train the models in a DP manner. However, in this case, a fixed DP guarantee is achieved, and we cannot operate at different privacy-utility trade-offs during inference, which may be beneficial when serving users with different levels of trustworthiness. Moreover, DP training does not prevent the model stealing attacks since the model can be still reconstructed via black-box access to it. Hence, in this paper, we focus on embedding privacy-preserving mechanisms into the inference phase. We simply lift DP training assumption on the models and assume non-private training.

In a recent line of work [15]–[17], it has been shown that, in distributed training tasks, over-the-air computation (OAC) can be exploited to use communication resources much more efficiently, and to significantly improve the learning performance. Instead of conventional digital communication, in OAC, clients transmit their updates simultaneously in an uncoded manner such that the receiver automatically gets the aggregated signal. Hence, besides communication efficiency, OAC also helps preserving privacy of the clients. Any noise received simultaneously with the aggregated signal at the receiver is effective at preserving the privacy of all the signals transmitted by the clients [18]–[22]. In this work, we extend the use of OAC beyond distributed training and exploit it for efficient and private distributed edge inference.

In particular, we introduce two different ensemble methods along with our private edge inference exploiting OAC. Our main contributions are as follows:

1) To the best of our knowledge, this is the first work to employ OAC for distributed inference through an ensemble of models. We show that OAC improves both the privacy and the bandwidth efficiency.
2) We provide flexible privacy guarantees depending on the scenario without imposing any restrictions on the training phase.
3) We systematically compare and discuss privacy of the introduced ensemble methods, and show that the proposed framework with OAC performs significantly better than orthogonal counterparts while using less resources.
4) To facilitate further research and reproducibility, we publicly share the source code of our

framework on github.com/selimfirat/oac-based-private-ensembles.

II. System Model and Problem Definition

Notation: Boldface lowercase letters denote vectors (e.g., $\boldsymbol{p}$), boldface uppercase letters denote matrices (e.g., $\boldsymbol{P}$), non-boldface letters denote scalars (e.g., p or P), and uppercase calligraphic letters denote sets (e.g., $\mathcal{P}$). Blackboard bold letters denote function domains (e.g., $\mathbb{P}$). $\mathbb{R}$, $\mathbb{N}$, $\mathbb{C}$ denote the set of real, natural and complex numbers, respectively. We define $[n] \triangleq \{1, 2, \ldots, n\}$, where $n \in \mathbb{N}$.

System Model: We consider privacy-preserving ensemble classification at the wireless edge. In this setting, there are n clients each with a separate trained model $f_i : \mathbb{R}^d \to \mathbb{R}^k, i \in [n]$, for a classification task. We assume that local models are trained by using non-intersecting datasets.

We assume that the clients are connected to a central inference server (CIS) via a wireless medium, and, at time t, we assume each client i knows its channel gain $h_{i,t} \in \mathbb{C}$. In our setting, the channel gains change across users and time steps, but they stay the same per inference round. To reduce the total power consumption and to amplify the privacy guarantees, we consider random participation of the clients in each inference round such that each client i independently participates with probability p. To limit the power consumption, only the clients whose channel gains are larger than a certain threshold participate the inference. This is one of the sources of randomness determining p. Hence, p is a tunable parameter via such a transmission threshold. If necessary, via additional randomness, p can be made even smaller. Each participating client makes a prediction denoted by $f_i(\boldsymbol{x}_t)$. The clients have a bandwidth of k channel uses to convey their predictions to the CIS.

Let $\boldsymbol{y}_{i,t} \in \mathbb{R}^k$ denote the signal transmitted by client i. The received signal at CIS is

$$\boldsymbol{z}_t = \sum_{i \in \mathcal{P}_t} h_{i,t} \boldsymbol{y}_{i,t} + \boldsymbol{n}_t, \tag{1}$$

where $\boldsymbol{n}_t \in \mathbb{R}^k$ is the independently and identically distributed (i.i.d.) additive white Gaussian noise (AWGN) with variance $\sigma^2_{\text{channel}}$, i.e., $\boldsymbol{n}_t \sim \mathcal{N}(\boldsymbol{0}, \sigma^2_{\text{channel}} \boldsymbol{I}_k)$.

After receiving $\boldsymbol{z}_t$, CIS processes it via a function $s : \mathbb{R}^k \to [k]$ and outputs the most probable class.

Threat Model: In our problem, the purpose is to limit the privacy leakage of clients' local models. This is equivalent to limiting the leakage about the individual datasets $\mathcal{D}_i, i \in [n]$. In our threat model, we assume all the clients are trusted, i.e., they are not interested in the sensitive

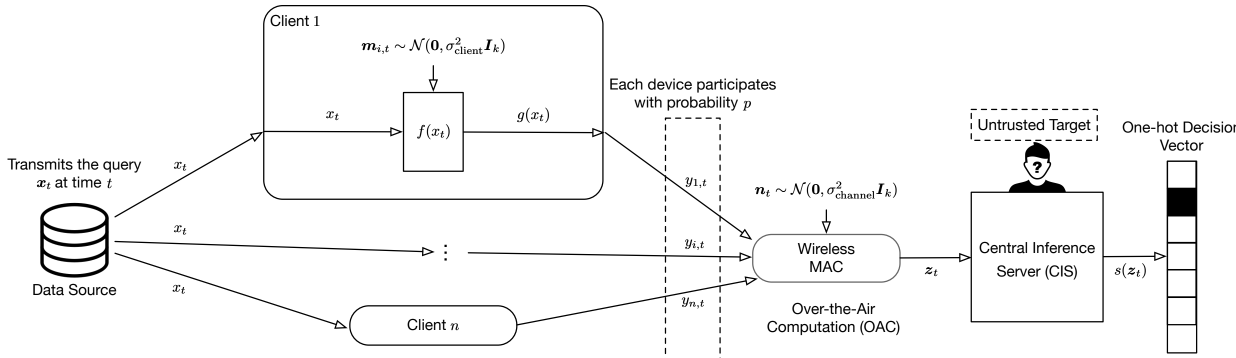

Fig. 1. Overview of our ensemble framework for private inference.

features of the training datasets. On the other hand, CIS is honest but curios, i.e., it does not deviate from the protocol, but by using the signals it receives from the clients, it may try to infer sensitive information about the datasets. Hence, our goal is to limit the leakage to CIS about the datasets via $\boldsymbol{z}_t$ while trying to maximize the inference accuracy.

III. Methodology

Here, we introduce the modules of our framework gradually, which is summarized in Fig. 1.

A. *Ensemble Methods*

Having received the query $\boldsymbol{x}_t$, each participating client makes a local prediction. We present alternative ways of doing this by introducing different classes of models, f_i's. Common to all of them, let $\boldsymbol{r}_{i,t} \in \mathbb{R}^k$ be a vector containing classifier scores (beliefs) for each class, where k is the number of classes and j^{th} element of $\boldsymbol{r}_{i,t}$, denoted by $(\boldsymbol{r}_{i,t})_j$, contains the score of client i for class j. We normalize the sum of the scores in $\boldsymbol{r}_{i,t}$ to 1, i.e., $\|\boldsymbol{r}_{i,t}\|_1 = 1$, and hence, the maximum possible score of a class is 1.

Definition 1. $\text{ToOneHot}(j, l)$ *function outputs an l dimensional one-hot vector for $j \leq l$, where only the j^{th} dimension is 1 and the rest are 0.*

Belief summation method sums beliefs of the participating clients for all the classes and the CIS later selects the class with the highest total score. Thus, it uses the following model for client i:

$$f_i(\boldsymbol{x}_t) = \boldsymbol{r}_{i,t}. \tag{2}$$

Majority voting with OAC method allows participating clients to vote for a class and the CIS later selects the class with highest number of votes. Hence, it uses the following model for client i:

$$f_i(\boldsymbol{x}_t) = \text{ToOneHot}\left(\arg\max_{j\in[k]}(\boldsymbol{r}_{i,t})_j, k\right). \tag{3}$$

Hence, while belief summation with OAC combines local discriminative scores, majority voting with OAC combines predicted labels.

B. Ensuring Privacy

Next, we explain how we make our inference procedure privacy-preserving by introducing some randomness. First, we formally define DP for our ensemble inference task as follows.

Let $\mathcal{L}$ and $\mathcal{L}'$ be the sets of local models of the clients, which differ at most in one of the clients, i.e., $\mathcal{L} = \{f_j\} \cup \{f_i : i \in [n] \setminus j\}$ and $\mathcal{L}' = \{f'_j\} \cup \{f_i : i \in [n] \setminus j\}$ such that $f_j \neq f'_j$. Such $\mathcal{L}$ and $\mathcal{L}'$ are called *neighboring* sets. In our case, since we aim to protect the local models from CIS, $\boldsymbol{z}_t$ can be considered as a randomized function, and the set of local models $\mathcal{L}$ or $\mathcal{L}'$ can be considered as its inputs. Hence, all the DP guarantees given in the paper will consider local-model-level privacy guarantees.

Definition 2. *Let $M : \mathbb{L} \rightarrow \mathbb{R}^k$ be a randomized algorithm and $\mathcal{L}$ and $\mathcal{L}'$ are two possible neighboring model sets. For $\varepsilon > 0$ and $\delta \in [0, 1)$, M is called (ε, δ)-DP if*

$$\Pr(M(\mathcal{L}) \in \mathcal{R}) \leq e^{\varepsilon} \Pr(M(\mathcal{L}') \in \mathcal{R}) + \delta, \tag{4}$$

for all neighboring pairs $(\mathcal{L}, \mathcal{L}')$ and $\forall\ \mathcal{R} \subset \mathbb{R}^k$.

To achieve DP guarantees, the output released to an adversary should be randomized. In our paper, we consider releasing a noisy version of model outputs, $f_i(\boldsymbol{x}_t)$, for each client with a Gaussian noise [23]. Note that in OAC, $\boldsymbol{z}_t$ already has channel noise, which provides some degree of privacy guarantees. However, to achieve the desired level of DP, channel noise may not be large enough and we cannot control or reliably know its variance. Thus, it is not a reliable source of randomness [22], and we ignore the channel noise while analysing privacy guarantees. Instead, we have each client add some additional Gaussian noise before releasing their contributions. Note that ignoring the channel noise in the privacy analysis results in weaker privacy guarantees. In reality, the privacy guarantees are slightly better than the ones we obtain in this work due to channel noise. We generate a noisy version of our model prediction as follows.

$$g_i(\boldsymbol{x}_t) = f_i(\boldsymbol{x}_t) + \boldsymbol{m}_{i,t}, \tag{5}$$

where $\boldsymbol{m}_{i,t} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2_{\text{client}}\boldsymbol{I}_k)$. One of the main advantages of OAC is that the noise added by different clients are also aggragated at CIS. Thus, it has a further privacy amplification effect. We provide the analysis of the privacy guarantees achieved by our framework in Section IV. This analysis reveals that DP guarantees are directly dependent on the variance of the aggregated noise at the CIS. Hence, to obtain DP guarantees, independent of the number of participating clients, each client should add a Gaussian noise with $\sigma^2_{\text{client}} = \sigma^2/|\mathcal{P}_t|$, where σ^2 is a constant depending on the desired DP guarantees and $\mathcal{P}_t$ is the set of participating clients. Hence, we assume that the number of participating clients is known to the other participating clients, but secret from the CIS.

C. Transmission

We need to make sure that each client's noisy score $g_i(\boldsymbol{x}_t)$ is received at CIS at the same power level. Recall that the channel gain for each client is perfectly known by that client, which then employs channel inversion to cancel its effect. Thus, each client scales the signal by $1/h_{i,t}$. Note that since a client does not participate the inference if its channel has a low gain, this scaling does not result in an excessive power usage. The CIS may require a specific power level for the reception of the signals depending on the available power of the clients. Hence, the clients further scale their signals with a constant denoted by A_t, and the transmitted signal is

$$\boldsymbol{y}_{i,t} = \begin{cases} A_t g_i(\boldsymbol{x}_t)/h_{i,t}, & \text{if } i \in \mathcal{P}_t \\ \boldsymbol{0}, & \text{otherwise} \end{cases} \tag{6}$$

D. Final Decision by CIS

The signal received by the CIS at time t becomes

$$\boldsymbol{z}_t = A_t \left(\sum_{i \in \mathcal{P}_t} \boldsymbol{m}_{i,t} + \sum_{i \in \mathcal{P}_t} f_i(\boldsymbol{x}_t) \right) + \boldsymbol{n}_t. \tag{7}$$

Thus, the variance of total noise received by the CIS at time t is $\sigma^2_{\text{CIS}} = \sigma^2_{\text{channel}} + |\mathcal{P}_t| A_t^2 \sigma^2_{\text{client}}$, i.e., $\boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{0}, \sigma^2_{\text{CIS}}\boldsymbol{I}_k)$. After receiving $\boldsymbol{z}_t$, CIS multiplies the received signal by $\frac{1}{A_t}$ to recover

Algorithm 1 Over-the-Air Private Ensemble Inference

Input: Trained client model $f_i(\cdot)$ for every client i, CIS model $s(\cdot)$, new sample $\boldsymbol{x}_t$ at timestep t

Output: Index of the decided class

function OTA_PRIVATE_ENSEMBLE

  Let $\mathcal{P}_t$ contain each client i with probability p, independently

  for each client $i \in \mathcal{P}_t$ in *parallel* **do**

    Client i receives $\boldsymbol{x}_t$

    Calculate $f_i(\boldsymbol{x}_t)$ ▷ Client Model

    $g_i(\boldsymbol{x}_t) = f_i(\boldsymbol{x}_t) + \mathcal{N}(\mathbf{0}, \sigma^2/|\mathcal{P}_t|\boldsymbol{I}_k)$ ▷ Add noise

    Transmit $A_t g_i(\boldsymbol{x}_t)/h_{i,t}$

  end for

  $\boldsymbol{z}_t = \boldsymbol{n}_t + A_t \sum_{i \in \mathcal{P}_t} g_i(\boldsymbol{x}_t)$ ▷ Air Sum

  CIS receives $\boldsymbol{z}_t$

  return $s(\boldsymbol{z}_t)$ ▷ CIS Model

end function

the desired signal and applies the $\arg\max$ function to decide the most probable class. That is, it applies $s(\boldsymbol{z}_t) = \arg\max_{j \in [k]} \frac{1}{A_t} \boldsymbol{z}_{t,j}$.

Algorithm 1 summarizes all the steps introduced in this section.

IV. Privacy Analysis

In this section, we provide the privacy analysis of the proposed over-the-air ensembling scheme. We first analyze the case in which all the clients participate.

Theorem 1. *If all the clients participate in the inference, i.e., $p = 1$, then, Algorithm 1 is (ε, δ)-DP such that for any $\varepsilon > 0$,*

$$\delta = \Phi(1/(\sqrt{2}\sigma) - \varepsilon\sigma/\sqrt{2}) - e^{\varepsilon}\Phi(-1/(\sqrt{2}\sigma) - \varepsilon\sigma/\sqrt{2}), \tag{8}$$

where Φ is the CDF of standard normal distribution.

Proof. Our theorem is a special case of the following lemma.

Lemma 1 (Theorem 8 in [24])**.** *Let $f : \mathbb{L} \rightarrow \mathbb{R}^k$ be a function with $||f(\mathcal{L})-f(\mathcal{L}')||_2 \leq C$, where $\mathcal{L}$ and $\mathcal{L}'$ are neighboring inputs and $||\cdot||_2$ is L_2 norm. A mechanism $M(\mathcal{L}) = f(\mathcal{L})+\mathcal{N}(0, \tilde{\sigma}^2 \boldsymbol{I}_k)$ is (ε, δ)-DP if and only if*

$$\Phi\left(C/(2\tilde{\sigma}) - \varepsilon\tilde{\sigma}/C\right) - e^{\varepsilon}\Phi\left(-C/(2\tilde{\sigma}) - \varepsilon\tilde{\sigma}/C\right) \leq \delta. \tag{9}$$

To apply Lemma 1 directly in our case, we need to calculate the L_2 sensitivity, C, of $\boldsymbol{z}_t$ without any noise, i.e., $\boldsymbol{m}_{i,t} = \boldsymbol{0}, \forall i \in \mathcal{P}_t$. We denote this quantity by $\tilde{\boldsymbol{z}}_t$. Consider neighboring sets $\mathcal{L}$ and $\mathcal{L}'$. We denote the noiseless vector received by the CIS by $\tilde{\boldsymbol{z}}_t$ when the set of local models is $\mathcal{L}$, and by $\tilde{\boldsymbol{z}}_t'$ when it is $\mathcal{L}'$. Then,

$$C = \max_{\tilde{\boldsymbol{z}}_t, \tilde{\boldsymbol{z}}_t'} \|\tilde{\boldsymbol{z}}_t - \tilde{\boldsymbol{z}}_t'\|_2 = \max_{\tilde{\boldsymbol{z}}_t \tilde{\boldsymbol{z}}_t'} \left(\sum_{j=1}^{k} (\tilde{\boldsymbol{z}}_{t,j} - \tilde{\boldsymbol{z}}_{t,j}')^2 \right)^{1/2}. \tag{10}$$

We know that $\tilde{\boldsymbol{z}}_{t,j} \in [0, A_t], \forall j \in [k]$ and $\|\tilde{\boldsymbol{z}}_{t,j}\|_1 = A_t$. The same also applies to $\tilde{\boldsymbol{z}}_t'$. Hence, $\|\tilde{\boldsymbol{z}}_t - \tilde{\boldsymbol{z}}_t'\|_2$ is maximized when $\tilde{\boldsymbol{z}}_t$ and $\tilde{\boldsymbol{z}}_t'$ have only one non-zero element, and the indices of these non-zero elements are different in both vectors. Then, $C = \max_{\tilde{\boldsymbol{z}}_t, \tilde{\boldsymbol{z}}_t'} \|\tilde{\boldsymbol{z}}_t - \tilde{\boldsymbol{z}}_t'\|_2 = \sqrt{2}A_t$.

Finally, by substituting $C = \sqrt{2}A_t$ and $\tilde{\sigma} = \sigma A_t$ into (9), we obtain (8). □

Next, we present the amplification effect of client sampling on the privacy guarantees.

Theorem 2. *If each client independently participate in inference with probability $p < 1$, then Algorithm 1 is (ε', δ')-DP, where, for any $\varepsilon' > 0$,*

$$\delta' = \frac{p}{1-(1-p)^n}\Big(\Phi(1/(\sqrt{2}\sigma) - \varepsilon\sigma/\sqrt{2}) - e^{\varepsilon}\Phi(-1/(\sqrt{2}\sigma) - \varepsilon\sigma/\sqrt{2})\Big), \tag{11}$$

where $\varepsilon = \log(1 + ((1-(1-p)^n)/p)(e^{\varepsilon'} - 1))$.

Proof. Without loss of generality, let $\mathcal{L}$ and $\mathcal{L}'$ are two neighboring sets of models differing only in the first client's model, i.e. it is either f_1 or f_1'. Let us write the output distribution of Algorithm 1 as mixture distributions. When the model set is $\mathcal{L}$, we have $\mu = (1-\eta)\mu_0 + \eta\mu_1$ and when the model set is $\mathcal{L}'$, we have $\mu' = (1-\eta)\mu_0 + \eta\mu_1'$. In these expressions, η is the probability that client 1 is sampled, μ_0 is the probability distribution when client 1 is not sampled, μ_1 is the probability distribution when client 1 is sampled and the model set is $\mathcal{L}$ and μ_1' is the probability distribution when client 1 is sampled and the model set is $\mathcal{L}'$. Recall that we sample client models each with probability p from $\mathcal{L}$ or $\mathcal{L}'$, and the CIS receives non-zero vectors only when $|\mathcal{P}_t| > 0$. Hence, $\eta = \Pr\{\text{Client 1 is sampled} \mid |\mathcal{P}_t| > 0\}$, resulting in $\eta = p/(1-(1-p)^n)$ via Bayes' rule.

Lemma 2 (Theorem 1 in [25])**.** *A mechanism $\mathcal{M}$ is (ε', δ')-DP if and only if*

$$\sup_{\mathcal{L},\mathcal{L}'} D_\alpha(\mathcal{M}(\mathcal{L})||\mathcal{M}(\mathcal{L}')) \leq \delta', \tag{12}$$

where $\alpha = e^{\varepsilon'}$ and $D_\alpha(\mu||\mu') \triangleq \int_Z \max\{0, d\mu(z) - \alpha d\mu'(z)\}d(z)$.

Lemma 2 implies that it is enough to bound $D_\alpha(\mu||\mu')$ to provide DP guarantees. For this, we use the relation in Lemma 3, which is called *advanced joint convexity* of D_α.

Lemma 3 (Theorem 2 in [25])**.** *For $\alpha \geq 1$, we have*

$$D_{\alpha'}\left(\mu||\mu'\right) = \eta D_\alpha\left(\mu_1||(1-\beta)\mu_0 + \beta\mu_1'\right) \tag{13}$$

where $\alpha' = 1 + \eta(\alpha - 1)$ and $\beta = \alpha'/\alpha$.

We further upper bound (13) via convexity:

$$D_{\alpha'}\left(\mu||\mu'\right) \leq \eta(1-\beta)D_\alpha(\mu_1||\mu_0) + \eta\beta D_\alpha(\mu_1||\mu_1'). \tag{14}$$

To bound $D_\alpha(\mu_1||\mu_0)$, observe that there exist a coupling between μ_1 and μ_0 as follows. For μ_0, to guarantee $|\mathcal{P}_t| > 0$, let us first sample exactly one client c other than client 1 since we know that client 1 is not sampled. Then apply Poisson sampling on the remaining set, i.e., $[n] \setminus \{c, 1\}$, to determine the other participating clients. For μ_1, assume we have the same realization of Poisson sampling on $[n] \setminus \{c, 1\}$ as in μ_0. Further, by definition of μ_1, client 1 is also sampled. Hence, μ_1 and μ_0 can be seen as output distributions of Algorithm 1 such that the input client sets differ in only one element. Hence, $D_\alpha(\mu_1||\mu_0) \leq \delta$ due to Theorem 1. Similarly, to bound $D_\alpha(\mu_1, \mu_1')$, a coupling exists between μ_1 and μ_1' such that user 1 is sampled and f_1 and f_1' are the models in user 1, for μ_1 and μ_1', respectively. To determine the other participating clients, the same realization of Poisson sampling on $[n] \setminus \{1\}$ is applied in both μ_1 and μ_1'. Since the input client sets also differ in one element, in this case, due to Theorem 1, we have $D_\alpha(\mu_1, \mu_1') \leq \delta$. If we put the bounds for $D_\alpha(\mu_1, \mu_0)$ and $D_\alpha(\mu_1, \mu_1')$ into (14), we obtain $D_{\alpha'}(\mu||\mu') \leq \eta\delta$, from which (11) follows. The expression for ε can be directly derived from the expression $\alpha' = 1 + \eta(\alpha - 1)$. □

V. Simulations

A. The Datasets and Experimental Setup

We employ four different datasets to demonstrate the effectiveness of our framework: CIFAR-10, CIFAR-100, FashionMNIST and IMDB. CIFAR-10 contains 50.000 training images, 10.000

test images, and 10 target classes [26]. CIFAR-100 contains the same splits except that target classes are partitioned into 100 subclasses [26]. FashionMNIST has 50.000 training images, 10.000 test images, and 10 target classes [27]. IMDB dataset has 25.000 training texts, 25.000 test texts, and 2 target classes [28]. For all datasets, we use predefined training and test sets, except that we split 10% of the training set as the validation set and only use the remaining 90% for training.

For image datasets, we use MobileNetV3-Large [29] except we change its final layer to make it compatible with the target number of classes. Instead of training from scratch, we fine-tune a pre-trained version [30] of it for 50 epochs. To make sizes of the images compatible to our network, we interpolate them to 224×224 images. Since the network receives three channel inputs, for each FashionMNIST sample, we feed the same single channel grayscale image to all input channels. For text datasets, we use DistilBERT-base-uncased [31] model, and again, we fine-tune a pre-trained model [32] for 3 epochs.

We repeat all the experiments with 5 different random seeds, and report the average results. We compute and report Macro-F1 scores by averaging per-class F1 scores on the test set. We randomly split the training data among the clients equally. We consider $n = 20$ clients with a participation probability of $p = 1.0$ and a channel signal-to-noise ratio (SNR) of 10 dB, except when they are changed gradually in Section V-C.

B. Comparison with the Baselines

In Table I, in terms of their Macro-F1 scores, we compare the proposed OAC-based methods with the best client model and the ensemble methods with orthogonal transmission. We choose the model with the highest Macro-F1 score on the same validation set as the best client model. For fairness, the client having the best model transmits its inference over the k channels. In orthogonal methods, all the devices transmit their inferences via different channels, i.e., $|\mathcal{P}_t| \times k \in O(nk)$ channels in total. We observe that, compared to the best client model, ensemble methods significantly improve the test scores, especially in the private setting. Moreover, while orthogonal and OAC-based methods perform competitively in the non-private setting, when privacy is involved, best client model and orthogonal methods perform near-random, and significantly worse than the OAC-based methods. Note that orthogonal methods use $|\mathcal{P}_t| \times k$ channels, whereas OAC-based methods only use k channels; yet, OAC-based methods outperform orthogonal ones in the private setting.

TABLE I

COMPARISON WITH THE BASELINES

Privacy	Method	CIFAR-10	CIFAR-100	FashionMNIST	IMDB
$\epsilon = \infty$	Best Client Model	$86.37_{\pm 0.33}$	$44.73_{\pm 1.60}$	$89.55_{\pm 0.23}$	$89.31_{\pm 0.31}$
	Orthogonal Majority Voting	$89.97_{\pm 0.14}$	$62.51_{\pm 0.74}$	$\mathbf{91.92_{\pm 0.15}}$	$90.59_{\pm 0.06}$
	Orthogonal Belief Summation	$90.09_{\pm 0.12}$	$\mathbf{63.85_{\pm 0.60}}$	$91.91_{\pm 0.11}$	$\mathbf{90.64_{\pm 0.05}}$
	Majority Voting with OAC	$89.96_{\pm 0.14}$	$62.55_{\pm 0.67}$	$\mathbf{91.92_{\pm 0.13}}$	$90.62_{\pm 0.10}$
	Belief Summation with OAC	$\mathbf{90.14_{\pm 0.16}}$	$63.83_{\pm 0.59}$	$91.91_{\pm 0.13}$	$\mathbf{90.64_{\pm 0.07}}$
$\epsilon = 1$	Best Client Model	$12.19_{\pm 0.22}$	$1.20_{\pm 0.05}$	$12.29_{\pm 0.30}$	$53.58_{\pm 0.32}$
	Orthogonal Majority Voting	$22.59_{\pm 0.10}$	$2.41_{\pm 0.14}$	$23.43_{\pm 0.55}$	$65.29_{\pm 0.23}$
	Orthogonal Belief Summation	$22.22_{\pm 0.13}$	$2.22_{\pm 0.12}$	$23.30_{\pm 0.55}$	$64.94_{\pm 0.24}$
	Majority Voting with OAC	$\mathbf{81.27_{\pm 0.10}}$	$\mathbf{24.24_{\pm 0.21}}$	$\mathbf{84.18_{\pm 0.21}}$	$\mathbf{89.32_{\pm 0.07}}$
	Belief Summation with OAC	$80.13_{\pm 0.24}$	$20.04_{\pm 0.24}$	$83.81_{\pm 0.22}$	$89.15_{\pm 0.08}$

Previous studies suggest that ensembling via belief averaging generally performs better than majority voting [33], [34]. Our non-private results also support this argument as beliefs contain more information compared to conveying local decisions. However, when $\varepsilon = 1$, majority voting outperforms belief summation for both orthogonal and OAC-based settings. This can be explained by the fact that the increasing noise levels result in relatively unreliable beliefs, since the individual values of beliefs are smaller, and thus more sensitive to the noise added for privacy.

C. Analysis of Ensembles with OAC for Varying Conditions

Fig. 2 shows the performance of our OAC-based methods on CIFAR-10 dataset for varying channel SNR, p, and ε values. The left figure shows that the performance of the methods slightly increases as the channel SNR increases, especially for SNR values below 2 dB. In the right figure, we observe that higher p improves the performance significantly in the private setting $(\varepsilon = 1)$. Although lower p has a privacy amplification effect which decreases the noise variance required to attain $\varepsilon = 1$, we observe that its privacy amplification effect is not as significant as the impact of a fewer client participation on the inference performance. In the non-private setting $(\varepsilon = \infty)$, having higher participation also helps to get higher macro-F1 score, but not as much as in the private setting. These plots also show that private setting is more sensitive to these varying conditions for both p and channel SNR.

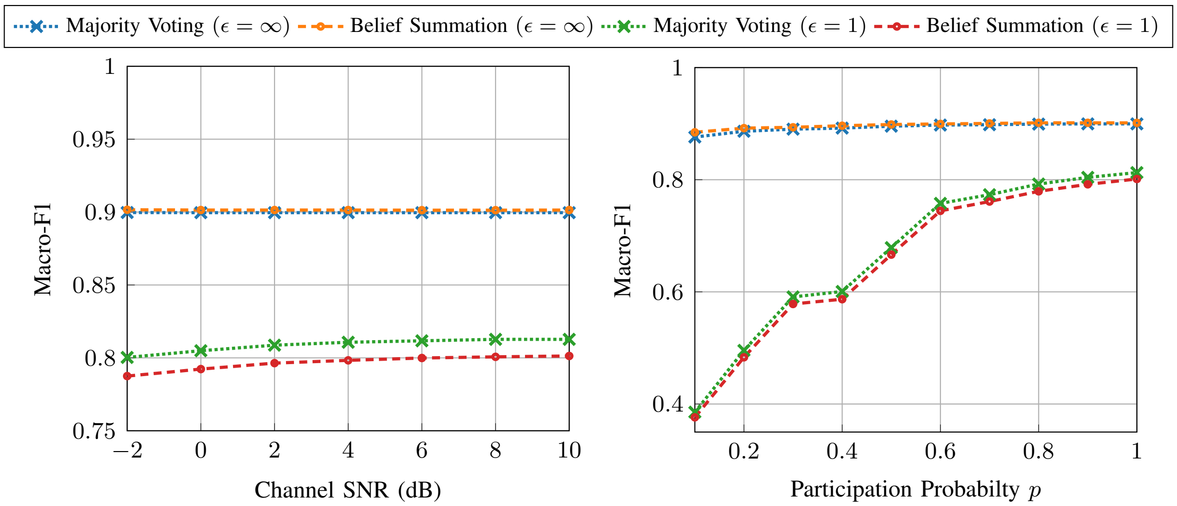

Fig. 2. Comparison of ensemble methods with OAC for varying channel SNR (left) and participation probability p (right) on CIFAR-10 dataset.

VI. Conclusion

We have introduced a private edge inference framework with ensembling. We have exploited OAC for bandwidth-efficient and private wireless edge inference for the first time in the literature. We have provided DP guarantees exploiting both distributed noise addition and random participation. We have systematically evaluated the introduced ensemble methods with OAC and shown that distributed edge inference with OAC performs significantly better than its orthogonal counterpart while using less resources. We have observed that while transmitting class scores from each client is more informative as an ensembling method, making and transmitting local decisions can be more reliable when noise is introduced to guarantee privacy.

References

[1] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *arXiv preprint arXiv:2104.02151*, 2021.

[2] C.-J. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jia, B. Jia *et al.*, "Machine learning at facebook: Understanding inference at the edge," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2019, pp. 331–344.

[3] Q. Lan, Q. Zeng, P. Popovski, D. Gündüz, and K. Huang, "Progressive feature transmission for split inference at the wireless edge," *arXiv preprint arXiv:2112.07244*, 2021.

[4] D. Gündüz, D. B. Kurka, M. Jankowski, M. M. Amiri, E. Ozfatura, and S. Sreekumar, "Communicate to learn at the edge," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 14–19, 2020.

[5] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Guest editorial special issue on distributed learning over wireless edge networks—part i," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3575–3578, 2021.

[6] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Wireless image retrieval at the edge," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 89–100, 2020.

[7] ——, "Joint device-edge inference over wireless links with pruning," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2020, pp. 1–5.

[8] J. Shao, Y. Mao, and J. Zhang, "Learning task-oriented communication for edge inference: An information bottleneck approach," *arXiv preprint arXiv:2102.04170*, 2021.

[9] T. G. Dietterich *et al.*, "Ensemble learning," *The handbook of brain theory and neural networks*, vol. 2, no. 1, pp. 110–125, 2002.

[10] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[11] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[12] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 601–618.

[13] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[14] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[15] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[16] ——, "Federated learning over wireless fading channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3546–3557, 2020.

[17] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 491–506, 2019.

[18] M. Seif, R. Tandon, and M. Li, "Wireless federated learning with local differential privacy," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 2604–2609.

[19] D. Liu and O. Simeone, "Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 170–185, 2020.

[20] A. Sonee and S. Rini, "Efficient federated learning over multiple access channel with differential privacy constraints," *arXiv preprint arXiv:2005.07776*, 2020.

[21] M. S. E. Mohamed, W.-T. Chang, and R. Tandon, "Privacy amplification for federated learning via user sampling and wireless aggregation," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3821–3835, 2021.

[22] B. Hasırcıoğlu and D. Gündüz, "Private wireless federated learning with anonymous over-the-air computation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5195–5199.

[23] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[24] B. Balle and Y.-X. Wang, "Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," in *International Conference on Machine Learning*. PMLR, 2018, pp. 394–403.

[25] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: tight analyses via couplings and divergences," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 6280–6290.

[26] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[27] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

[28] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[29] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[30] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 1485–1488.

[31] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[32] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[33] L. I. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 2, pp. 281–286, 2002.

[34] D. Wang, H. Xu, and Q. Wu, "Averaging versus voting: A comparative study of strategies for distributed classification," *Mathematical Foundations of Computing*, vol. 3, no. 3, p. 185, 2020.