Learning to Generate Image Embeddings with User-level Differential Privacy

Zheng Xu* Maxwell Collins* Yuxiao Wang Liviu Panait Sewoong Oh
Sean Augenstein Ting Liu Florian Schroff H. Brendan McMahan

Google Research

Abstract

Small on-device models have been successfully trained with user-level differential privacy (DP) for next word prediction and image classification tasks in the past. However, existing methods can fail when directly applied to learn embedding models using supervised training data with a large class space. To achieve user-level DP for large image-to-embedding feature extractors, we propose DP-FedEmb, a variant of federated learning algorithms with per-user sensitivity control and noise addition, to train from user-partitioned data centralized in the datacenter. DP-FedEmb combines virtual clients, partial aggregation, private local fine-tuning, and public pretraining to achieve strong privacy utility trade-offs. We apply DP-FedEmb to train image embedding models for faces, landmarks and natural species, and demonstrate its superior utility under same privacy budget on benchmark datasets DigiFace, EMNIST, GLD and iNaturalist. We further illustrate it is possible to achieve strong user-level DP guarantees of $\epsilon < 2$ while controlling the utility drop within 5%, when millions of users can participate in training .

1. Introduction

Representation learning, by training deep neural networks as feature extractors to generate compact embedding vectors from images, is a fundamental component in computer vision. Metric learning, a kind of representation learning using supervised data, has been widely applied to image recognition, clustering, and retrieval [61, 75, 77]. Machine learning models have the capacity to memorize training data [10, 11], leading to privacy risks when the models are deployed. Privacy risk can also be audited by membership inference attacks [9, 63], i.e. detecting whether certain data was used to train a model and potentially exposing users' usage behaviors. Defending against such risks is a critical responsibility when training on privacy-sensitive data.

*The first two authors contributed equally. Correspondence to Zheng Xu xuzheng@google.com

Differential Privacy (DP) [23] is an extensively used quantifiable measurement of privacy risk, now generally accepted as a standard notion of privacy in both industry and government [5, 18, 50, 70]. Applied to machine learning, DP requires a training procedure with explicit randomness, and guarantees that the distribution over output models is quantifiably similar given a certain scope of change to the training dataset. A DP guarantee with respect to the change of a single arbitrary training example is known as *example-level DP*, which provides plausible deniability (in the binary hypothesis testing sense of [38]) that any single example (e.g., image) occurred in the training dataset. If we instead consider how the distribution of output models changes if the data (including even the number of examples) from any single user change arbitrarily, we have *user-level DP* [22]. This ensures model training is quantifiably insensitive to all of the data from any one user, and hence it is impossible to tell if a user has participated in training with high confidence. This guarantee can be exponentially stronger than example-level DP if one user may contribute many examples to training.

Recently, DP-SGD [1] (essentially, SGD with the additional steps of clipping each individual gradient to have a maximum norm, and adding correspondingly calibrated noise) has been used to achieve example-level DP for relatively large models in language modeling and image classification tasks [4, 16, 42, 45, 81], often utilizing techniques like large batch training and pretraining on public data. DP-SGD can be modified to guarantee user-level DP, which is often combined with federated learning algorithms and called DP-FedAvg [49]. User-level DP has only been studied for small on-device models that have less than 10 million parameters [36, 49, 57].

We consider user-level DP for relatively large models in representation learning with supervised data. In our setting, similar to federated learning (FL), the data are user-partitioned; but in contrast to decentralized FL, we are primarily motivated by centralized data that benefit from access to richer computation resources and the ability to form virtual clients at random. Throughout this work, we use *user* as the basic unit of data partitioning and the granular-

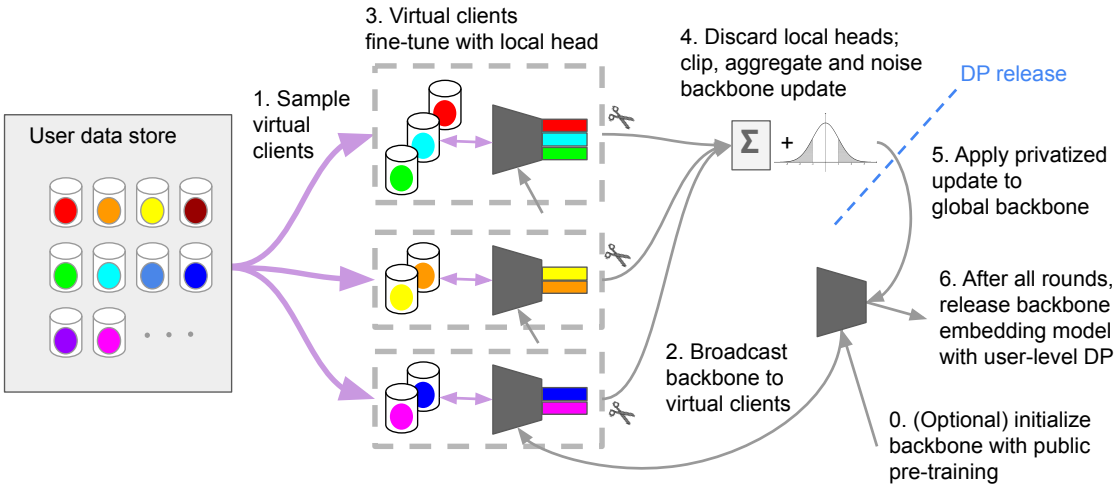

Figure 1. DP-FedEmb combines virtual clients, local fine-tuning, partial aggregation, and public pretraining to achieve strong privacy utility trade-offs. Colors indicate different users and the simplified case of a single class per user, coloring the softmax head accordingly.

ity for privacy; a user owns their (image) data, and *class*, *identity* and *label* are used interchangeably for the supervised information.

Though typically each user only contributes images for a small number of classes, the combined class space from the union of all users can be very large, which proves challenging for existing DP algorithms. When the model size is fixed independent of the number of users, and at a relatively small scale with a few million parameters, previous methods can only achieve strong user-level DP when millions of users are available [36, 49, 57]. In contrast, when considering learning embedding models for applications like facial images, the number of classes (and hence, the total model size) can grow linearly with the number of users, and so simply scaling up to larger datasets with more users no longer ensures that good privacy-utility trade-offs can be achieved. For example, in a standard multi-class training paradigm with 128-dimensional embedding vectors, with one million users we expect the final dense layer for prediction alone to have over 128 million trainable parameters. Further, the fact that most users will only have examples from a small number of classes implies that gradients are approximately sparse, whereas DP-SGD requires the addition of dense noise to the full gradient, leading to a poor signal-to-noise ratio in the updates. Hence, existing methods can easily fail on the problems we consider.

We propose DP-FedEmb to train embedding models with user-level differential privacy; Fig. 1 provides a high-level overview. DP-FedEmb combines public pretraining, virtual clients, local fine-tuning, and partial aggregation to achieve strong privacy-utility trade-offs. The key to the approach is partitioning the model into a backbone network that generates embeddings, and a classification softmax head specific to the classes in the training data. In each training round, users are grouped into virtual clients and initialized from the global backbone. A local randomly-initialized softmax head layer is added for the limited number of classes on the virtual client, and the complete local model is fine-tuned in order to produce an update to the backbone. The local head parameters are not included in the private aggregation and hence require no noise addition. This is in contrast to existing methods like DP-FedAvg/DP-SGD, which would add noise to all parameters including the softmax head. The backbone updates are clipped to a maximum L2 norm, aggregated across virtual clients, and combined with appropriate DP noise. At this point, the noised update is the output of a DP mechanism and satisfies the corresponding DP guarantee. This update is then applied to the global backbone, which inherits the DP guarantee, and passed to the next round of training. DP-FedEmb significantly improves the scalability of DP training for embedding models, as only the parameters of the backbone network are privatized and released, and the size of this portion of the model does not grow with the number of users. Pretraining the backbone network on public data to learn general visual representations before applying DP-FedEmb for more privacy sensitive tasks further improves performance.

We demonstrate the superior performance of DP-FedEmb for embedding models by experiments on datasets with moderate size of users and classes (DigiFace of $98.96K$ or 9047 users / identities, Google Landmarks Dataset of 1262 users and 2028 classes, and iNaturalist of 9275 users and 1203 classes). We also show relatively strong privacy guarantees of single digit ϵ can be achieved while maintaining strong utility if millions users can participate in training. To our knowledge, this is the first report of training a commonly used large vision model, ResNet-50, with non-negligible noise for user-level DP.

2. Learning Embedding Models

2.1. Problem formulation and centralized training

We learn a *backbone* network parameterized by θ, that outputs an embedding vector $z = f(\theta, x)$ for an input image x. The backbone network, θ, is trained on paired examples of image x and class y. The training dataset is naturally partitioned by M users, i.e., $\mathcal{D} = \bigcup_{i=1}^{M} \mathcal{D}_i$.

We adopt the popular multi-class training framework for embedding models, where a proxy weight vector w_j is learnt for each class j. We use ω to denote the union of proxy weight vectors $\{w_j\}$, which is called the *head* of the network. Given a training image-class pair (x, y), logits are computed by taking the inner product between the embedding vector $f(\theta, x)$ and the proxy weight vectors in the network head ω. This is effectively passing $f(\theta, x)$ through a dense network layer parameterized by ω without the bias terms. With a supervised training loss ℓ, such as the cross entropy loss, the following objective is optimized

$$\min_{\theta,\omega} \sum_{(x,y)\in\mathcal{D}} \ell(\langle \omega,\ f(\theta, x)\rangle,\ y)\,, \tag{1}$$

where we overload the inner product notation, i.e., $\langle \omega,\ f(\theta, x)\rangle$, to denote a set of inner products with each element in ω. Typically, variants of the gradient descent method are used to solve the optimization in (1). In each iteration, the average gradient is computed on a sampled minibatch of data $B \subset \mathcal{D}$, and is then used to update the model parameters θ and ω. Furthermore, when the output space, i.e., the number of classes, is very large, sampled softmax is often applied [35], where only a subset of proxy weights sampled from ω are used in each training iteration.

2.2. User-level DP and DP-FedAvg

To achieve user-level DP, we control the sensitivity of each user and add corresponding noise for anonymization. To effectively control the sensitivity, it is important to understand and account for the contributions of each user in the model updates; hence it is convenient to consider the data at a granularity of users instead of individual samples. Grouping together each user's data, the objective (1) can be rewritten as

$$\min_{\theta,\omega} \sum_{i=1}^{M} \sum_{(x,y)\in\mathcal{D}_i} \ell(\langle \omega,\ f(\theta, x)\rangle,\ y)\,. \tag{2}$$

The above objective of two level sum is often found in federated learning [73], which can be optimized by the (generalized) FedAvg algorithm [48, 58]. In generalized FedAvg, each round t starts with the server broadcasting $\theta^{(t)}, \omega^{(t)}$ to a subset of clients. Each client i will then update the local model parameters by ClientOpt with private data $\mathcal{D}_i$, and send back the updates for model parameters $\Delta_i(\theta^{(t)}), \Delta_i(\omega^{(t)})$. The model deltas from sampled clients are then aggregated and used by ServerOpt to get $\theta^{(t+1)}, \omega^{(t+1)}$ for the next round.

The generalized FedAvg algorithm can be extended for user-level DP by clipping the model deltas and adding noise proportional to the sensitivity [27,49]. We can use either independent Gaussian noise [49], or correlated noise that can achieve comparable privacy-utility trade-off without relying on the assumption of sampling [36]. The two variants are effectively applying DP-SGD [1] or DP-FTRL [36] as ServerOpt in the generalized FedAvg framework. Unlike the cross-device FL setting where sampling is extremely hard, it is possible to control user sampling in the datacenter and use DP-SGD. But DP-FTRL provides the possibility of handling the online setting where the user data are streamed instead of collected, and can be accounted for zCDP [7] reported by US census bureau [70]. A complete description of DP-FedAvg for training a backbone network to generate image embeddings is in Alg. 2 in Appendix B

In addition to the flexibility of generalized FedAvg for user-level DP, there are a few side effects of FedAvg that make it particularly effective for differentially private training. The model deltas are computed based on data for each user before clipping in DP, which can potentially reduce the bias introduced by clipping. As [16] suggested that averaging gradients from augmented data before clipping can improve training for example-level DP, model deltas from user data for user-level DP can be considered a natural extension to improve the bias-variance trade-off. The communication efficiency of FedAvg that leads to infrequent aggregation and model release is also desirable for DP training. The local model updates by private data on clients introduce no additional privacy cost, and only communication rounds between clients and server have to be accounted for DP. Though the theoretical advantages of FedAvg are only proved under certain assumptions [74,78], FedAvg with local updates can achieve communication efficiency and fast convergence in various practical applications [73].

2.3. Proposed DP-FedEmb method

While generalized DP-FedAvg can be applied to train a backbone network θ to generate embedding $f(\theta, x)$ from image x, there are challenges that significantly affect the efficiency and feasibility of the method. We propose DP-FedEmb with a few key features: virtual clients, partial aggregation, local fine-tuning, public pretraining, and parameter freezing. Details of DP-FedEmb are provided in Alg. 1.

Virtual clients. Data heterogeneity is one of the key problems in federated optimization [73]. When we train embedding models in the multi-class framework, the class space can be very large and each user may only observe a limited number of classes. In the extreme case, when training embedding models on facial images [61,66], each user may only have images for their own identity. This signif-

Algorithm 1: DP-FedEmb: learning embedding model θ with user-level DP

Input: SERVEROPT with learning rate α; CLIENTOPT with learning rate β_1 and β_2; clip norm γ and noise multiplier σ; (optional) pretrained model $\theta^{(0)}$

for round $t = 0, 1, \ldots, T-1$ **do**
  Sample a subset of users $\mathcal{U}^{(t)}$
  Partition users $\mathcal{U}^{(t)}$ to virtual clients $\mathcal{S}^{(t)}$
  for each virtual client $V \in \mathcal{S}^{(t)}$ **in parallel do**
    Initialize backbone $\theta_V^{(t,0)} = \theta^{(t)}$
    Randomly initialize head $\omega_V^{(t,0)}$
    for $k = 0, \ldots, K-1$ **do**
      Sample minibatch $B \subset \bigcup_{i \in V} \mathcal{D}_i$
      Compute gradients $\nabla_{\theta_V} \ell_B, \nabla_{\omega_V} \ell_B$, where $\ell_B = \mathbb{E}_{(x,y) \in B}\, \ell(\langle \omega_V,\, f(\theta_V, x) \rangle,\, y)$
      Update $\theta_V^{(t,k+1)}$ by CLIENTOPT, $\theta_V^{(t,k)}$, $\nabla_{\theta_V} \ell_B$, β_1
      Update $\omega_V^{(t,k+1)}$ by CLIENTOPT, $\omega_V^{(t,k)}$, $\nabla_{\omega_V} \ell_B$, β_2
    end
    Compute clipped model update $\Delta_V^{(t)} = \text{Clip}(\theta_V^{(t,K)} - \theta_V^{(t,0)},\, \gamma)$
  end
  Aggregate model updates $\Delta^{(t)} = \text{AddNoise}\big(\sum_{V \in \mathcal{S}^{(t)}} \Delta_V^{(t)}, \sigma\gamma\big) / |\mathcal{S}^{(t)}|$
  Update global backbone parameters $\theta^{(t+1)} = \text{SERVEROPT}(\theta^{(t)}, \Delta^{(t)}, \alpha)$
end

icantly limits the advantage of local updates due to client drift [39], and even with specialized regularization like [82], FedSGD [48] with frequent aggregation and model release has to be used instead of FedAvg. It is challenging to use some specialized techniques for handling data heterogeneity [73] in DP training. Instead, we propose a simple yet effective approach: randomly groups the data of sampled users into *virtual clients*.

Unlike the cross-device FL setting where the on-device data of users cannot be directly communicated, virtual clients are feasible for user data in the datacenter. It is important to guarantee that a user will not be included in two virtual clients in a single round for user-level DP, analogous to microbatches for DP-SGD and example-level DP [1, 51]. When the grouping of users for virtual clients is fixed in advance across rounds, the granularity of the DP definition can slightly change: the adjacent dataset for DP is based on virtual clients (a group of users) instead of a single user, which has conceptually stronger privacy guarantees. However, when virtual clients are randomly regrouped across rounds as in Alg. 1, we can only show user-level DP as discussed in Appendix C. Virtual clients also control the interpolation between federated training and centralized training: when all users are grouped into a single virtual client, federated training is equivalent to centralized training, which removes heterogeneity but is challenging for DP mechanism. Virtual clients are used for both baseline DP-FedAvg and the proposed DP-FedEmb method.

Partial aggregation and local fine-tuning. Another challenge is the number of parameters in DP training. A common backbone θ of ResNet-50 for 128 dimensional embedding vectors has 23.77 million parameters. However, the parameter size of head ω can linearly grow with the number of classes. Taking FaceNet [61, 66] as an example again, ω can easily grow to 1280 million for 10 million identities in real-world applications. Sampled softmax [35, 72] can be applied to improve training efficiency. However, as both backbone θ and head ω are shared among users and need to be privatized by adding noise during training, the combined parameter size of (θ, ω) will significantly affect the privacy utility trade-off, which cannot be mitigated by sampled softmax.

In DP-FedEmb, inspired by federated reconstruction [65] and DP personalization [34], we only aggregate and privatize the backbone network θ, which is used in inference and has fixed parameter size that does not grow with classes. A local head ω_V is randomly initialized and updated on each virtual client V. A fine-tuning approach is adopted for local updates, where different learning rates β_1, β_2 are used for the backbone θ_V and head ω_V, respectively. When combined with virtual clients, the partial aggregation and local fine-tuning approach can be interpreted in various ways: each virtual client is performing transfer learning given a shared backbone network for representation learning; the data of each class are their own positive samples as well as negative samples for other classes on the same virtual client; the size of local head ω_V is also significantly smaller than ω for all classes, which is effectively a user-based sampling for softmax.

Public pretraining. The parameter size of the backbone to be privatized can still be large after applying partial aggregation and local fine-tuning with virtual clients, *e.g.*, 23.77 million for ResNet-50. Inspired by recent research on applying DP-SGD for example-level DP on large language modeling [45, 81] and image classification [16, 42], we use a model pretrained on public images to initialize the DP training of the backbone network. There is a relatively clear distinction between the public and private domains for our task: we use public images collected from open webpages for pretraining, and then privately train on users' data collected in a datacenter.

Parameter freezing. Neural networks are known to be

overparameterized, and not all weights are equally important [26, 84]. Freezing some parameters to be non-trainable has been shown to be effective when the privacy budget is small [64], especially when combined with public pretraining for large models [16, 81]. For backbone convolutional neural networks with normalization layers, we experiment with training parameters with all normalization layers, and some of the convolutional kernels. However, freezing is found to be less efficient in our setting that performs representation learning, instead of classification, for a moderate size model in the high-utility-moderate-noise regime.

DP mechanism and hyperparameters. Similar to generalized DP-FedAvg, we perform clipping for model deltas and add noise for aggregated updates. The clip norm γ is estimated by adaptive clipping [3] in the parameter tuning stage. For DP-FedEmb, we perform extensive studies on several configurations in Sec. 3. Differentially private hyperparameter tuning [55] is a topic out of the scope of this paper, and automating hyperparameter tuning is an important future work. Either independent Gaussian noise like DP-SGD [49] or tree-based correlated noise like DP-FTRL [36] can be added. Under the same noise multiplier, DP-FedEmb will achieve the same privacy bound as DP-FedAvg with virtual clients, while utility can be improved for training a backbone network with a large head.

3. Experiments

We conduct experiments to train image-to-embedding backbone networks with user-level DP. We use the DigiFace dataset [6] of synthetic faces based on ethical and responsible development considerations, and verified that the conclusions on DigiFace are very similar to conclusions generated from experiments on natural facial images. We randomly split the DigiFace dataset of $110K$ identities and $1.22M$ into subsets of $98.96K$ identities with $1.10M$ images for training, 5443 identities with $58.24K$ images for validation, and 5598 identities with $60.82K$ images for testing. We extensively use a smaller training set, DigiFace10K, which contains the 9047 training identities of 72 images sampled from the DigiFace training set. We also run experiments on public datasets of natural images: EMNIST, Google Landmarks Dataset (GLD) and iNaturalist (iNat) dataset. These datasets are summarized in Tab. 3.

In our setting, each user holds only the images of their own identities, i.e., user-level DP is also identity-level DP. We use ResNet-50 [31] and MobileNetV2 [32, 60, 73] as backbone networks, replace batch normalization [33] with group normalization [80], and use a multi-class framework with a large softmax head to train the backbone. The dimension of the embeddings are 128 for all experiments.

We evaluate the performance of the backbone network based on predicting identity matches from the distance between two image embeddings. By varying a threshold on the pairwise similarity, a recall versus false accept rate (FAR) curve on the test data can be generated for a trained model. A scalar value of recall@FAR=$1\mathrm{e}{-3}$ is often reported. The privacy guarantees are computed by either using Renyi differential privacy (RDP) [53] and converting to (ϵ, δ)-DP by [8], or DP-FTRL accounting without restart [36]. More discussion on privacy accounting can be found in Appendix C. We aim for single-digit ϵ when δ is small, and sometimes relax to $\epsilon \sim 20$ as we use the stronger substitute-one DP definition [56].

We compare the proposed DP-FedEmb with non-private oracle performance of centralized training, and baseline methods DP-FedAvg. We tune the learning rate with learning rate scheduling for standard centralized training. The centralized baseline is provided as an oracle for non-private training performance. We exclude tricks like data augmentation for either centralized or federated training as the goal is not achieving state-of-the-art performance. Virtual clients are used to improve DP-FedAvg performance, and the same tuning strategy is applied for DP-FedEmb and DP-FedAvg. In most of the experiments, unless otherwise specified, we fix the hyperparameters in the federated setting and only tune the learning rates; the backbone networks are pretrained on classifying the 1000 classes of ImageNet [59]; both CLIENTOPT and SERVEROPT are SGD optimizers with momentum 0.9; and more details are provided in Sec. 3.3. Code is released at https://github.com/google-research/federated/tree/master/dp_visual_embeddings.

3.1. Privacy-utility-computation trade-off

We study the privacy-utility-computation trade-offs of training ResNet-50 on DigiFace10K in Fig. 2 and Fig. 6. Figure 2a shows the privacy-utility trade-off. Without adding noise, the federated algorithms (DP-)FedAvg and (DP-)FedEmb can achieve even better results than the non-private centralized baseline, which is consistent with recent empirical and theoretical justifications that FedAvg is more accurate when learning representations [14, 15]. When the same noise multiplier is used, *i.e.*, under same privacy budget, DP-FedEmb outperforms DP-FedAvg; and the margin increases when increasing the noise. We can observe the advantage of DP-FedEmb over DP-FedAvg even if we only have a small number of identities in DigiFace: the size of the head is $9047 \times 128 \sim 1.16M$, only 4.6% of the backbone ResNet-50 with $23.77M$ parameters. For large scale data with 10 million identities, the size of the head can grow to $1280M$, which is much larger than the backbone networks, and DP-FedAvg can easily fail in such settings.

It can be difficult to achieve a strong formal differential privacy bound without significantly hurting utility for DigiFace10K, which only has a small number of total users. We consider the practical setting of more available users, and

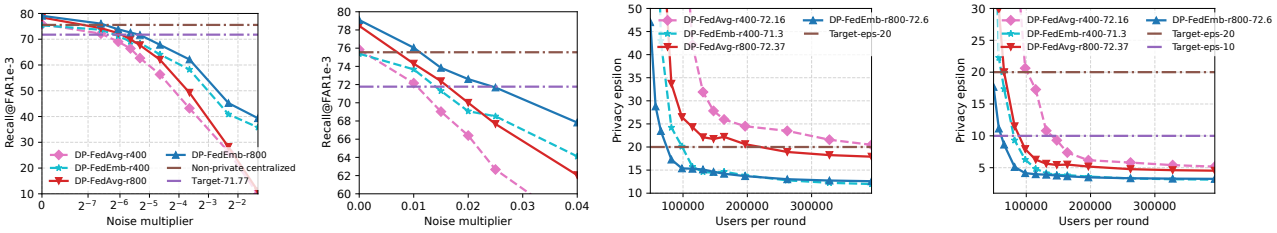

Figure 2. (a) Recall@FAR=1e−3 on DigiFace validation set under different noise multiplier; (b) zoom in the high utility regime in (a); (c) and (d) privacy-computation trade-off by extrapolating based on 3M and 10M total users, respectively. "r400" and "r800" represent the result at 400 or 800 training rounds; target 71.77% is 95% of centralized non-private recall@FAR=1e−3 at 75.55%.

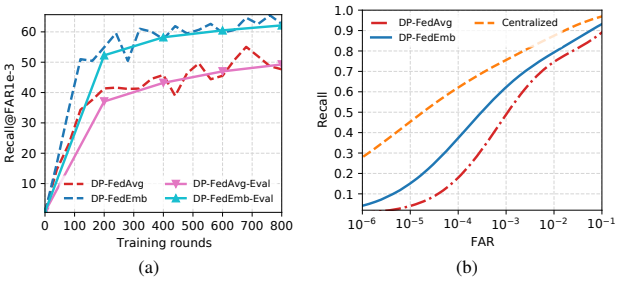

Figure 3. DP-FedEmb and DP-FedAvg on DigiFace10K under the same privacy budget with noise multiplier 0.08, and additional oracle baseline of centralized training w/o DP: (a) recall@FAR=1e−3 on validation dataset during training; dashed lines are approximation by sampling a subset of validation users; (b) ROC curve of trained model on test set with log scale x-axis.

study the privacy-computation trade-off in Figs. 2c, 2d, 6b and 6c based on extrapolation. A key hypothesis following [36,49] is used: utility (Recall@FAR) is non-decreasing when simultaneously increasing the number of clients per round and noise multiplier. The hypothesis is based on the fact that the signal-to-noise ratio is non-decreasing when linearly increasing the number of clients per round and noise multiplier, and has been verified in practice [50,57].

We first choose the noise multiplier that is within 5% of recall@FAR=1e−3 compared to centralized non-private training in Figs. 2a and 6a, based on simulation that samples 64 virtual clients that each have 32 users per round: recall@FAR=1e−3 is 72.16 for running DP-FedAvg with 0.01 noise multiplier for 400 rounds, 72.37 for running DP-FedAvg with 0.015 noise multiplier for 800 rounds, 71.3 for running DP-FedEmb with 0.015 noise multiplier for 400 rounds, 72.6 for running DP-FedEmb with 0.02 noise multiplier for 800 rounds, and 71.85 for running DP-FTRL-FedEmb with 0.26 noise multiplier for 800 rounds. Then we linearly increase the number of users sampled and use the increased noise multiplier in RDP accounting to compute privacy bound ϵ given $\delta = 10^{-7}$ to generate Figs. 2c, 2d and 6b, and compute zCDP for Fig. 6c. Comparing curves of r400 and r800, training longer with larger noise is more effective than training shorter with smaller noise. Figure 2c suggests $\sim 98K$ users per round is enough for DP-FedEmb to achieve single-digit ϵ if $3M$ users are available, while $\sim 57K$ users per round are needed if $10M$ users are available in Fig. 2d. $98K$ users is $48\times$ the number of users per round in our current simulation, which can be achieved by training with $8\times$ computing resources for $6\times$ longer. In Fig. 6b, there is a crossover point when using DP-FTRL versus DP-SGD for DP-FedEmb, and DP-FTRL is more effective for relatively large privacy ϵ. Figure 6c shows that DP-FTRL-FedEmb can achieve zCDP smaller than 2.6, as used by US Census Bureau [70], when $8\times$ users per round and $10M$ total users are available.

3.2. Model evaluation

In Tab. 1, we summarize the quantitative results from the privacy-utility-computation trade-off analysis in Sec. 3.1. Each experiment runs three times to compute the mean and standard deviation. For similar recall@FAR=1e−3 on the DigiFace validation set, DP-FedEmb achieves stronger privacy guarantee than baseline DP-FedAvg, and the advantage of DP-FedEmb is expected to be more pronounced if a head for larger $10M$ identities is used for training. When $10M$ users are available and $64\times$ users per round in training, privacy $\epsilon = 3.90$ of single digit and zCDP$= 1.28$ smaller than 2.6 can be achieved when recall@FAR=1e−3 is within a 5% drop compared with non-private centralized training. In addition to validation performance, the private models also perform well on the left-out test dataset. Figure 3 presents the training curves and ROC curves for comparing DP-FedEmb and DP-FedAvg under the same privacy budget. DP-FedEmb outperforms DP-FedAvg in all training rounds, and trains a stronger private model with better recall at different false accept rates.

3.3. Ablation study

We primarily use MobileNetV2 for ablation studies on DigiFace10K for two reasons: MobileNetV2 is smaller and faster for training in experiments; to test the generalization of DP-FedEmb and avoid overfitting on ResNet-50.

Algorithm	Hyperparameters		Privacy (10M users)		Recall@FAR=1e−3	
	Noise	SerLR	RDP-ϵ	zCDP	Validation	Test
Centralized	0	0.05	∞	∞	75.55 ± 0.05	75.53 ± 0.12
DP-FedAvg	0.015×64	0.5	5.62	-	72.57 ± 0.12	72.37 ± 0.09
DP-FedEmb	0.02×64	0.2	3.90	-	72.63 ± 0.05	72.37 ± 0.09
DP-FTRL-FedEmb	0.26×64	0.2	9.67	1.28	72.2 ± 0.29	71.87 ± 0.26

Table 1. Quantitative results of privacy and utility on the DigiFace10K dataset. The client learning rate and clip norm for federated algorithms are 0.002 and 0.6, respectively; $\delta = 10^{-7}$ for privacy guarantees. Centralized training has a standard learning rate scheduling, while tricks like data augmentation are excluded for all methods. The privacy guarantees are extrapolated based on $10M$ users and $131K$ users per round. A strong privacy guarantee can be achieved within a 5% drop on recall@FAR=1e−3.

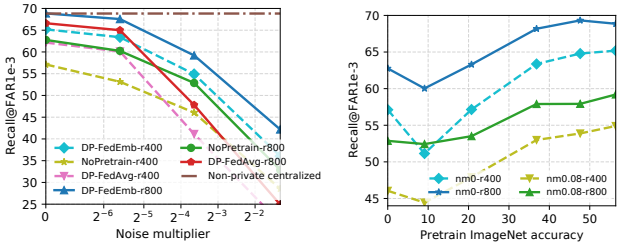

Figure 4. Recall@FAR=1e−3 on DigiFace validation set when training a MobileNet (a) with DP-FedEmb w/ pretraining, and baselines of DP-FedEmb w/o pretraining and DP-FedAvg; (b) with different pretrained models.

Parameter freezing and public pretraining. For similar parameter size, MobileNetV2 outperforms ResNet-50 with frozen parameters, and Fig. 4a shows the privacy-utility trade-off. Recall@FAR=1e−3 of DP-FedEmb-r800 on MobileNetV2 only drops from 68.86% to 67.56% when 0.02 noise is added, while ResNet-50 drops from 79.09% to 72.6%. However, ResNet-50 still outperforms MobileNetV2 by a large margin in the high utility regime. DP-FedEmb consistently outperforms DP-FedAvg when noises are added. In Fig. 4b, though the private fine-tuning utility is not linearly increasing with pretraining accuracy, there seems to be a general positive correlation: better pretrained models can lead to better private models except for one outlier where a inferior pretrained model causes difficulty in training. More discussion on parameter freezing and public pretraining is provided in Appendix D.4.

Federated settings. In the above experiments, we fix important hyperparameters for the federated setting: users per virtual client is 32, virtual clients per round is 64, examples per client is capped at 2048, the head learning rate (LR) scale β_2/β_1 is 100, the buffer size for data shuffling on clients is 2048, and the batch size for local SGD is 32. In Fig. 9, instead of tuning these hyperparameters in advance, we conduct a study on these hyperparameters to understand DP-FedEmb. Among these hyperparameters, Figs. 9a and 9d suggest virtual clients and head LR scale are particularly important for DP-FedEmb to be on par with non-private centralized training, which is an important contribution of this work. Figures 9a to 9c suggest users per virtual client, clients per round, and examples per client only need to be large enough under privacy consideration and computation resources for the best practice. The head LR scale has a large tuning range between 50 and 500 in Fig. 9d. Recall@FAR=1e−3 is not sensitive to shuffle buffer size in Fig. 9e. The model utility can be potentially improved if we further tune the client batch size as suggested by Fig. 9f. We fixed the learning rate and other hyperparameters while varying one of the hyperparameters in the ablation study. How to automate tuning, especially tuning with differential privacy guarantees, is an important future work.

Learning rate (LR). For experiments in Secs. 3.1 and 3.2, server and client learning rates are first tuned for non-private federated training with adaptive clipping of quantile 0.5 [3]. Then server learning rate is tuned when adding noise for private tuning, while estimated clip norm and client learning rate are fixed. The tuning range of learning rates are $\{1, 2, 5\} * 10^{-n}$. Figures 5a and 5b suggest the optimal learning rates are similar for training 400 rounds or 800 rounds in non-private training. We then fix the client learning rate to be 0.002 and use the estimated clip norm 0.6 for MobileNetV2 in Fig. 5c, and observe the fixed clip results are very similar to adaptive clipping results. The best server learning rate for private training with noise can be smaller than non-private training, where the difference is even more notable for larger model ResNet-50 in Fig. 5d.

Variants of DP-FedEmb. We use local fine-tuning with different learning rates β_1, β_2 to update backbone and head parameters. An alternative is to reconstruct the head first before fine-tuning the backbone [41, 65]. We empirically find that head reconstruction can only achieve similar performance as the proposed fine-tuning when there are same or more number of updates on the backbone network, and hence use local fine-tuning for efficiency. It is also possible to use binary or triplet loss within a virtual clients. In our preliminary results, they achieve inferior results compared to DP-FedEmb that uses multi-class cross-entropy loss. We leave other improvement like arcface loss [17] as future work.

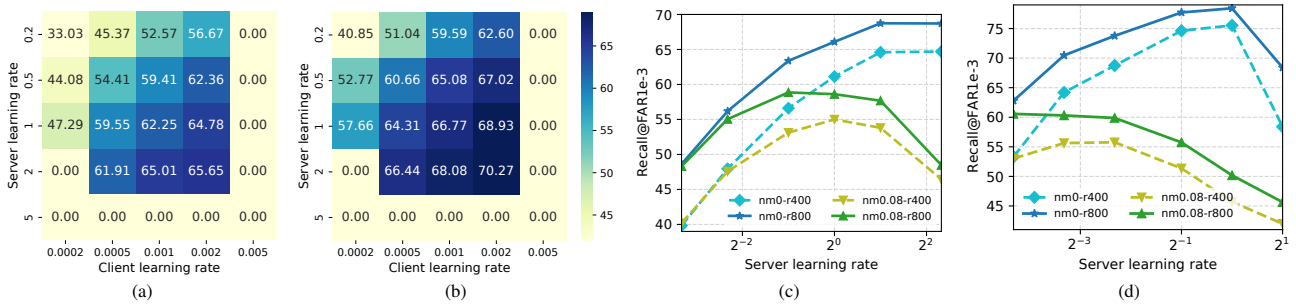

Figure 5. Study for learning rates. (a) and (b) recall@FAR=1e−3 after training MobileNetV2 using DP-FedEmb with adaptive clipping [3] and 0 noise for 400 rounds and 800 rounds, respectively; (c) and (d) varying server learning rate with fixed client learning rate and clip norm for MobileNetV2 and ResNet-50.

Dataset	Algorithm	Recall@FAR=1e−3 / 0.1	
		Approx	AllPair
DigiFace	DP-FedEmb	-	19.76 ± 0.43
	DP-FedAvg	-	13.38 ± 0.27
EMNIST	DP-FedEmb	10.64 ± 0.5	10.47 ± 0.5
	DP-FedAvg	9.78 ± 0.44	9.67 ± 0.41
GLD	DP-FedEmb	26.18 ± 0.44	27.07 ± 0.04
	DP-FedAvg	24.48 ± 1.0	26.27 ± 0.17
iNat	DP-FedEmb	40.49 ± 1.08	40.93 ± 0.94
	DP-FedAvg	29.57 ± 0.78	29.6 ± 0.65

Table 2. The utility under same privacy budget for DigiFace [6], EMNIST [67], GLD [68] and iNat [69] datasets.

3.4. Additional results

We run additional experiments of ResNet-50 on the larger DigiFace of $98.96K$ users. Because a lot of users in DigiFace have only 5 images, we set each virtual client to contain 64 users, and use 3072 samples per virtual client. We sample 128 virtual clients per round, and a relatively large noise 1.39 in RDP accounting can achieve $\epsilon = 24.86, \delta = 10^{-5}$ without extrapolation. The utility measured by recall@FAR=1e−3 is shown in Tab. 2. Though the utilities of both methods are significantly degraded by the large noise, DP-FedEmb is much better than the DP-FedAvg baseline because the noise is only added to backbone of $\sim 2.4M$ parameters instead of backbone plus head of $\sim 15M$ parameters. The full table including EMNIST [67] results for reproducibility and hyperparameter choices is provided in Tab. 4 in Appendix D.

In Tab. 2, we also conduct experiments with MobileNetV2 on Google Landmark Dataset (GLD) [32, 68, 77] and iNaturalist (iNat) dataset [32, 47, 69] to demonstrate the generalization of DP-FedEmb. We use a public model pretrained on ImageNet, and report extra approximate recall@FAR by computing pairwise similarity for minibatches, which is easy to reproduce and consistent with the all pair recall@FAR. We fix the hyperparameters for the federated settings, and compare the performance of DP-FedEmb and DP-FedAvg under the same privacy budget (noise multiplier). Since each user already has multiple classes in GLD, we use a smaller number of users, 8, in each virtual client. We also use a smaller number of virtual clients per round, 32, for fast experiments and strong sampling effect. Recall@FAR=1e−3 of DP-FedEmb and DP-FedAvg with small noise multiplier 0.02 on GLD outperforms centralized training. For iNat, we use an even smaller four users per virtual client and train for only 400 rounds, and use a relatively large noise multiplier 0.5 to get a single-digit $\epsilon = 16.06$ DP guarantee for 9275 users. We report recall@FAR=0.1 instead of recall@FAR=1e−3 for the challenging iNat task. In all experiments, DP-FedEmb consistently outperforms DP-FedAvg.

4. Conclusion

This paper presented DP-FedEmb for training embedding models with user-level differential privacy. We show how practical utility with strong privacy guarantees can be achieved in the data center, thanks to key algorithm design choices around the construction of virtual clients and in the selection of what information is shared among users. Our experiments validate this improves the privacy utility tradeoff upon vanilla DP-FedAvg for supervised representation learning. Though strong formal DP bounds at practical levels of utility could only be achieved when millions of users participate in training, DP-FedEmb is designed to be exceptionally scalable when model size and class space increases with number of users. DP-FedEmb can also be applied to decentralized FL when each real client contains multiple classes, possibly reducing the necessity of virtual clients. Finally, DP is a worst-case guarantee that can be improved by both algorithmic design and advanced accounting methods; the non-negligible noise we added for the small scale datasets in experiments are ready to be empirically audited for privacy.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. 1, 3, 4, 14

[2] Kareem Amin, Alex Kulesza, Andres Munoz, and Sergei Vassilvtiskii. Bounding user contributions: A bias-variance trade-off in differential privacy. In *International Conference on Machine Learning*, pages 263–271. PMLR, 2019. 13

[3] Galen Andrew, Om Thakkar, H Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 5, 7, 8, 13, 17

[4] Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private bert. *arXiv preprint arXiv:2108.01624*, 2021. 1

[5] Apple Privacy Team. Learning with privacy at scale. Available at https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html, 2017. 1

[6] Gwangbin Bae, Martin de La Gorce, Tadas Baltrušaitis, Charlie Hewitt, Dong Chen, Julien Valentin, Roberto Cipolla, and Jingjing Shen. Digiface-1m: 1 million digital face images for face recognition. In *2023 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2023. 5, 8, 17

[7] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016. 3, 13

[8] Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems*, 33:15676–15688, 2020. 5

[9] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022. 1

[10] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, 2019. 1

[11] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021. 1

[12] Yannis Cattan, Christopher A Choquette-Choo, Nicolas Papernot, and Abhradeep Thakurta. Fine-tuning with differential privacy necessitates an additional hyperparameter search. *arXiv preprint arXiv:2210.02156*, 2022. 16

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 13

[14] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Fedavg with fine tuning: Local updates lead to representation learning. *arXiv e-prints*, pages arXiv–2205, 2022. 5

[15] Liam Collins, Aryan Mokhtari, Sewoong Oh, and Sanjay Shakkottai. Maml and anil provably learn representations. *arXiv preprint arXiv:2202.03483*, 2022. 5

[16] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022. 1, 3, 4, 5

[17] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 7, 13

[18] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. *Advances in Neural Information Processing Systems*, 30, 2017. 1

[19] Xin Dong, Sai Qian Zhang, Ang Li, and HT Kung. Spherefed: Hyperspherical federated learning. *arXiv preprint arXiv:2207.09413*, 2022. 13

[20] Vadym Doroshenko, Badih Ghazi, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. Connect the dots: Tighter discrete approximations of privacy loss distributions. *arXiv preprint arXiv:2207.04380*, 2022. 13, 14

[21] DP Team. Google's differential privacy libraries., 2020. https://github.com/google/differential-privacy. 14

[22] Cynthia Dwork. Differential privacy in new settings. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 174–183. SIAM, 2010. 1, 13

[23] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006. 1, 13

[24] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. 14

[25] Alessandro Epasto, Mohammad Mahdian, Jieming Mao, Vahab Mirrokni, and Lijie Ren. Smoothly bounding user contributions in differential privacy. *Advances in Neural Information Processing Systems*, 33:13999–14010, 2020. 13

[26] Jonathan Frankle, David J. Schwab, and Ari S. Morcos. Training batchnorm and only batchnorm: On the expressive power of random features in CNNs, 2021. 5, 16

[27] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017. 3, 13

[28] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. User-level private learning via correlated sampling. *arXiv preprint arXiv:2110.11208*, 2021. 13

[29] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch,

Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 13

[30] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 13

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[32] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *European Conference on Computer Vision*, pages 76–92. Springer, 2020. 5, 8

[33] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 5

[34] Prateek Jain, John Rush, Adam Smith, Shuang Song, and Abhradeep Guha Thakurta. Differentially private model personalization. *Advances in Neural Information Processing Systems*, 34:29723–29735, 2021. 4, 13

[35] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*, 2014. 3, 4

[36] Peter Kairouz, Brendan Mcmahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning (ICML)*, pages 5213–5225, 2021. 1, 2, 3, 5, 6, 13, 14, 15, 16

[37] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. 13

[38] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015. 1

[39] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 4

[40] Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight approximate differential privacy for discrete-valued mechanisms using fft, 2020. 13, 14

[41] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022. 7

[42] Alexey Kurakin, Steve Chien, Shuang Song, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328*, 2022. 1, 4

[43] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 18

[44] Daniel Levy, Ziteng Sun, Kareem Amin, Satyen Kale, Alex Kulesza, Mehryar Mohri, and Ananda Theertha Suresh. Learning with user-level privacy. *Advances in Neural Information Processing Systems*, 34:12466–12479, 2021. 13

[45] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021. 1, 4

[46] Yuhan Liu, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Michael Riley. Learning discrete distributions: user vs item-level privacy. *Advances in Neural Information Processing Systems*, 33:20965–20976, 2020. 13

[47] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 8

[48] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282. PMLR, 2017. 3, 4

[49] Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2, 3, 5, 6, 13, 14

[50] Brendan McMahan and Abhradeep Thakurta. Federated learning with formal differential privacy guarantees, 2022. 1, 6

[51] H Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. A general approach to adding differential privacy to iterative training procedures. *arXiv preprint arXiv:1812.06210*, 2018. 4, 13, 14

[52] Qiang Meng, Feng Zhou, Hainan Ren, Tianshu Feng, Guochao Liu, and Yuanqing Lin. Improving federated learning face recognition via privacy-agnostic clusters. *arXiv preprint arXiv:2201.12467*, 2022. 13

[53] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017. 5, 13

[54] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020. 13

[55] Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. *arXiv preprint arXiv:2110.03620*, 2021. 5

[56] Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *arXiv preprint arXiv:2303.00654*, 2023. 5, 15

[57] Swaroop Ramaswamy, Om Thakkar, Rajiv Mathews, Galen Andrew, H Brendan McMahan, and Françoise Beaufays. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*, 2020. 1, 2, 6, 13

[58] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. 3

[59] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5

[60] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 5

[61] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 1, 3, 4, 13

[62] Zebang Shen, Jiayuan Ye, Anmin Kang, Hamed Hassani, and Reza Shokri. Share your representation only: Guaranteed improvement of the privacy-utility tradeoff in federated learning. In *The Eleventh International Conference on Learning Representations*, 2023. 13

[63] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017. 1

[64] Hakim Sidahmed, Zheng Xu, Ankush Garg, Yuan Cao, and Mingqing Chen. Efficient and private federated learning with partially trainable networks. *arXiv preprint arXiv:2110.03450*, 2021. 5, 16

[65] Karan Singhal, Hakim Sidahmed, Zachary Garrett, Shanshan Wu, Keith Rush, and Sushant Prakash. Federated reconstruction: Partially local federated learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 4, 7, 13

[66] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. 3, 4, 13

[67] TFF Authors. TensorFlow Federated EMNIST dataset, 2022. https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/emnist. 8, 15, 17

[68] TFF Authors. TensorFlow Federated Google Landmark v2 dataset, 2022. https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/gldv2. 8, 15, 17

[69] TFF Authors. TensorFlow Federated iNaturalist dataset, 2022. https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/inaturalist. 8, 15, 17

[70] US Census Bureau. Disclosure avoidance for the 2020 census: An introduction, 2021. 1, 3, 6

[71] Salil Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer, 2017. 14

[72] Sagar M Waghmare, Hang Qi, Huizhong Chen, Mikhail Sirotenko, and Tomer Meron. Efficient image representation learning with federated sampled softmax. *arXiv preprint arXiv:2203.04888*, 2022. 4, 13

[73] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Blaise Aguera y Arcas, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, et al. A field guide to federated optimization. *arXiv:2107.06917*, 2021. 3, 4, 5, 13, 14

[74] Jianyu Wang, Rudrajit Das, Gauri Joshi, Satyen Kale, Zheng Xu, and Tong Zhang. On the unreasonable effectiveness of federated averaging with heterogeneous data. *arXiv preprint arXiv:2206.04723*, 2022. 3

[75] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009. 1

[76] Yandong Wen, Weiyang Liu, Adrian Weller, Bhiksha Raj, and Rita Singh. Sphereface2: Binary classification is all you need for deep face recognition. *arXiv preprint arXiv:2108.01513*, 2021. 13

[77] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584, 2020. 1, 8, 13

[78] Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local sgd better than minibatch sgd? In *International Conference on Machine Learning*, pages 10334–10343. PMLR, 2020. 3

[79] Lin Wu, Chunhua Shen, and Anton van den Hengel. Personnet: Person re-identification with deep convolutional neural networks. *arXiv preprint arXiv:1601.07255*, 2016. 13

[80] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 5

[81] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021. 1, 4, 5

[82] Felix Yu, Ankit Singh Rawat, Aditya Menon, and Sanjiv Kumar. Federated learning with only positive labels. In *International Conference on Machine Learning*, pages 10946–10956. PMLR, 2020. 4

[83] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 13

[84] Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal?, 2019. 5

A. Related work

Differential Privacy (DP), introduced by [23], is a formal mathematical notion of privacy protection. Formally, two datasets D and D' are said to be neighboring if they differ at most by one entry. A randomized mechanism $\mathcal{A}$ is said to be (ϵ, δ)-differentially private if $\mathbb{P}(\mathcal{A}(D) \in S) \leq e^{\epsilon}\mathbb{P}(\mathcal{A}(D') \in S) + \delta$, for all neighboring D and D'. We refer to this original definition as *example-level DP*, and several variations have been proposed, including Renyi DP (RDP) [53], Privacy Loss Distribution (PLD) [20, 40], and concentrated DP (zCDP) [7].

A formal definition of *user-level DP* is introduced in [22], where the unit of privacy protection is extended from a single entry (in the original example-level DP) to every entry that belongs to the same user. The dependence of the utility on the number of users and the number of samples per user has been studied for various tasks: empirical risk minimization and mean estimation [44], estimating discrete distributions [46], and PAC learning [28]. Extensions to heterogeneous users in sample size have been studied in [2, 25]. User-level DP is particularly useful in federated learning, where the natural unit of privacy is a user (i.e., a client) [27, 51] and standard private training algorithms respect user-level DP [49]. The privacy-utility trade-off for user-level DP is investigated in [34, 62] where feature extractors are federated trained and classifiers are personalized.

Representation learning and metric learning are active research directions in computer vision. Recently, a lot of progress has been made towards representation learning with large-scale unsupervised data [13, 29, 30, 83]. However, we consider representation learning with supervised data as it is widely used for downstream tasks like face recognition and clustering [61, 66], person re-identification [79], and landmark recognition [77], which can significantly benefit from privacy protection. Two technical frameworks are often used in the supervised representation learning tasks due to the large output space: triplet and its variants with hard negative mining [61], and multi-class training with proxy weights [17, 66, 76]. We propose DP-FedEmb based on the multi-class approach for the following reasons: the two approaches can achieve similar performance when trained with large-scale data [54]; multi-class training is simple and flexible, and can be more efficient when less data are touched every iteration; and negative sampling can trigger non-trivial computational and privacy cost. To the best of our knowledge, differentially private models have not been trained for large-scale representation learning.

Federated learning is an active research topic primarily designed for learning from decentralized data [37, 73]. We propose DP-FedEmb based on federated learning algorithms as they are suitable for user-level DP. User-level DP can be achieved in federated learning by variants of DP-FedAvg [3, 27, 36, 49, 57]; the previous works train relatively small models for image classification and language modeling tasks. Closest to DP-FedEmb is [65], which proposed federated reconstruction that performs partially local training for personalization; [19] fixed the softmax and train the feature extractor before calibrating for image classification tasks; [72] modified sampled softmax for large output space in federated learning. These works are designed for learning with decentralized data, and do not consider differential privacy. [52] use differential privacy on proxy vectors to mitigate the privacy concerns when clients exchange weight vectors of identities for federated training, which is different from our motivation of training a differentially private model that will not memorize a specific user's data.

B. DP-FedAvg algorithm

Algorithm 2: Learning embedding model θ with generalized DP-FedAvg [49, 73].

Input: SERVEROPT with learning rate α;
CLIENTOPT with learning rate β;
clip norm γ and noise multiplier σ;
(optional) pretrained model $\theta^{(0)}$

for round $t = 0, 1, \ldots, T-1$ **do**
  Sample a subset $\mathcal{U}^{(t)}$ of users
  for each client $i \in \mathcal{U}^{(t)}$ **in parallel do**
    Initialize parameters $(\theta_i, \omega_i)^{(t,0)} = (\theta, \omega)^{(t)}$
    for $k = 0, \ldots, K-1$ **do**
      Sample minibatch $B \subset \mathcal{D}_i$
      Compute gradients $\nabla_{(\theta_i,\omega_i)} \ell_B$, where $\ell_B = \mathbb{E}_{(x,y)\in B}\, \ell(\langle \omega_i,\ f(\theta_i, x)\rangle,\ y)$
      Update $(\theta_i,\ \omega_i)^{(t,k+1)}$ by CLIENTOPT, $(\theta_i,\ \omega_i)^{(t,k)}$, $\nabla_{(\theta_i,\omega_i)} \ell_B$, β
    end
    Compute clipped model update $\Delta_i^{(t)} = \text{Clip}((\theta_i, \omega_i)^{(t,K)} - (\theta_i, \omega_i)^{(t,0)},\ \gamma)$
  end
  Aggregate model updates $\Delta^{(t)} = \text{AddNoise}\left(\sum_{i\in\mathcal{U}^{(t)}} \Delta_i^{(t)}, \sigma\gamma\right) / |\mathcal{U}^{(t)}|$
  Update global parameters $(\theta, \omega)^{(t+1)} = \text{SERVEROPT}((\theta, \omega)^{(t)}, \Delta^{(t)},\ \alpha)$
end

C. Remark on privacy accounting

The accounting and differential privacy definition of DP-FedEmb depends on the DP mechanism applied in noise addition. If independent Gaussian noise similar to DP-SGD [1, 49] is used in Alg. 1, we adopt the substitute-one notation for DP definition [24, 71] and leveraged privacy amplification for uniform sampling. If tree-based noise similar to DP-FTRL [36] is used in Alg. 1, we adopt the add-or-remove with special element notation in [36] for DP definition. We use uniformly sample users in each round of Algs. 1 and 2. Though DP-FTRL [36] assumes a different data streaming pattern, the practical effect is likely negligible. We use implementation in [21] for RDP accounting for DP-FedEmb, and use the open-sourced implementation by [36] for DP-FTRL-FedEmb. While privacy loss distribution (PLD) [20, 40] accounting can be tighter than RDP accounting, the current implementation [21] does not support substitute-one and uniform sampling. Future improvement on privacy accounting can help further improve the guarantees obtained in our experiments.

Accounting for virtual clients. We provide more discussion on the accounting of virtual clients proposed in this paper. We consider user-level DP where datasets adjacency in the DP definition is based on changing all data of a single user, which is one kind of group-level DP stronger than example-level DP. Under virtual clients described in Alg. 1 and Sec. 2.3, though we cannot formally show the (stronger) "virtual client"-level DP due to the randomized grouping of clients, we can show user-level DP by the following key idea: when one user is replaced in one round, at most one virtual client is affected; the sensitivity is controlled by clipping the updates from virtual clients; noise is added proportional to clip norm, and hence proportional to sensitivity; a formal guarantee for Gaussian mechanism can be shown for noise proportional to sensitivity. The same logic can be used to prove for microbatches [51] in DP-SGD for example-level DP is analogously to virtual clients in DP-FedAvg and DP-FedEmb for user-level DP.

There is indeed a nuance in applying virtual clients in practice. Although add-or-remove-one neighboring relationship is popular in DP definition, it can be challenging in virtual clients. Following a worst case reasoning, adding or removing one user in a virtual client can *arbitrarily* change the signal of the virtual client. Even though the norm of virtual clients is clipped, the sensitivity of the mechanism may be doubled. Trying to adopt the add-or-remove-one DP definition will cause the sensitivity of virtual clients (of more than one user) to double compared to without grouping users by virtual clients. However, the sensitivity is consistent between virtual clients (of more than one user) and without virtual clients for the substitute-one DP definition. Another nuance comes from the amplification by sampling used to achieve strong privacy guarantees. For add-or-remove-one DP definition, Poisson sampling [1] is assumed for privacy accounting but not enforced in simulation.

By adopting the substitute-one DP definition, our accounting assumption and simulation consistently use uniform sampling. Conceptually, the substitute-one DP guarantees can be twice as strong as add-or-remove-one DP guarantees [56, Section 2.1.1]. Though the DP guarantees of different DP definition is not directly comparable, we can potentially relax the target DP guarantees of $\epsilon \leq 10$ for add-or-remove-one DP to $\epsilon \leq 20$ for substitute-one DP in practice [56, Section 5.2.2]. Hence we use substitute-one DP definition in this paper [1]. Note that all the nuances discussed also apply to microbatches and DP-SGD, which is often overlooked in the past.

D. Additional experimental details

D.1. Dataset statistics

Dataset	Train			Validation			Test		
	Users	Classes	Images	Users	Classes	Images	Users	Classes	Images
DigiFace	$98.96K$	$98.96K$	$1.10M$	5443	5443	$58.24K$	5598	5598	$60.82K$
DigiFace10K	9047	9047	$0.65M$	-			-		
EMNIST	6800	36	$0.58M$	3400	26	$17.68K$	-		
GLD	1262	2028	$0.18M$	-	2028	$19.53K$	-		
iNat	9275	1203	$0.12M$	-	1203	$35.64K$	-		

Table 3. The statistics of simulation datasets. The Google Landmarks Dataset (GLD) and iNaturalist (iNat) dataset are preprocessed by Tensorflow Federated [68, 69]. For the training of EMNIST, we use images of class $0-35$ in the union of the 3400 train and test clients in Tensorflow Federated dataset [67]; and use images of class $36-62$ in the 3400 test clients for validation. The shape of an image is 112×112 for DigiFace/DigiFace10K, 224×224 for GLD and iNat, and 28×28 for EMNIST.

D.2. DP-FTRL-FedEmb curves

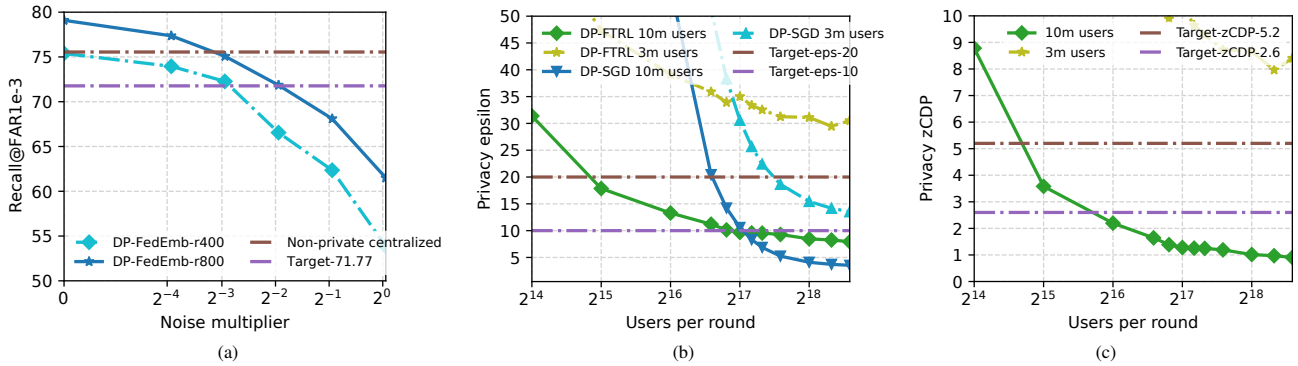

Figure 6. (a) Recall@FAR=1e−3 on DigiFace validation set under different noise multiplier when use DP-FTRL [36] for DP-FedEmb. (b) and (c) privacy-computation trade-off by extrapolating based on $3M$ and $10M$ total users; ϵ by RDP accounting for DP-FedEmb and DP-FTRL-FedEmb, and zCDP for DP-FTRL-FedEmb are reported, respectively.

D.3. Training and ROC curves

D.4. Parameter freezing and public pretraining

Parameter freezing. In Figs. 2a and 6a, we notice that the utility measured by recall@FAR=1e−3 can decrease faster when increasing the noise multiplier than observed for models in previous work [36]. After using DP-FedEmb to reduce the size of parameters to be noised, the ResNet-50 backbone still has $\sim 24M$ parameters, which is $6\times$ the language model

[1] In a previous version of the draft, we use privacy accounting for add-or-remove-one DP definition but did not account for the sensitivity inflation of virtual clients. We have correct the privacy guarantees to consistently use the substitute-one DP definition, and it does not affect our conclusion. Virtual clients are used for both DP-FedEmb and DP-FedAvg, which is necessary under the extreme heterogeneity, for example, when each user only has images of a single identity. Both DP-FedEmb and DP-FedAvg can achieve user-level DP and are compared under the same DP definition.

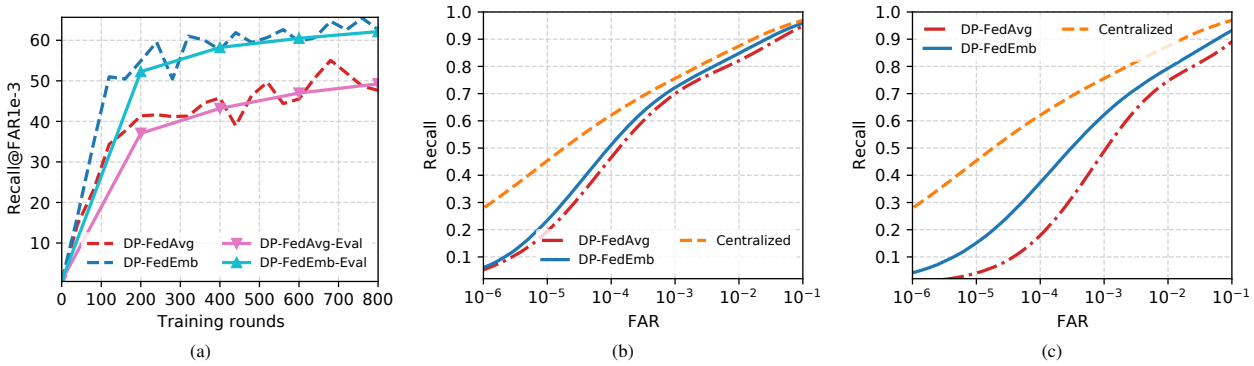

Figure 7. DP-FedEmb and DP-FedAvg on DigiFace10K under the same privacy budget, and additional oracle baseline of centralized training w/o DP: (a) recall@FAR=1e−3 on validation dataset during training, and dashed lines are approximation by sampling a subset of validation users; (b) and (c) ROC curve of trained model on test set, with noise multiplier 0.02 and 0.08, respectively; the x-axis in (b) and (c) are in logarithmic scale.

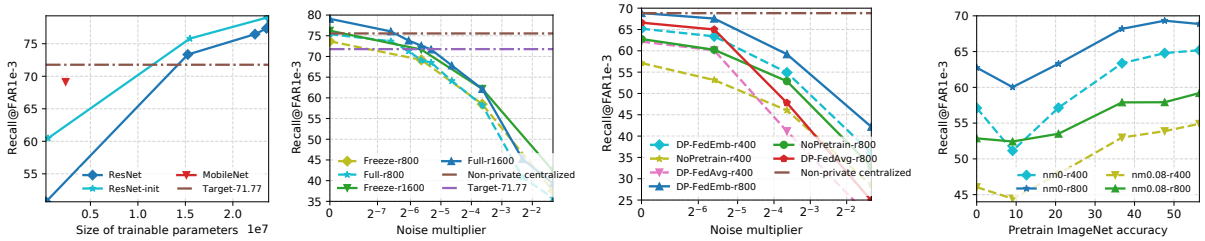

Figure 8. Recall@FAR=1e−3 on DigiFace validation set when training (a) a ResNet with partial frozen parameters for 800 rounds without noise; (b) a partially frozen ResNet with DP-FedEmb; (c) a MobileNet with DP-FedEmb w/ pretraining, and baselines of DP-FedAvg and DP-FedEmb w/o pretraining; (d) a MobileNet with different pretrained models.

in [36] that has $\sim 4M$ parameters. We explore freezing parameters and the alternative model architecture MobileNetV2 of $\sim 2.4M$ parameters. We train parameters of all normalization layers, and gradually freeze the convolutional kernels from lower level to higher level (w or w/o the input convolutional layers) to generate Fig. 8a. For image-to-embedding models, recall@FAR=1e−3 linearly increases with the size of parameters, which is different from the observation that models are redundant for image classification [26, 64]. Additionally training the input convolutional layers [12] is more efficient than only training the higher levels of the network. In Fig. 8b, we freeze the convolutional kernels of intermediate two groups of residual blocks (out of the total four groups) in ResNet-50, which leads to a backbone network of $15.48M$ parameters. The partially frozen model is inferior to the full model for small-medium noise, and only effective in the low-utility regime of large noise 0.4.

For similar parameter size, MobileNetV2 outperforms ResNet-50 with frozen parameters, and Fig. 4a shows the privacy-utility trade-off. Recall@FAR=1e−3 of DP-FedEmb-r800 on MobileNetV2 only drops from 68.86% to 67.56% when 0.02 noise is added, while ResNet-50 drops from 79.09% to 72.6%. However, ResNet-50 still outperforms MobileNetV2 by a large margin in the high utility regime. DP-FedAvg is worse than DP-FedEmb when noises are added.

Public pretraining. Even though the input image size of DigiFace is 112×112, different from the ImageNet pretraning image size of 224×224, the pretrained scale-invariant backbone can consistently improve the performance by $> 5\%$ under the same noise level, as shown in Fig. 4a. Comparing curves of round 400 and round 800, the gain of public pretraining is larger when trained with a smaller number of rounds. We also pretrain a few different MobileNetV2 models on ImageNet by varying the total training epochs, and summarize the results in Fig. 4b. Though the private fine-tuning utility is not linearly increasing with pretraining accuracy, there seems to be a general positive correlation: better pretrained models can lead to better private models except for one outlier where a inferior pretrained model causes difficulty in training. Without private training, the recall@FAR=1e−3 of these pretrained models on DigiFace (with ImageNet validation accuracy) are smaller than 0.6%. Due to the domain difference, the utility of the pretrained model on DigiFace can be low, and it may not be consistent

with the accuracy on ImageNet. For example, a pretrained MobileNetV2 can achieve 56.43% accuracy on ImageNet while only 0.27% recall@FAR=$1e{-}3$ on DigiFace, but it can boost the recall@FAR=$1e{-}3$ of training with DP-FedEmb and 0 noise, from 57.12% for round 400 and 62.76% for round 800 to 65.19% and 68.86%, respectively. Finally, pretraining may not always help. For example, when pretraining from the (preprocessed) Google Landmark (GLD) dataset, the final recall@FAR=$1e{-}3$ can be worse than without pretraining.

D.5. Ablation study curves

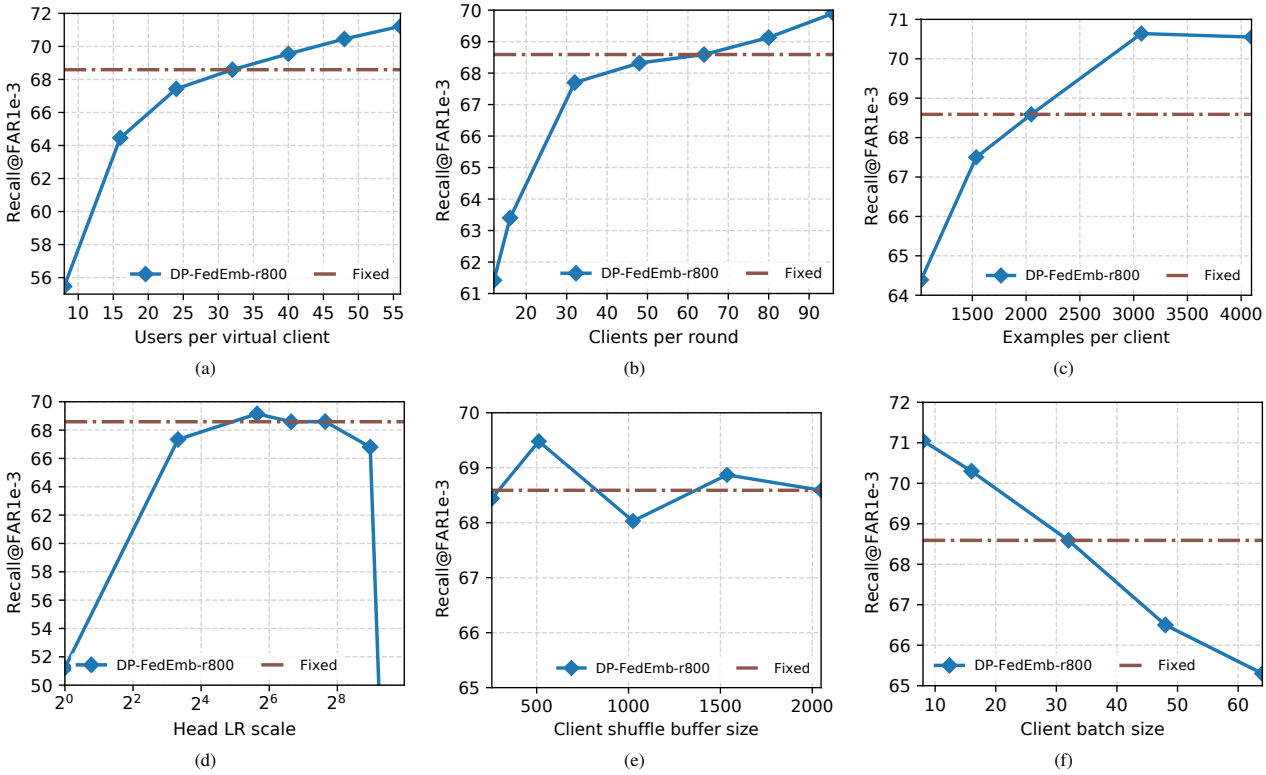

Figure 9. Ablation study for the fixed hyperparameters in (DP-)FedEmb; MobileNetV2 is trained for 800 rounds on DigiFace10K with adaptive clipping [3] and zero noise; all the other hyperparameters are fixed when (a) users per dynamic client (b) clients per round (c) examples per client (d) head LR scale β_2/β_1 (e) client shuffle buffer size (f) client batch size is varied.

D.6. Full table of additional results

Dataset	Algorithm	Hyperparameters				Recall@FAR=$1e{-}3$ / 0.1	
		Noise	SerLR	CliLR	Clip	Approx	AllPair
DigiFace	DP-FedEmb	1.39	$5e{-}3$	$2e{-}3$	0.5	-	19.76 ± 0.43
	DP-FedAvg		$2e{-}3$		1.5	-	13.38 ± 0.27
EMNIST	DP-FedEmb	0.62	0.02	$5e{-}3$	1	10.64 ± 0.5	10.47 ± 0.5
	DP-FedAvg					9.78 ± 0.44	9.67 ± 0.41
GLD	DP-FedEmb	0.02	1	$5e{-}4$	0.3	26.18 ± 0.44	27.07 ± 0.04
	DP-FedAvg		0.5		0.7	24.48 ± 1.0	26.27 ± 0.17
iNat	DP-FedEmb	0.5	0.02	$5e{-}4$	0.2	40.49 ± 1.08	40.93 ± 0.94
	DP-FedAvg		0.01	$1e{-}3$	1	29.57 ± 0.78	29.6 ± 0.65

Table 4. The utility under same privacy budget for DigiFace [6], EMNIST [67], GLD [68], and iNat [69] datasets.

For EMNIST, we train the embedding model on images of class $0 - 35$ and test on images of class $36 - 62$. Using a

relatively large noise multiplier 0.62 for 200 rounds, and sampling 8 users per virtual client and 32 virtual clients per round, $\epsilon = 9.28, \delta = 10^{-4}$ can be achieved given 6800 users. A small nework with two convolutional layers similar to LeNet [43] is used as the backbone network, and no pretrained model is used for initialization. We provide results on EMNIST primarily for reproducibility as the scale of EMNIST is smaller than the other datasets used in this draft.