dSNE: a visualization approach for use with decentralized data

Debbrata K. Saha,[1,4] V. D. Calhoun,[1,2,3,4] Yuhui DU,[4] Zening FU,[4] Sandeep R. Panta,[3] S. M. Plis[2,4]
[1] Georgia Institute of Technology
[2] Georgia State University
[3] The Mind Research Network
[4] Tri-Institutional Center for Translational Research in Neuroimaging and Data Science (TReNDS), Georgia State University, Georgia Institute of Technology, and Emory University, Atlanta, GA 30303
dsaha34@gatech.edu, vcalhoun@gsu.edu, ydu@gsu.edu, zfu@gsu.edu, spanta@mrn.org, s.m.plis@gmail.com

---

Abstract

Visualization of high dimensional large-scale datasets via an embedding into a 2D map is a powerful exploration tool for assessing latent structure in the data and detecting outliers. It plays a vital role in neuroimaging field because sometimes it is the only way to perform quality control of large dataset. There are many methods developed to perform this task but most of them rely on the assumption that all samples are locally available for the computation. Specifically, one needs access to all the samples in order to compute the distance directly between all pairs of points to measure the similarity. But all pairs of samples may not be available locally always from local sites for various reasons (e.g. privacy concerns for rare disease data, institutional or IRB policies). This is quite common for biomedical data, e.g. neuroimaging and genetic, where privacy-preservation is a major concern. In this scenario, a quality control tool that visualizes decentralized dataset in its entirety via global aggregation of local computations is especially important as it would allow screening of samples that cannot be evaluated otherwise. We introduced an algorithm to solve this problem: decentralized data stochastic neighbor embedding (dSNE). In our approach, data samples (i.e. brain images) located at different sites are simultaneously mapped into the same space according to their similarities. Yet, the data never leaves the individual sites and no pairwise metric is ever

directly computed between any two samples not collocated. Based on the Modified National Institute of Standards and Technology database (MNIST) and the Columbia Object Image Library (COIL-20) dataset we introduce metrics for measuring the embedding quality and use them to compare dSNE to its centralized counterpart. We also apply dSNE to various multi-site neuroimaging datasets and show promising results which highlight the potential of our decentralized visualization approach.

1. Introduction

Large-scale datasets have proven to be highly effective in facilitating solutions to many difficult machine learning problems [1]. High tolerance to mistakes and possible problems with individual data samples in applications relevant to internet businesses,[1] together with advances in machine learning methodologies (such as deep learning [2]) are able to effectively average out problems with individual samples lead to improved performance in recognition tasks. The story is different in the domains working with biomedical data, such as neuroimaging, where a data sample is a magnetic resonance image (MRI) of the entire brain containing on the order of 100,000 volumetric pixels (voxels). There, the data collection process for each sample is expensive, considerations of data privacy often prevent pooling data collected at multiple places, thus the datasets are not as large. Yet they are large enough to be difficult to manually vet each sample. An incorrect data sample may still lead to wrong conclusions and quality control is an important part of every analysis. Methods for simultaneous embedding of multiple samples are welcomed, as it is very difficult to scan through each and are used in practice for quality control [3].

A common way of visualizing a dataset consisting of multiple high dimensional data points is embedding it to a 2 or 3-dimensional space. Such embedding can be an intuitive exploratory tool for quick detection of underlying structure and outliers. Although linear methods such as principal component analysis (PCA) provide the functionality they are not usually useful when there is a need to preserve and convey hidden

[1] It is expected that the mistakes average out and even if they do not the cost of displaying an image that the user did not request is low.

nonlinear structure in the data. Many methods were developed for the task on nonlinear data embedding and visualization including Sammon mapping [4], curvilinear components analysis [5], Stochastic Neighbor Embedding [6], Isomap [7], Maximum Variance Unfolding [8], Locally Linear Embedding [9], Laplacian Eigenmaps [10]. The problem with these approaches is in their inability to retain local and global structure in a single map. A method to handle this situation efficiently was alternatively proposed: t-distributed stochastic neighbor embedding (t-SNE) [11]. The embeddings resulting from t-SNE applications are usually intuitive and interpretable which makes it an attractive tool for domain scientists [3, 12].

All of these methods, however, are built on the assumption that the input dataset is locally available. t-SNE, for example, needs computation of the pairwise distances between all samples (points) in the dataset. Yet, in many situations it is impossible to pull the data together into a single location due to legal and ethical considerations. This is especially true for biomedical domain, where the risk of identifiability of anonymized data often prevents open sharing. Any site or institution very rarely shared the biomedical data for the concern of revealing the identity of individual [13].

Meanwhile, many systems allow virtually pooling datasets located at multiple research sites and analyzing them using algorithms that are able to operate on decentralized datasets [14, 15, 16]. The importance of operating on sensitive data without pooling it together and thus generating truly large-scale neuroimaging datasets is so high that researchers successfully engage into manually simulating a distributed system [17]. For all of the applications of the above systems quality control is essential and intuitive visualization of the complete virtual dataset physically spread across multiple locations is an important and much needed tool for filtering out participating sites with bad data, incorrect processing or simply mistakes in the input process.

To cope with this situation, we proposed a way to embed a decentralized dataset into a 2D map that is spread across multiple locations such that the data at each location cannot be shared with others due to e.g. privacy concerns [18]. Recently privacy-preserving analysis on distributed datasets has studied in [19, 20, 21, 22, 23, 24]. Some methods ensure the differential privacy only on private data and several methods take care both public and private data. Even the methods that are seemingly suitable to this

setting (after a possible modification) do not seem to address the problem of inability to compute the distance between samples located at different sites in iterative manner. For example, a method of embedding multiple modalities into the same space was suggested in [25] . We could think of the modalities as our locations and modify their approach to our settings. This, however, is not straightforward as, again, the approach requires measuring co-occurrence, which transcends the borders of local sites.

We base our approach on availability of public anonymized datasets, which we use as a reference and build the overall embedding around it. This is most similar to the landmark points previously used for improving computational efficiency [26, 27]. In fact, we start with a method that resembles the original landmark points approach. We show that it is not as flexible and does not produce good results. Then we introduce a dynamic modification that indeed is able to generate an embedding that reflects relations between points spread across multiple locations. Unlike the original landmark point approach, we use t-SNE as our base algorithm. We call the decentralized data algorithm dSNE. To evaluate the performance and to compare with the centralized version we use the MNIST [28] and COIL-20 dataset [29] and take advantage of the known classes of the samples to introduce a metric of overlap and roundness to quantify the comparisons. We evaluate and compare our algorithms in a range of various settings, establishing 4 experiments with MNIST data and 2 experiments of COIL-20 data. Furthermore, we apply our approach on five different multi-site neuroimaging datasets.

2. Methods

In the general problem of data embedding we are given a dataset of N points $\mathbf{X} = [\boldsymbol{x}_1 \ldots, \boldsymbol{x}_N]$, such that each point $\boldsymbol{x}_i \in \mathbb{R}^n$ with the task of producing another dataset $\mathbf{Y} = [\boldsymbol{y}_1 \ldots, \boldsymbol{y}_N]$, such that each point $\boldsymbol{y}_i \in \mathbb{R}^m$, where $m << n$. Usually $m = 2$ for convenience of visualization. Of course, this is an incomplete definition of the problem, as for any interesting results $\mathbf{Y}$ must be constrained by $\mathbf{X}$.

2.1. t-SNE

In t-SNE the distances between the points in $\mathbf{Y}$ must be as close to the distances between same points in $\mathbf{X}$ as possible given the weighted importance of preserving the relations with nearby points over those that are far. At first, t-SNE converts the high dimensional Euclidean distances between datapoints into conditional probability that represent similarities (see Algorithm 1).

Algorithm 1 `PairwiseAffinities`

Input: p (site index), ρ (perplexity), $\mathbf{X} \in \mathbb{R}^{N \times C \times K_p}$
Output: $\mathbf{P}$
1: Eq. (1) to compute p_{ij} with perplexity ρ
2: $\mathbf{P}_{ij} = (\mathbf{P}_{ji} + \mathbf{P}_{ij})/(2n)$

The similarity of datapoint x_j to datapoint x_i is the conditional probability, $p_{j|i}$, that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian distribution centered at x_i.

Algorithm 2 `tSNE`

Input:
Data: $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots \boldsymbol{x}_N], \boldsymbol{x}_i \in \mathbb{R}^n$
Objective parameters: ρ (perplexity)
Optimization parameters: T (number of iterations), η (learning rate), α (momentum)
Output: $\mathbf{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}, \boldsymbol{y}_i \in \mathbb{R}^m, m << n$
1: $\{p_{ij}\} =$ `PairwiseAffinites` $0, \rho, \mathbf{X}$
2: $\mathbf{Y} \propto \mathcal{N}(0, 10^{-4}\mathbf{I}), \mathbf{I} \in \mathbb{R}^{m \times m}$ initialize from Gaussian
3: **for** $i = 1$ to T **do**
4: Eq. (2) to compute low-dimensional affinities q_{ij}
5: Eq. (3) to compute δC / $\delta \boldsymbol{y}_i$
6: $\boldsymbol{y}_i^t = \boldsymbol{y}_i^{t-1} + \eta(\delta C/\delta \boldsymbol{y}_i) + \alpha(t)(\boldsymbol{y}_i^{t-1} - \boldsymbol{y}_i^{t-2})$
7: **end for**

At the same way we can compute q_{ij} from low dimensional output data. Algorithm 2 outlines the full procedure. In this algorithm, equations (1) and (2) are used to compute high dimensional and low dimensional pairwise affinities respectively. To embed data points into low dimensional space, t-SNE tries to minimize the mismatch between conditional probabilities (referred to as similarities) in higher and lower dimensional space. To achieve this, t-SNE minimizes the sum of Kullback-Leibler di-

vergence of all data points using the gradient descent method. The gradient of the Kullback-Leibler divergence between $\mathbf{P}$ and the Student-t based joint probability distribution $\mathbf{Q}$ is expressed in (3).

$$p_{j|i} = \frac{\exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/2\sigma_i(\rho)^2)}{\sum_{k\neq i}\exp(-||\boldsymbol{x}_i - \boldsymbol{x}_k||^2/2\sigma_i(\rho)^2)} \tag{1}$$

$$q_{ij} = \frac{(1+||\boldsymbol{y}_i - \boldsymbol{y}_j||^2)^{-1}}{\sum_{k\neq l}(1+||\boldsymbol{y}_k - \boldsymbol{y}_l||^2)^{-1}} \tag{2}$$

$$\frac{\delta C}{\delta y_i} = 4\sum_j (p_{ij} - q_{ij})(\boldsymbol{y}_i - \boldsymbol{y}_j)(1+||\boldsymbol{y}_i - \boldsymbol{y}_j||^2)^{-1} \tag{3}$$

Inspired by the overall quite satisfactory performance of t-SNE on a range of tasks, we base on it our algorithms for decentralized datasets.

2.2. dSNE

In a scenario that we consider here, no data can leave a local site and thus it seems impossible to compute distances of samples across the sites. Without those distances (see equation (1)) we will not be able to obtain a common embedding. Fortunately, in neuroimaging there are now multiple large public repositories of MRI data which are able to provide diverse datasets for analyses [30, 31, 32].

First we introduce some notation for sending and receiving messages. For a matrix $\mathbf{X}$, $\mathbf{X}^{\rightarrow p}$ means that it is sent to site p and $\mathbf{X}^{\leftarrow p}$ means that it is received from site p. We assume that a shared dataset is accessible to all local sites and the sites have it downloaded.

We implemented two algorithms: 1) Single-shot dSNE and 2) Multi-shot dSNE. Detailed procedure and experimental results of Single-shot dSNE are provided in [18] and Appendix A.

2.2.1. Multi-shot dSNE

For multi-shot dSNE we at first pass the reference data from centralized site C to each local site. Now each local site's data consists of two portions. One is its local dataset, for which we need to preserve privacy, and another one is the shared reference

Algorithm 3 `GradStep`

Input:
Data embeddings: $\mathbf{Y}_p$ (local), $\mathbf{Y}_s$ (shared), $\mathbf{P}$
Optimization parameters: η, α

Output: $\hat{\mathbf{Y}}_p$ (local), $\hat{\mathbf{Y}}_s$ (shared)

1: Eq. (2) to compute low-dimensional affinities q_{ij}
2: Eq. (3) to compute δC / $\delta \boldsymbol{y}_i$
3: $\hat{\boldsymbol{y}}_i = \eta(\delta C/\delta \boldsymbol{y}_i) + \alpha(\boldsymbol{y}_i^{t-1} - \boldsymbol{y}_i^{t-2})$
4: group $\hat{\boldsymbol{y}}_i$ into $\hat{\mathbf{Y}}_p$ (local) and $\hat{\mathbf{Y}}_s$ (shared)

dataset both comprise the combined datasets. One master node plays the most crucial role in every multi-shot iteration. The pseudocode and overall procedure for multi-shot dSNE are shown in Algorithm 6. In multi-shot dSNE each local site computes gradient (Algorithm 3) based on the combined datasets and passes the embedding vector Y for shared data to master node at each iteration. The master node collects the shared data with different embedding vectors Y from different local sites and computes the average of Y values. Next, the master node passes the new computed value of embedding vector Y for reference data to all local sites and the local site updates (Algorithm 4) it's local and shared embedding vector Y. After that, one global mean is computed in master node by taking the mean value from each site's samples. Finally, master node passes the global mean to each local site and every site fixes their global position using global mean (Algorithm 5). Note, at each iteration the embedding vector Y for the shared dataset will be the same at all local sites. So at every iteration the local values of different sites are influenced by the same and common reference data.

Algorithm 4 `UpdateStep`

Input:
Data embeddings: $\mathbf{Y}_p$, $\mathbf{Y}_s$, $\hat{\mathbf{Y}}_p$, $\hat{\mathbf{Y}}_s$

Output: $\mathbf{Y}_p$, $\mathbf{Y}_s$

1: $\mathbf{Y}_p = \mathbf{Y}_p + \hat{\mathbf{Y}}_p$
2: $\mathbf{Y}_s = \mathbf{Y}_s + \hat{\mathbf{Y}}_s$

2.3. Comparison Metrics

The assessment of cluster quality control is an essential part of cluster analysis. Different techniques and approaches are proposed and used recently for the measure-

Algorithm 5 `deMean`

Input:

Data embeddings: $\mathbf{Y}$, $\mathbf{Y}_{Gmean}$

Output: $\mathbf{Y}$

1: $\mathbf{Y} = \mathbf{Y} - \mathbf{Y}_{Gmean}$

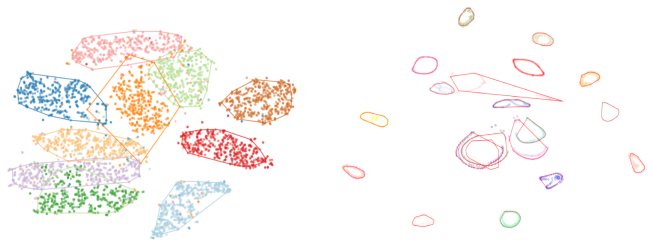


Figure 1: A t-SNE output on centralized MNIST and COIL-20 dataset and outlier-free convex hull boundaries.

ment of the cluster quality control. Davies-Bouldin index (DB) [33] is used to find out compact and well separated cluster by measuring similarity between clusters. The Dunn index [34] measures the ratio of inter and intra-cluster distance to validate the cluster performance where large index value represents good cluster outcome. The SD index [35] evaluates clustering schemes considering scattering within and between clusters where index is slightly influenced by the maximum value of clusters' number. The Bayesian information criterion (BIC) index [36] which is derived from Bayes's theorem considers appropriate fitting of model and the corresponding complexity to the applied dataset. The silhouette coefficient [37] takes into account both tightness and separation to measure the quality of a cluster. Besides there is another approach called the external index where evaluation is measured based on the comparison of computed cluster and predefined cluster structure. For the external comparison metric different approaches such as an F-measure, entropy, purity, and rand index can be used. A good comparison is illustrated between external and internal indices in [38].

A great concern in our distributed low dimensional embedding is to ensure point of a specific class is embedded in another cluster. We consider here two a priori classified clustering datasets (1) the MNIST dataset [28] and (2) the COIL-20 dataset [29] whose clusters are known and described in [11]. So in our case we consider that the

Algorithm 6 `multishotDSNE`

Input:

Objective parameters: ρ (perplexity)

Optimization parameters: T, η, α

Shared Data: $\mathbf{X}_s = [\boldsymbol{x}_1^s, \boldsymbol{x}_2^s \dots \boldsymbol{x}_{N_s}^s], \boldsymbol{x}_i^s \in \mathbb{R}^n$

Data at site $p \forall p$: $\mathbf{X}_p = [\boldsymbol{x}_1^p, \boldsymbol{x}_2^p \dots \boldsymbol{x}_{N_p}^p], \boldsymbol{x}_i^p \in \mathbb{R}^n$

Output: $\mathbf{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_N\}, \boldsymbol{y}_i \in \mathbb{R}^m, m << n, N = \sum_p N_p + N_s$

$\mathbf{Y}_s \propto \mathcal{N}(0, 10^{-4}\mathbf{I}), \mathbf{I} \in \mathbb{R}^{m \times m}$ initialize from Gaussian
for $p = 0$ to P **do** ▷ Initialize at sites
    $\mathbf{Y}_s^{\rightarrow p}$
    $\mathbf{P}_p \leftarrow$ `PairwiseAffinities` $p, \rho, [\mathbf{X}_p, \mathbf{X}_s]$
    $\mathbf{Y}_p \propto \mathcal{N}(0, 10^{-4}\mathbf{I}), \mathbf{I} \in \mathbb{R}^{m \times m}$
end for
for $i = 0$ to T **do**
    for $p = 0$ to P **do** ▷ At local sites
        $\hat{\mathbf{Y}}_p, \hat{\mathbf{Y}}_s \leftarrow$ `GradStep` $[\mathbf{Y}_p, \mathbf{Y}_s, \mathbf{P}_p]$
    end for
    $\hat{\mathbf{Y}} \leftarrow 0$ ▷ At the master
    for $p = 0$ to P **do** ▷ At the master
        $\hat{\mathbf{Y}}_s^{\leftarrow p}$
        $\hat{\mathbf{Y}} \leftarrow \frac{1}{P}\hat{\mathbf{Y}}_s$ ▷ Average local $\hat{\mathbf{Y}}_s$
    end for
    for $p = 0$ to P **do** ▷ At local sites
        $\hat{\mathbf{Y}}^{\rightarrow p}$
        $\mathbf{Y}_p, \mathbf{Y}_s \leftarrow$ `UpdateStep` $[\mathbf{Y}_p, \mathbf{Y}_s, \hat{\mathbf{Y}}_p, \hat{\mathbf{Y}}_s]$
    end for
    for $p = 0$ to P **do** ▷ At local sites
        $\mathbf{Y}_{mean} = \overline{\mathbf{Y}}$ ▷ Average $\mathbf{Y}$
    end for
    for $p = 0$ to P **do** ▷ At the master
        $\mathbf{Y}_{mean}^{\leftarrow p}$
        $\mathbf{Y}_{Gmean} \leftarrow \frac{1}{P}\mathbf{Y}_{mean}$ ▷ Average $\mathbf{Y}_{mean}$
    end for
    for $p = 0$ to P **do** ▷ At local sites
        $\mathbf{Y}_{Gmean}^{\rightarrow p}$
        $\mathbf{Y} \leftarrow$ `demean` $[\mathbf{Y}, \mathbf{Y}_{Gmean}]$
    end for
end for

known number of clusters are our ground truth. We introduce three new validation techniques (1) **K-means ratio** (2) **Intersection area** and (3) **Roundness** to compute only the cluster performances based on the low dimensional embedding. An objective

comparison of inherently subjective visualization algorithms is difficult. We used the k-means criterion which is the ratio of intra and inter-cluster distances between the clusters:

$$\alpha = \frac{\sum_{d=0}^{9} \sum_{S \epsilon X_d} ||\boldsymbol{\mu}_d - \boldsymbol{x}_s^d||_2}{\sum_{(i,j),(i>j),(i \neq j)} ||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2} \tag{4}$$

In an attempt to quantify perceptional quality of the resulting embeddings we have developed two additional metrics: Intersection area which is basically the overlap between the clusters and roundness. To remove sensitivity to noise we first remove the outliers (See Figure 1) in each digit's cluster [39]. Then we compute the convex hull for each digit and use them to compute the measure of the overlap– the sum of all polytope areas minus the area of the union of all the polytopes normalized by this union's area and the roundness– the ratio of the area of each polytope to the area of the circumscribed circle. To compute the roundness we represent a cloud with a convex hull. To remove the effect of differences in perimeter, etc. we first normalize the measure. This normalization effectively approximates the process of making the perimeters equal. For the same perimeter a circle has the largest area. We compute the area of the polytope and use it as our measure of roundness.

3. Data

We base our experiments and validation on seven datasets:

1. MNIST dataset[2] [28]
2. COIL-20 dataset[3] [29]
3. Autism Brain Imaging Data Exchange (ABIDE) fMRI dataset[4] [40]
4. Pediatric Imaging Neurocognition Genetics (PING) dataset[5] [41]
5. Structural Magnetic Resonance Imaging (sMRI) dataset

[2] `https://www.kaggle.com/c/digit-recognizer`
[3] `http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php`
[4] `http://fcon_1000.projects.nitrc.org/indi/abide/`
[5] The full dataset is available here: `http://pingstudy.ucsd.edu/Data.php`

6. Function Biomedical Informatics Research Network (fBIRN) structural MRI dataset [42] and
7. Bipolar and Schizophrenia Network for Intermediate Phenotypes (BSNIP) structural MRI dataset [42]

MNIST dataset for handwritten images of all digits in the 0 to 9 range were taken from a Kaggle competition which has $28,000$ gray-scale images of handwritten digits. Each image contains $28 \times 28 = 784$ pixels. Among these data, we randomly (but preserving class balance) pick 5,000 different samples from the data set for our needs. At first, we reduce dimension of the data from 784 to 50 using PCA. Then, dSNE and t-SNE are used to generate (x, y) coordinates in two dimensional space.

COIL-20 dataset contains images of 20 different objects. Each object was placed on a motorized turntable against a black background and taken picture in every 5 degrees interval between the rotation of $0 - 360°$. So eventually each object was viewed from 72 equally spaced orientations, yielding a total of 1,440 images. The images contain $32 \times 32 = 1024$ pixels.

ABIDE fMRI dataset contains data of 1153 subjects accessible through the COINS data exchange (http://coins.mrn.org/dx). The ABIDE data has been pre-processed down to multiple spatial and temporal quality control (QC) measures[6]. For ABIDE, because of the low dimension of QC measures, we do not use dimensionality reduction at the beginning but directly run our t-SNE and dSNE to produce the embeddings.

PING is a multi-site study which contains neuro developmental histories, information about developing mental and emotional functions, multimodal brain imaging data, and genotypes for well over 1000 children and adolescents between the ages of 3 to 20. We take 632 subjects from this multi-sites data for our experiment.

sMRI scans (3D T1-weighted pulse sequences) are pre-processed through voxel based morphometry (VBM) pipeline using the SPM5 software. VBM is a technique using MRI that facilitates examination of focal differences in brain anatomy, using the statistical approach of parametric mapping [43]. The unmodulated gray matter

[6]The full list is available here: `https://github.com/preprocessed-connectomes-project/quality-assessment-protocol/tree/master/normative_data`

concentration images from the VBM pipeline are registered to the SPM template. In some cases the non-modulated maps are preferred to modulated maps in accordance with existing literature as per [44]. To reduce computation load on the system, and improve run time of the proposed algorithm, for each scan, the voxel values at every location from all the brain slices are first added across slices, resulting in a matrix size of 91×109. For each scan, all the voxel values from this image are given as inputs to t-SNE and dSNE algorithm.

fBIRN and **BSNIP** datasets that are used in this study are collected from the fBIRN and BSNIP projects respectively. fBIRN data are collected from 7 imaging sites and BSNIP data are collected from 6 imaging sites. The subject was selected based on head motion ($\leq 3^\circ$ and $\leq 3mm$) and good functional data providing near full brain successful normalization [45]. These criteria yielded a total of 311 subjects (160 schizophrenia (SZ) patients and 151 healthy control (HC)) for the fBIRN dataset and 419 subjects (181 SZ and 238 HC) for the BSNIP dataset. In this study, the Neuromark pipeline [42] was adopted to extract reliable intrinsic connectivity networks (ICNs) that are replicated across independent dataset.

4. Experimental setup

4.1. MNIST data

Experiment 1 (Effect of the sample size):. Often centralized stores accumulate more data than any separate local site can contain. Our goal is to check the adaptability of our algorithm in this case. In this experiment, every local site contains only one digit and the reference dataset contains all digits (0-9). We consider 2 cases: when each site contains 400 samples and each digit in the reference consists of 100 samples; and the inverse case, when sites only have 100 samples, while the reference digits are represented by 400.

Experiment 2 (No diversity in the reference data):. In this experiment, the reference dataset contains a single digit that is also present at all of the three (3) local sites. We run an experiment for each of the 10 digits. Each site contains 400 samples for each of its corresponding digits. Reference dataset contains 100 samples of its digit.

Experiment 3 (Missing digit in reference data):. In this experiment we investigate the effect of the case when a digit is missing from shared data. This approximates the case of unique conditions at a local site. Each local site out of 10 contains a single digit. We run 10 experiments; in each, the reference dataset is missing a digit. For each of the experiments we have 2 conditions: in one the reference dataset is small (100 samples for each digit but the missing one) while the sites are large (400 samples per site); in another the reference data is large (400 samples per digit) and the sites are small (only 100 samples).

Experiment 4 (Effect of the number of sites):. In this experiment, we investigate whether the overall size of the virtual dataset affects the result. Every local site, as well as the reference data, contains all digits (0-9) 20 samples per digit. We continuously increase the number of sites from 3 to 10. As a result, the total number of samples across all sites increases.

4.2. COIL-20 data

Experiment 1 (Effect of the sample size):. Like the experiment 1 of MNIST dataset, we apply the same strategy where centralized stores accumulate more data than any separate local site can contain. In this experiment, every local site contains only one type of object and the reference dataset contains all type of objects (1 to 20). We consider 2 cases: when each site contains 52 samples of its corresponding object and each object in the reference consists of 20 samples; and the inverse case, when sites only have 20 samples, while the reference objects are represented by 52.

Experiment 2 (Missing object in reference data):. In this experiment we investigate the effect of the case when some objects are missing from shared data like experiment 3 of MNIST dataset. Each local site out of 20 contains a single object. We run 10 experiments with different random seed; every time, the reference dataset is missing objects from local site 16-20. For each of the experiments we have 2 conditions: in first case the reference dataset is small (20 samples for each object but the missing ones) while the sites are large (52 samples per site); in another the reference data is large (52 samples per object) and the sites are small (only 20 samples).

4.3. ABIDE fMRI data

To simulate a consortium of multiple sites for ABIDE dataset, we randomly split these data into ten local and one reference datsets for dSNE experiment. We run three different dSNE experiments on ABIDE data. In each experiment, we pick 10 sites randomly to act as local sites and accumulate rest of the site's data to form reference sample. And beside that, we collect all data together and run t-SNE on this centralized data to access the performance with our decentralized setup.

4.4. sMRI data

Our sMRI data consists of four different ages people. The age range for four different samples are below 11, 11 to 18, 30 to 34 and above 64 respectively. We run three experiments on this dataset.

In first experiment, we place data for each individual age range people into unique local sites(four sites total) and form reference sample by taking 100 samples from each local site. In second experiment, we keep local sites identical compared to experiment 1 but create reference data by taking 100 samples from site 1 only. We analysis the behavior of common data samples case in experiment 3 where different local sites contain data of same age range people. In experiment 3, we keep the same set up as experiment 1 for reference samples, but we randomly take 50 samples from site 2 and place in site 1 and besides we take 100 samples from site 4 and distribute equally between site 2 and site 3.

4.5. PING data

We collect PING data from five different data sources and run four different experiments on them. For experiment 1, we place data for each data source into unique local site(total five local sites) and form our reference sample by taking very low samples from each local site. We keep the same set up for experiment 2 compared to experiment 1, but take 100 samples from site 2 only to create reference sample. We test the case of common data samples for different local sites in experiment 3. Here we randomly take 30 samples from site 2 and place in site 1, take 20 samples from site 3 and place in site 2, take 10 samples from site 2 and place in site 3, take 10 samples from

site 1 and place in site 4. We form reference sample by the same manner of experiment 1. Finally, for experiment 4, we keep sites 1,3 and 4 unchanged and create reference sample by taking the whole data from site 2 and site 5 (eventually create total 3 local sites and one referenced sample).

4.6. fBIRN and BSNIP data

For the experiment of fBIRN and BSNIP, We run t-SNE separately on both of these datasets. As we collected fBIRN data from seven different sites, we considered them as local sites (seven local sites total) and accumulated BSNIP data from six imaging sites and used as reference sample for our dSNE experiment. We also ran t-SNE on combined datasets (fBIRN + BSNIP) for visual comparisons with dSNE.

5. Results

5.1. MNIST data

For MNIST experiment, we only present the results of Experiment 1. The rest experimental results (Experiments 2, 3 and 4) are presented in [18]. Figure 2 represents the result of experiment 1. In the center, the upper figure represents the best, and the lower one is for worst performing run and the layout is colored by digits. On the right, the upper and lower figures are the same as column 2 but colored by sites. In the boxplots, t-SNE was computed on pooled data; SMALL and LARGE represent smaller and larger size of the reference dataset in dSNE runs. The comparison metrics show that when the shared portion contains large amount of data the metrics are better than in the case of smaller number of samples in the reference dataset. However, the cluster roundness degrades with the size of the sample in the shared data. dSNE clusters are less round compared to the centralized t-SNE.

5.2. COIL-20 data

Figure 3 depicts the results of Experiment 1 of COIL-20 dataset. In the center, the upper figure represents the best, and the lower one is for worst performing run and the layout is colored by objects. On the right, the upper and lower figures are the same as

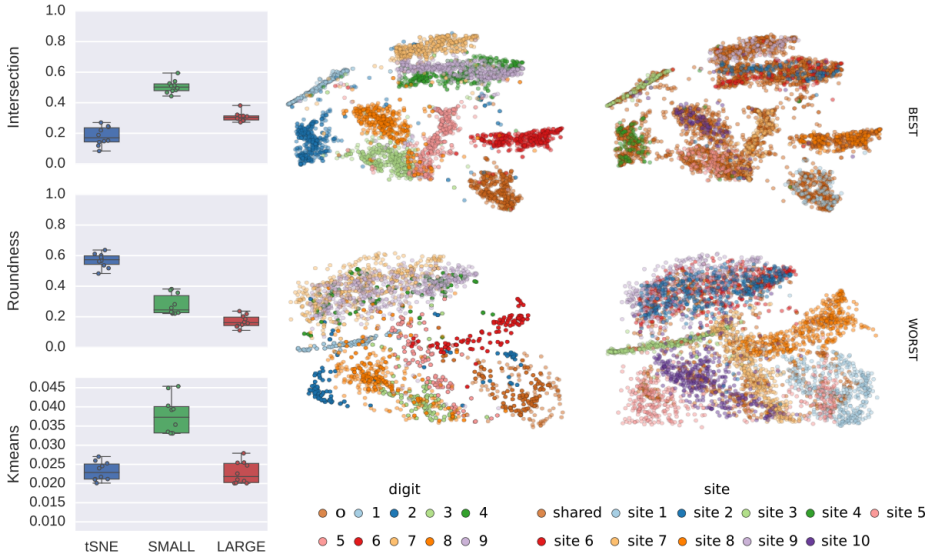


Figure 2: MNIST Experiment 1: Reference data contains samples of all MNIST digits but it is either small or large amount. In the boxplots, tSNE was computed on pooled data; SMALL and LARGE represent smaller and larger size of the reference dataset in dSNE runs. In the center, the upper figure represents the best, and the lower figure – the worst performing run and the layout is colored by digits. On the right, the upper and lower figures are the same as column 2 but colored by sites.

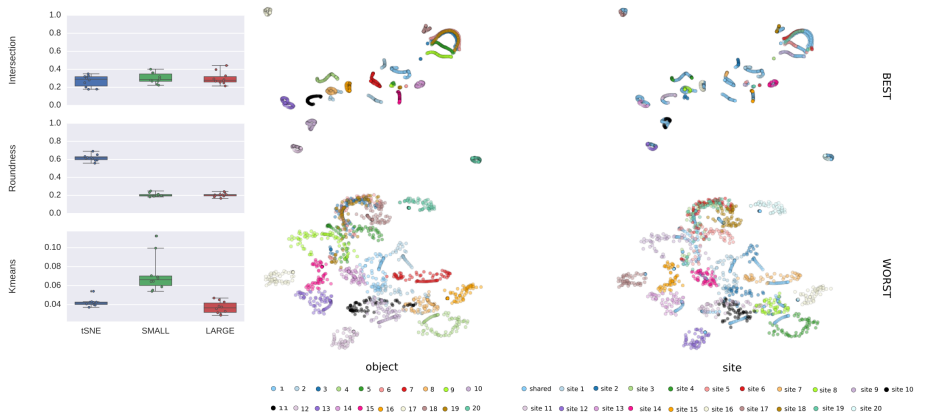


Figure 3: COIL-20 Experiment 1: Reference data contains samples of all COIL-20 objects but it is either small or large amount. In the boxplots, tSNE was computed on pooled data; SMALL and LARGE represent smaller and larger size of the reference dataset in dSNE runs. In the center, the upper figure represents the best, and the lower figure – the worst performing run and the layout is colored by objects. On the right, the upper and lower figures are the same as column 2 but colored by sites.

column 2 but colored by sites. In the boxplots, t-SNE was computed on pooled data; SMALL and LARGE represent smaller and larger size of the reference dataset in dSNE runs. In comparison metrics, we see similar type characteristics like experiment 1 of MNIST dataset. For large amount of shared data, the comparison metric shows better performance than in the case of smaller samples in the reference dataset.

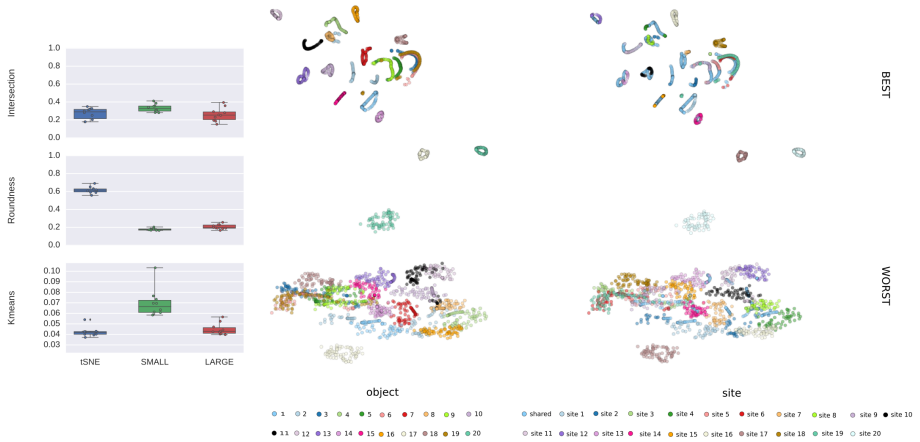


Figure 4: COIL-20 Experiment 2: the reference dataset is totally missing one unique COIL-20 object that is present at one of the local sites. In the boxplots, tSNE was computed on pooled data; SMALL and LARGE represent smaller and larger size of the reference dataset in dSNE runs. In the center, the upper figure represents the best, and the lower figure – the worst performing run and the layout is colored by objects. On the right, the upper and lower figures are the same as column 2 but colored by sites.

Figure 4 depicts the results of Experiment 2 of COIL-20 dataset. In comparison metric, we always obtain better results for kmeans, intersection ratio and roundness for large reference samples compared to the case of fewer samples in the reference. For small reference samples, we observe highly overlapped clusters for which it is very hard to distinguish the clusters for different objects.

5.3. Biomedical Data

We investigate performance of multi-shot dSNE in comparison with the embedding produced by t-SNE on the pulled data using the QC metrics of the ABIDE fMRI, sMRI, PING, fBIRN and BSNIP datasets.

5.3.1. ABIDE fMRI data

The layout of ABIDE data is shown in Figure 5. Results show 10 different clusters for centralized data. For three random splits of our decentralized simulation we obtain 10 different clusters as well. Notably, the split into the clusters in the embedding is stable regardless of the split into sites.

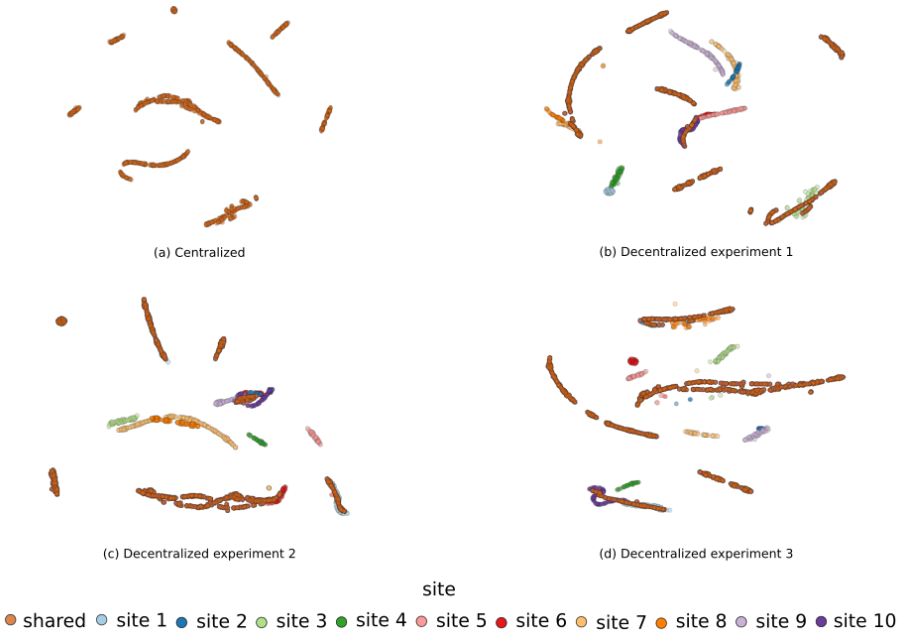
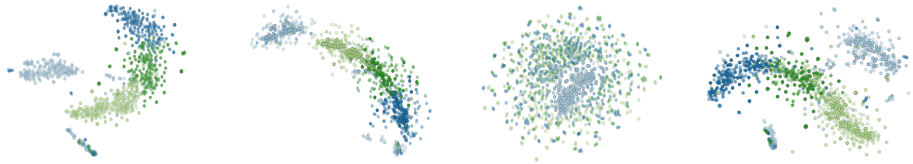


Figure 5: Experiment for QC metrics of the ABIDE dataset. (a) The tSNE layout of pooled data; (b),(c) and (d) are the dSNE layout for three different experiments of. In each experiment, there exists total 10 local sites and one reference data. Like tSNE, in decentralized setup, we get ten clusters as well and each site is marked by unique color.

5.3.2. sMRI data

Figure 6 depicts the results of sMRI Experiments. For experiment 2, we get bad results where all of the clusters of different age people are overlapped and don't get separate clusters at all. But for experiment 1 and 3, we get very noticeable output compared to centralized one. In our dSNE setup, the clusters are not clearly separable, but even for centralized setup of t-SNE, we don't get significantly separable clusters.

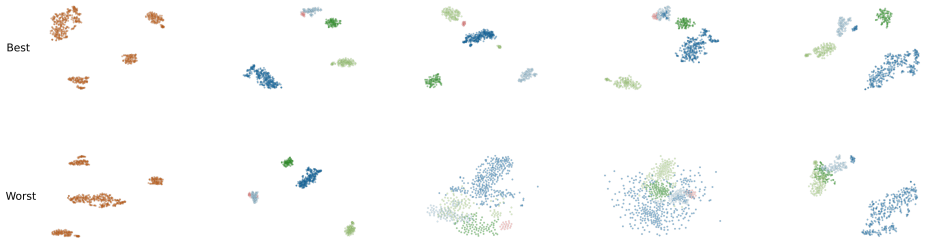


(a) Centralized (b) Decentralized experiment 1 (c) Decentralized experiment 2 (d) Decentralized experiment 3

Figure 6: Experiment for QC metrics of the sMRI. (a) The tSNE layout of pooled data. (b), (c) and (d) are the dSNE layout for three different experiments. In all experiments, four classes of different ages people participate and each class is marked by unique color.

5.3.3. PING data

Figure 7 depicts the experimental results of PING data. We observe very significant results. We get four major clusters after running t-SNE on pooled data and in best case scenario, we obtain same number of clusters as well in all of our dSNE experiments.



(a) Centralized (b) Experiment 1 (c) Experiment 2 (d) Experiment 3 (e) Experiment 4

Figure 7: Experiment for QC metrics of the PING dataset. (a) The tSNE layout of pooled data. In all columns, the top figure presents the best, and the lower one represents the worst performing run. (b), (c) (d) and (e) are the dSNE layout of four different dSNE experiments. In all dSNE experiments, we get four clusters like tSNE in best case scenario.

5.3.4. fBIRN and BSNIP data

Figure 8 depicts the results of fBIRN and BSNIP Experiments. On the left, we present the t-SNE layout where top and bottom ones stand for the layout of fBIRN and BSNIP respectively. We get pretty much well separated groups(HC and SZ) for fBIRN but not in BSNIP. In the center, we present the t-SNE layout of combined datasets where top and bottom figures are colored by groups and sites respectively. We ran t-SNE on the combined datasets but during plotting we only show fBIRN subjects. Because our major goal is to see how subjects from the same group of fBIRN data from different sites embed together in low dimensional space. On the right, we present dSNE

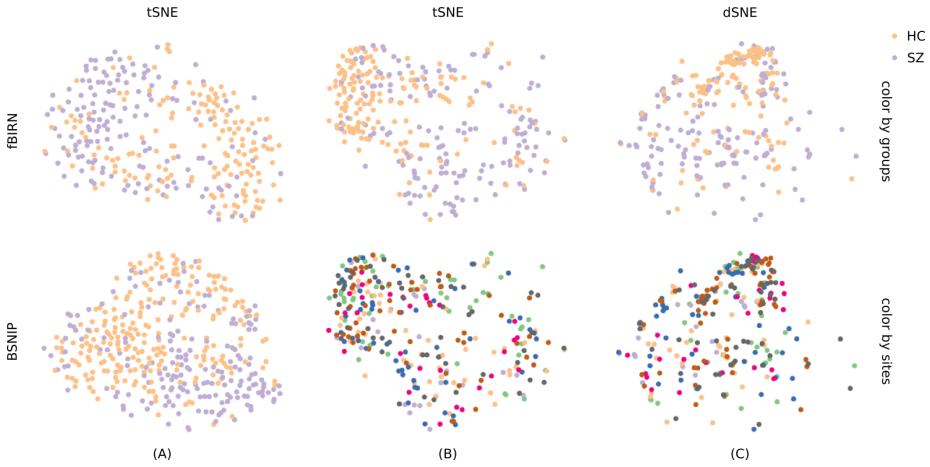


Figure 8: Experiment for QC metrics of fBIRN and BSNIP dataset. A) Top one represents the tSNE layout of fBIRN and bottom one is for BSNIP respectively. (B) Top one is the t-SNE layout of combined(fBIRN + BSNIP) datasets which is colored by groups and bottom one is same layout but colored by sites. (C) Top one is dSNE layout of combined(fBIRN + BSNIP) datasets which is colored by groups and bottom one is same layout but colored by sites.

layout of combined datasets where top and bottom figures are colored by groups and sites respectively. Again, we only plotted fBIRN after running dSNE on the combined datasets. From the layout of t-SNE and dSNE on the combined datasets, we notice that subjects from HC group are densely clustered in both plot. We get weakly densed clusters for SZ group but embeddings are pretty much similar in both t-SNE and dSNE plot. In dSNE layout, we get much overlaps compared to t-SNE. It's reasonable in a sense that there is no direct communication between private local sites. So during plotting some points can be embedded in a very small dense region. We also plotted the embedding colored by sites. We did not see any evidence of bias in which data are grouped by sites in embedding.

6. Discussion

The current practice of data sharing faces great challenge as the field grows more concerned with the potential of subject de-identification. Based on scanning location and diagnosis for a specific disease one may be able to identify subjects who has been suffering for rare diseases [46] [47]. Yet, the inability to combine datasets from differ-

ent research groups can be devastating as individual studies rarely contain enough data to meet the challenges of the questions of foremost importance for biomedical research. Addressing the problem of data scarcity at individual sites and in an attempt to improve statistical power, several methods that operate on data spread across multiple sites have been introduced. Virtual Pooling and Analysis of Research Data (ViPAR) [48] is one of the proposed software platforms that tackles this problem. In this framework a secure and trusted master node or a server synchronizes with all the remote sites involved in the computation. At each iteration each remote site sends data via an encrypted channel to this secure server. These data are then stored in RAM on the VIPAR master server where they are analysed and subsequently removed without ever permanently being stored. But this process still relies on sending the data outside the original site, even if via an encrypted channel, which incurs severe bandwidth and traffic overhead and ultimately increases computational load. Nevertheless, it is a step forward in computing on private dataset. One of the promising approaches is the Enhancing Neuroimaging Genetics through Meta-Analysis (ENIGMA) consortium [49] which does not require local sites to share the data but only their summary statistics. ENIGMA uses both mega (if possible) and meta analysis. In meta analysis, each local site runs same analysis (e.g. regression) using the same measurements of the brain and finally aggregates summary statistics from all sites. This gives significant performance, but eventually addresses the problem of manual processing. To run the same type of simulations by maintaining same measurements criteria, the lead site has to coordinate with all the local sites before starting and after the completion of computation. Most importantly, this approach does not guarantee that it will prevent the re-identification of individuals. Meanwhile, it does no't support multi-shot computation, i.e. computing results in an iterative manner [50]. In many machine learning problems, there are many cases where statistics exchange in a single shot is not enough and multi-shot is needed to get to the optimal solution [18]. EXpectation Propagation LOgistic REgRession (EXPLORER) is one of the existing models where the computation occurs by the communication between the server and different local sites [51]. At first, server and a client draw a random number each and after that the server stores the summation of its random numbers and the client's number after collecting it from the client. In the next step, client adds it's

local data with its own random number and sends the summed up data to the server and finally the server gets info about the local data.

The recent ongoing research on federated learning, differential privacy and encrypted computing is described at [52]. The Intel corporation has started a collaboration with the University of Pennsylvania and 19 other institutions to advance real world medical research using the federated learning. They showed that a deep learning model which is trained by traditional approach on pooled data, can be trained up to 99% of the pooled accuracy using the federated learning approach [53].

Several tools and algorithms are introduced to handle this federated computing efficiently. PySyft [54] is one of OpenMined's Python code libraries which integrates cryptographic and distributed technologies with PyTorch and Tensorflow. It is developed to train an AI model in a secured way by ensuring patient privacy using the distributed data. Our platform COINSTAC [16] is another example of an open source platform addressing these tasks. Researchers at Google Inc. has introduced a model of federated learning using distributed data of user's mobile devices [55]. In this model, a mobile device downloads the current model and trains that model by accessing the data of this device. After that it summarizes the changes and sends them as an update to cloud using encrypted communication. Finally, all the updates coming from all devices are averaged in the cloud eventually improves the shared model.

A dynamic and decentralized platform called Collaborative Informatics and Neuroimaging Suite Toolkit for Anonymous Computation (COINSTAC) [16] was introduced to address the difficulty of data sharing. This platform gives scope to perform distributed computation by using the commonly used algorithm into privacy-preserving mode. But it's very challenging to use algorithms like t-distributed stochastic nonlinear embedding (t-SNE), shallow and deep neural networks, joint ICA, IVA etc. in distributed fashion. To enable decentralized deep neural network in this framework one feed-forward artificial neural network was developed [56]. It's capable of learning and running computation among distributed local site data. Each batch contains one gradient per site and mini-batch gradient descent was used to average the gradient across sites. One master node plays role for averaging the resulting gradients.

To fit traditional joint ICA into this framework, decentralized joint iCA(djICA)

[57] was developed. The overall demonstration of decentralized computation and performance compared to centralized analysis of this algorithm is described in [58].

Federated Averaging (FedAvg) is one of the computation techniques in federated environment introduced in 2016 to fit a global model to the training data which is distributed across local sites [55]. In this model, the weights of a neural network are initialized on the server and distributed to the local clients. Locally the model is trained on the local data over multiple epochs and the trained model is then delivered to the server. After collecting weights from all local clients, the server computes the average and sends these weights back to the local site again. This model gives satisfactory performance for IID data. But the accuracy drops when the local sites contain non IID data. To overcome the statistical challenge for non IID data, a proxy data sharing technique was introduced in 2018 [59]. In this approach, a publicly available shared dataset with uniformly distributed class labels is stored in the cloud accessible to all nodes. During the initialization of FedAvg, a warm up model is trained over these shared data and a fraction of those shared data are sent to each local site. Local model is then trained by FedAvg using the combination of shared data with their own local data over a fixed number of communication rounds. The cloud then aggregates all local models and updates the global model using FedAvg.

We used a very similar approach in dSNE [18] before proposing proxy data sharing technique [59]. In dSNE, one publicly available dataset is accessed by all local sites and each site updates their local data using the combination of shared data with their own local data. Like the communication round in [59], each site runs the operations over a fixed number of iterations to reach into optimal solution.

Considering the complexity of local client-side computation, communication cost, and test accuracy, another model called Loss-based Adaptive Boosting FederatedAveraging (LoAdaBoost FedAvg) was introduced where local clients with higher cross-entropy loss are further optimized before averaging in the cloud by FedAvg [60]. Here the proxy data sharing technique [59] is used for non IID data. In this method, cross entropy loss is used during training local models in a way that if any local model is underfitted, more epoch is used for training and fewer epoch is used for overfitted local model. Consequently, global model is not affected after the averaging of all local mod-

els in the cloud because all overfitting or underfitting problems are resolved on local sites. In dSNE, we also apply the averaging technique where local model is averaged after each iteration and transferred to all sites by master node.

The need for data assessment and quality control is growing day by day. But in a federated settings when the data is decentralized among multiple sites, local sites and institutions may not be willing to share their data to preserve the privacy of their subjects among many reasons. To cope with this situation we introduced a way to visualize a federated datasets in a single display: dSNE [18], which can be used for the quality control of decentralized data while preserving their privacy . One master node communicates with all local sites during whole computation period. At each iteration, the master node manipulates the reference data that is open and not private and common to all local sites. Averaging seems the simplest way but most sophisticated strategy which may turn significance results [50]. Our multi-shot approach also follows this strategy. The performances of tSNE and dSNE are presented in Figure 2 to 8. For the best case scenario, dSNE almost replicates tSNE and it shows significance performance in terms of comparison metrics. We obtain the optimal embedding and comparison metric values for large and more variable reference samples. The performance increases when reference data contains a large amount of samples which is shown in Figure 2 for MNIST dataset. We see a similar type of behavior for COIL-20 dataset which is shown in Figure 3 and Figure 4. The influence of large reference samples into performance was also shown in djICA [57]. We observe very significant results for five different biomedical dataset(ABIDE, sMRI, PING, fBIRN and BSNIP) which are shown in Figure 5 to 8.

We also implemented single-shot dSNE but didn't find significance results. We observed overlapped and crowding problems which is shown in Figure A.9 for MNIST dataset. In single shot dSNE, there is no way to communicate iteratively among different sites. For single shot, the major problem of averaging arises when different sites are widely varying sized and belong to different population [16]. For multi-shot dSNE, the reference dataset should be most variable and large; otherwise there is a high chance of

not getting the optimal embedding. In each run, we may not get better results because initially it randomly initializes the low dimensional points and optimizes iteratively. It may not reach into optimal solution for any certain random initialization.

The implementation and the deployment of dSNE algorithm in Coinstac framework is still in progress but very soon it will be able to run with real time multi-sites computation. We envision other mentioned frameworks adapting the approach for visualization and QA in their respective implementations.

7. Conclusion

Our approach enables embeddings of high dimensional private data which is spread over multiple sites into low dimensional space and visualize them into privacy-preserving mode. Throughout the whole computation, private data never leaves the sites and only minimal gradient information from the embedding space gets transferred across the sites. The clusters in the output embeddings are formed by class elements possibly present across many locations. Of course, we all know, that the algorithm is not explicitly aware (and does not require) of any prior existence of classes. The main idea of the iterative method is in sharing only the parts, that are related to the already public reference dataset. As the paper shows, this is enough to co-orient classes that are spread across locations. We consider this approach plausibly private as most of the information about individual samples was discarded. Extensive tests on MNIST and COIL-20 datasets demonstrate the usefulness of the approach and high quality of the obtained embeddings over a variety of settings. Meanwhile, Our consistent results over multi-site biomedical datasets establish the applicability of our approach in this field where privacy-preservation is a major concern. Although multi-shot dSNE is quite robust to various conditions and settings, such as changes in the number of sites, rare or missing data etc., the best performance is achieved when the reference dataset is dense. Notably, the single-shot dSNE, which is mostly an implementation for t-SNE of the previously existed landmark point method, tends to ignore the differences across the sites as there is no direct or indirect contact among local sites. It only relies on

reference data which basically influences the overall results. In contrast, our multi-shot approach provides enough information propagation among remote and local sites which finally enables better embeddings. Yet, all that is being exchanged is pertinent to the already public reference data. An alternative solution—an average of the gradients weighted by the quality of their respective local t-SNE—can be another remarkable method for our decentralized system. Yet, it is not immediately clear how one should approach this. Using this clustering measures on location specific data to weight each Y may bias the results toward good local groupings but unacceptable overall embedding, as we see in the single-shot dSNE. Evaluation of the overall result per iteration (using this metrics, or equivalently the ones we have already used in the paper) is unable to convey the contribution of each site to give it a proper weight. Because in the most general setting we are unable to know a priori, what data each site contains since the algorithm fully preserves site's privacy. Given these difficulties, we have to leave the problem for future, but indeed, interesting and important work. Finally, We conclude that dSNE is a valuable quality control tool for virtual consortia working with private data in decentralized analysis setups.

Acknowledgement

This work was supported by NIH R01DA040487; SP was in part supported by NSF IIS-1318759

References

[1] A. Halevy, P. Norvig, F. Pereira, The unreasonable effectiveness of data, IEEE Intelligent Systems 24 (2) (2009) 8–12.

[2] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, 2016.

[3] S. R. Panta, R. Wang, J. Fries, R. Kalyanam, N. Speer, M. Banich, K. Kiehl, M. King, M. Milham, T. D. Wager, et al., A tool for interactive data visualization: Application to over 10,000 brain imaging and phantom mri data sets, Frontiers in neuroinformatics 10.

[4] J. W. Sammon Jr, A nonlinear mapping for data structure analysis, Computers, IEEE Transactions on 100 (5) (1969) 401–409.

[5] P. Demartines, J. Hérault, Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets, IEEE Transactions on neural networks 8 (1) (1997) 148–154.

[6] G. Hinton, S. Roweis, Stochastic neighbor embedding, in: NIPS, Vol. 15, 2002, pp. 833–840.

[7] J. B. Tenenbaum, V. De Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, science 290 (5500) (2000) 2319–2323.

[8] K. Q. Weinberger, L. K. Saul, An introduction to nonlinear dimensionality reduction by maximum variance unfolding, in: AAAI, Vol. 6, 2006, pp. 1683–1686.

[9] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science (5500) (2000) 2323–2326.

[10] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural computation 15 (6) (2003) 1373–1396.

[11] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (2579-2605) (2008) 85.

[12] N. Bushati, J. Smith, J. Briscoe, C. Watkins, An intuitive graphical visualization technique for the interrogation of transcriptome data, Nucleic acids research (2011) gkr462.

[13] B. A. Malin, K. E. Emam, C. M. O'Keefe, Biomedical data privacy: problems, perspectives, and recent advances, Journal of the American Medical Informatics Association 20 (1) (2013) 2–6.

[14] A. Gaye, Y. Marcon, J. Isaeva, P. LaFlamme, A. Turner, E. M. Jones, J. Minion, A. W. Boyd, C. J. Newby, M.-L. Nuotio, et al., DataSHIELD: taking the analysis to the data, not the data to the analysis, International journal of epidemiology 43 (6) (2014) 1929–1944.

[15] K. W. Carter, R. W. Francis, K. Carter, R. Francis, M. Bresnahan, M. Gissler, T. Grønborg, R. Gross, N. Gunnes, G. Hammond, et al., Vipar: a software platform for the virtual pooling and analysis of research data, International journal of epidemiology (2015) dyv193.

[16] S. M. Plis, A. D. Sarwate, D. Wood, C. Dieringer, D. Landis, C. Reed, S. R. Panta, J. A. Turner, J. M. Shoemaker, K. W. Carter, et al., Coinstac: a privacy enabled model and prototype for leveraging and processing decentralized brain imaging data, Frontiers in Neuroscience 10.

[17] P. M. Thompson, J. L. Stein, S. E. Medland, D. P. Hibar, A. A. Vasquez, M. E. Renteria, R. Toro, N. Jahanshad, G. Schumann, B. Franke, et al., The ENIGMA consortium: large-scale collaborative analyses of neuroimaging and genetic data, Brain imaging and behavior 8 (2) (2014) 153–182.

[18] D. K. Saha, V. D. Calhoun, S. R. Panta, S. M. Plis, See without looking: joint visualization of sensitive multi-site datasets, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 2672–2678.

[19] M. Heikkilä, E. Lagerspetz, S. Kaski, K. Shimizu, S. Tarkoma, A. Honkela, Differentially private bayesian learning on distributed data, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 3226–3235.

[20] V. Rastogi, S. Nath, Differentially private aggregation of distributed time-series with transformation and encryption, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10, ACM, 2010, pp. 735–746.

[21] C. Li, P. Zhou, L. Xiong, Q. Wang, T. Wang, Differentially private distributed online learning, IEEE Transactions on Knowledge and Data Engineering 30 (8) (2018) 1440–1453.

[22] Z. Ji, X. Jiang, S. Wang, L. Xiong, L. Ohno-Machado, Differentially private distributed logistic regression using private and public data, BMC Medical Genomics 7 (1) (2014) S14.

[23] A. Rajkumar, S. Agarwal, A differentially private stochastic gradient descent algorithm for multiparty classification, in: N. D. Lawrence, M. Girolami (Eds.), Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, Vol. 22 of Proceedings of Machine Learning Research, PMLR, La Palma, Canary Islands, 2012, pp. 933–941.

[24] M. Pathak, S. Rane, B. Raj, Multiparty differential privacy via aggregation of locally trained classifiers, in: J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, A. Culotta (Eds.), Advances in Neural Information Processing Systems 23, Curran Associates, Inc., 2010, pp. 1876–1884.

[25] A. Globerson, G. Chechik, F. Pereira, N. Tishby, Euclidean embedding of co-occurrence data 8 (Oct) (2007) 2265–2295.

[26] V. De Silva, J. B. Tenenbaum, Sparse multidimensional scaling using landmark points, Tech. rep., Technical report, Stanford University (2004).

[27] V. De Silva, J. B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, Advances in neural information processing systems (2003) 721–728.

[28] Y. LeCun, C. Cortes, C. J. Burges, The mnist database of handwritten digits (1998).

[29] S. A. Nene, S. K. Nayar, H. Murase, Columbia object image library (coil-20), Tech. Rep. CUCS-005-96, Department of Computer Science, Columbia University (February 1996).

[30] F. X. Castellanos, A. Di Martino, R. C. Craddock, A. D. Mehta, M. P. Milham, Clinical applications of the functional connectome, Neuroimage 80 (2013) 527–540.

[31] D. Hall, M. F. Huerta, M. J. McAuliffe, G. K. Farber, Sharing heterogeneous data: the national database for autism research, Neuroinformatics 10 (4) (2012) 331–339.

[32] M. Ivory, Federal interagency traumatic brain injury research (fitbir) bioinformatics platform for the advancement of collaborative traumatic brain injury research and analysis, in: 143rd APHA Annual Meeting and Exposition (October 31-November 4, 2015), APHA, 2015.

[33] D. Davies, D. Bouldin, A cluster seperation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1 (2).

[34] J. C. Dunn, A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters, Journal of Cybernetics 3 (3) (1974) 32–57.

[35] M. Halkidi, M. Vazirgiannis, Quality scheme assessment in the clustering process, in: Proc. PKDD (Principles and Practice of Knowledge in databases). Lyon, France. Lecture Notes in Artificial Intelligence. Spring –Verlag Gmbh, Vol. 1910, 2000, pp. 265–279.

[36] A. Raftery, A note on bayes factors for log-linear contingency table models with vague prior information, Journal of the Royal Statistical Society 48 (2) (1986) 249–250.

[37] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65.

[38] E. Rendón, I. M. Abundez, C. Gutierrez, S. D. Zagal, A. Arizmendi, E. M. Quiroz, H. E. Arzate, A comparison of internal and external cluster validation indexes, in: Proceedings of the 2011 American Conference on Applied Mathematics and the 5th WSEAS International Conference on Computer Engineering and Applications, AMERICAN-MATH'11/CEA'11, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2011, pp. 158–163.

[39] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Transactions on Knowledge Discovery from Data (TKDD) 6 (1) (2012) 3.

[40] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto, et al., The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism, Molecular psychiatry 19 (6) (2014) 659–667.

[41] T. L. Jernigan, T. T. Brown, D. J. Hagler, N. Akshoomoff, H. Bartsch, E. Newman, W. K. Thompson, C. S. Bloss, S. S. Murray, N. Schork, D. N. Kennedy, J. M. Kuperman, C. McCabe, Y. Chung, O. Libiger, M. Maddox, B. Casey, L. Chang, T. M. Ernst, J. A. Frazier, J. R. Gruen, E. R. Sowell, T. Kenet, W. E. Kaufmann, S. Mostofsky, D. G. Amaral, A. M. Dale, The pediatric imaging, neurocognition, and genetics (ping) data repository, NeuroImage 124 (2016) 1149 – 1154, sharing the wealth: Brain Imaging Repositories in 2015.

[42] Y. Du, Z. Fu, J. Sui, S. Gao, Y. Xing, D. Lin, M. Salman, M. A. Rahaman, A. Abrol, J. Chen, E. Hong, P. Kochunov, E. A. Osuch, V. D. Calhoun, Neuromark: an adaptive independent component analysis framework for estimating reproducible and comparable fmri biomarkers among brain disorders, medRxivarXiv:https://www.medrxiv.org/content/early/2019/10/16/19008631.full.pdf.

[43] J. Ashburner, K. J. Friston, Unified segmentation, NeuroImage 26 (3) (2005) 839 – 851.

[44] S. A. Meda, N. R. Giuliani, V. D. Calhoun, K. Jagannathan, D. J. Schretlen, A. Pulver, N. Cascella, M. Keshavan, W. Kates, R. Buchanan, T. Sharma, G. D. Pearlson, A large scale (n=400) investigation of gray matter differences in schizophrenia using optimized voxel-based morphometry, Schizophrenia Research 101 (1) (2008) 95 – 105.

[45] Z. Fu, A. Caprihan, J. Chen, Y. Du, J. C. Adair, J. Sui, G. A. Rosenberg, V. D. Calhoun, Altered static and dynamic functional network connectivity in

alzheimer's disease and subcortical ischemic vascular disease: shared and specific brain connectivity abnormalities, Human Brain Mapping 40 (11) (2019) 3203–3221. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/hbm.24591.

[46] L. Sweeney, k-anonymity: A model for protecting privacy1, 2013.

[47] S. L, C. M, B.-S. M., Sharing sensitive data with confidence: The datatags system, 2015.

[48] K. W. Carter, R. W. Francis, the International Collaboration for Autism Registry Epidemiology, K. Carter, R. Francis, M. Bresnahan, M. Gissler, T. Grønborg, R. Gross, N. Gunnes, G. Hammond, M. Hornig, C. Hultman, J. Huttunen, A. Langridge, H. Leonard, S. Newman, E. Parner, G. Petersson, A. Reichenberg, S. Sandin, D. Schendel, L. Schalkwyk, A. Sourander, C. Steadman, C. Stoltenberg, A. Suominen, P. Surén, E. Susser, A. Sylvester Vethanayagam, Z. Yusof, ViPAR: a software platform for the Virtual Pooling and Analysis of Research Data, International Journal of Epidemiology 45 (2) (2015) 408–416. arXiv:http://oup.prod.sis.lan/ije/article-pdf/45/2/408/24170123/dyv193.pdf.

[49] P. Thompson, J. L. Stein, S. E. Medland, D. Hibar, A. Arias-Vásquez, M. E. Renteria, R. Toro, N. Jahanshad, G. Schumann, B. Franke, M. Wright, N. G. Martin, I. Agartz, M. Alda, S. Alhusaini, L. Almasy, J. Almeida, K. Alpert, N. C. Andreasen, W. Drevets, The enigma consortium: Large-scale collaborative analyses of neuroimaging and genetic data, Brain Imaging and Behavior 8.

[50] A. Sarwate, S. Plis, J. Turner, M. Arbabshirani, V. Calhoun, Sharing privacy-sensitive access to neuroimaging and genetics data: a review and preliminary validation, Frontiers in Neuroinformatics 8 (2014) 35.

[51] S. Wang, X. Jiang, Y. Wu, L. Cui, S. Cheng, L. Ohno-Machado, Expectation propagation logistic regression (explorer): Distributed privacy-preserving online model learning, Journal of Biomedical Informatics 46 (3) (2013) 480 – 496.

[52] Privacy-preserving ai in medical imaging: Federated learning, differential privacy, and encrypted computation.

[53] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, S. Bakas, Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation, CoRR abs/1810.04304. arXiv:1810.04304.

[54] Privacy-preserving ai in medical imaging: Federated learning, differential privacy, and encrypted computation.

[55] H. B. McMahan, E. Moore, D. Ramage, B. A. y Arcas, Federated learning of deep networks using model averaging, CoRR abs/1602.05629. arXiv:1602.05629.

[56] N. Lewis, S. Plis, V. Calhoun, Cooperative learning: Decentralized data neural network, in: 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 324–331.

[57] B. T. Baker, R. F. Silva, V. D. Calhoun, A. D. Sarwate, S. M. Plis, Large scale collaboration with autonomy: Decentralized data ica, in: 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), 2015, pp. 1–6.

[58] J. Ming, E. Verner, A. Sarwate, R. Kelly, C. Reed, T. Kahleck, R. Silva, S. Panta, J. Turner, S. Plis, V. Calhoun, Coinstac: Decentralizing the future of brain imaging analysis, F1000Research 6 (1512).

[59] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated learning with non-iid data (2018). arXiv:1806.00582.

[60] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, D. Liu, Loadaboost:loss-based adaboost federated machine learning on medical data (2018). arXiv:1811.12629.

Algorithm 7 `singleshotDSNE`

Input:

Objective parameters: ρ (perplexity)

Optimization parameters: T, η, α

Shared Data: $\mathbf{X}_s = [\boldsymbol{x}_1^s, \boldsymbol{x}_2^s \dots \boldsymbol{x}_{N_s}^s], \boldsymbol{x}_i^s \in \mathbb{R}^n$

Data at site $p \forall p$: $\mathbf{X}_p = [\boldsymbol{x}_1^p, \boldsymbol{x}_2^p \dots \boldsymbol{x}_{N_p}^p], \boldsymbol{x}_i^p \in \mathbb{R}^n$

Output: $\mathbf{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_N\}, \boldsymbol{y}_i \in \mathbb{R}^m, m << n, N = \sum_p N_p + N_s$

$\mathbf{Y}_s \leftarrow$ `tSNE` $\mathbf{X}^s$, ρ, T, η, α ▷ At the master node

for $p = 0$ to P **do**

  $\mathbf{Y}_s^{\rightarrow p}$

  Run `tSNE` on $[\mathbf{X}_p, \mathbf{X}_s]$ ▷ At local site p

  At each iteration only update $\mathbf{Y}_p$ ▷ At local site p

end for

$\mathbf{Y} \leftarrow []$ ▷ At the master

for $p = 0$ to P **do** ▷ At the master

  $\mathbf{Y}_p^{\leftarrow p}$

  $\mathbf{Y} \leftarrow [\mathbf{Y}, \mathbf{Y}_p]$

end for

$\mathbf{Y} \leftarrow [\mathbf{Y}, \mathbf{Y}_s]$

Appendix A. Single-shot dSNE

For Single shot d-SNE (Algorithm 7) we at first pass the reference data from centralized site C to each local site.

Now each local site's data consists of two portions. One is its local dataset, for which we need to preserve privacy, and another one is the shared reference dataset both comprise the combined datasets. Each local site runs the t-SNE algorithm on this combine data and produces an embedding into a low dimensional space. However, while computing each iteration of tSNE a local site computes gradient based on combined data, but it only updates the embedding vectors $\boldsymbol{y}$ for local dataset. The embedding for the shared data has been precomputed at the master node and shared with each local site. Similarly to the landmark points approach of [26] our method uses reference points to tie together data from multiple sites. In practice the samples in the shared dataset are not controlled by the researchers using our method, and it is hard to assess the usefulness of each sample in the shared data in advance. In the end each local site obtains an embedding of its data together with the embedding of the shared dataset. Since the embedding points of the shared dataset did not change, all local embeddings

are easily combined by aligning the points representing the shared data.

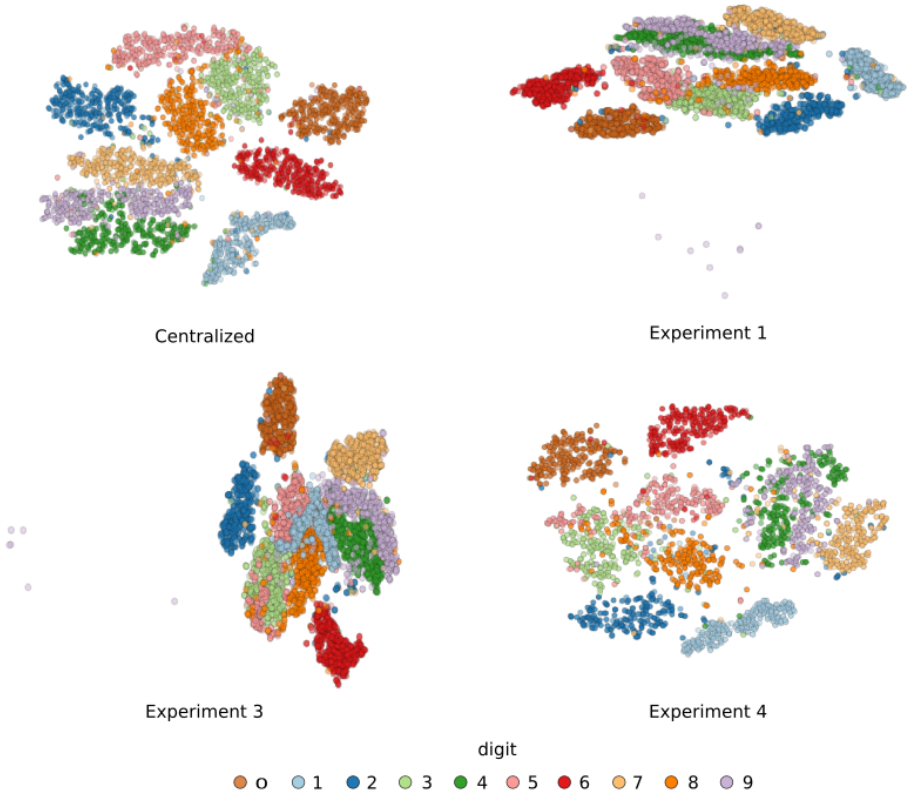


Figure A.9: Single-shot dSNE layout of MNIST data. Single-shot was run for the experiment of 1, 3 and 4 of MNIST dataset. For all experiments, we are able to embed and group same digits from different sites with-out passing any site info to others. Here every digit is marked by a unique color. Centralized - is the original tSNE solution for locally grouped data. Digits are correctly grouped into clusters but these clusters tend to heavily overlapped.