

RESEARCH ARTICLE

Privacy-preserving quality control of neuroimaging datasets in federated environments

Debrata K. Saha^{1,2}  | Vince D. Calhoun^{1,2,3} | Yuhui Du² | Zening Fu² |
Soo Min Kwon⁴ | Anand D. Sarwate⁴ | Sandeep R. Panta² | Sergey M. Plis^{2,3} 

¹Georgia Institute of Technology, Atlanta, Georgia, USA

²Tri-Institutional Center for Translational Research in Neuroimaging and Data Science (TReNDS), Georgia State University, Georgia Institute of Technology, and Emory University, Atlanta, Georgia, USA

³Georgia State University, Atlanta, Georgia, USA

⁴Rutgers, The State University of New Jersey, New Brunswick, New Jersey, USA

Correspondence

D. K. Saha, Georgia Institute of Technology, Atlanta, GA 30318, USA.
Email: dsaha34@gatech.edu

Funding information

National Institutes of Health, Grant/Award Numbers: 1R01DA040487, R01DA049238, R01MH121246, 2R01EB006841, 2RF1MH121885; National Science Foundation, Grant/Award Number: 2112455; Center of Biomedical Research Excellence (COBRE), Grant/Award Numbers: P20GM103472, 5P20RR021938; National Center for Research Resources; Function BIRN, Grant/Award Number: U24-RR021992; Stavros Niarchos Foundation; Leon Levy Foundation; NIMH, Grant/Award Numbers: R03MH096321, K23MH087770

Abstract

Privacy concerns for rare disease data, institutional or IRB policies, access to local computational or storage resources or download capabilities are among the reasons that may preclude analyses that pool data to a single site. A growing number of multi-site projects and consortia were formed to function in the federated environment to conduct productive research under constraints of this kind. In this scenario, a quality control tool that visualizes decentralized data in its entirety via global aggregation of local computations is especially important, as it would allow the screening of samples that cannot be jointly evaluated otherwise. To solve this issue, we present two algorithms: decentralized data stochastic neighbor embedding, dSNE, and its differentially private counterpart, DP-dSNE. We leverage publicly available datasets to simultaneously map data samples located at different sites according to their similarities. Even though the data never leaves the individual sites, dSNE does not provide any formal privacy guarantees. To overcome that, we rely on differential privacy: a formal mathematical guarantee that protects individuals from being identified as contributors to a dataset. We implement DP-dSNE with AdaClip, a method recently proposed to add less noise to the gradients per iteration. We introduce metrics for measuring the embedding quality and validate our algorithms on these metrics against their centralized counterpart on two toy datasets. Our validation on six multisite neuroimaging datasets shows promising results for the quality control tasks of visualization and outlier detection, highlighting the potential of our private, decentralized visualization approach.

KEYWORDS

federated neuroimaging, fMRI, quality control, sMRI

1 | INTRODUCTION

Even though the availability of public data continues to increase, there are still many “unsharable,” private datasets which arise multiple challenges for machine learning systems (Plis, Sarwate, Dylan,

D. K. Saha, V. D. Calhoun, and S. M. Plis contributed equally to this study.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2022 The Authors. *Human Brain Mapping* published by Wiley Periodicals LLC.

et al., 2016). The importance of operating on decentralized sensitive data and, as a result, of (virtually) pooling very large-scale neuroimaging datasets is exemplified by the success of the ENIGMA project (Thompson Paul, Stein Jason, Sarah, et al., 2014). The continuing growth in the value of large and diverse neuroimaging datasets should inevitably increase the demand for similar decentralized consortia. Multiple large consortia, such as the Global Imaging Genetics in Adolescents (GIGA) consortium (Gunter, Vivek, Chunshui, et al., 2019), are already leveraging decentralized approaches. Several decentralized systems are being developed to virtually pool and facilitate computation on distributed datasets, for example COINSTAC (Plis, Sarwate, Dylan, et al., 2016) and others (Amadou, Yannick, Julia, et al., 2014; Carter Kim et al., 2015; Carter Kim, Francis Richard, Carter, et al., ; Shuang, Jiang Xiaoqian, Yuan, Samuel, & Lucila, 2013). For all of them, quality control is essential.

Intuitive visualization of the complete virtual dataset that is physically spread across multiple locations is a much-needed tool for filtering out participating sites with bad data, detecting incorrect processing, or identifying mistakes in the input process. For example, consider a magnetic resonance image (MRI) data sample that consists of the entire brain, containing on the order of 100,000 volumetric pixels (voxels) (Scott, 2014). Outliers in smaller datasets at consortium sites make statistical analyses of the consortium data much more difficult. One solution is to develop methods for quality control of large-scale brain imaging data. Since it is challenging to scan through each data sample, an effective method of quality control is to simultaneously embed multiple samples onto a lower dimensional space for visualization. These visualizations have been shown to be useful tools to assess and monitor data quality, while revealing interesting relationships (Panta Sandeep, Runtang, Jill, et al., 2016). Beyond data quality, we can also use these approaches to visualize relationships among groups (e.g., diagnostic categories) or continuous measures (such as disease severity or cognitive performance) (Plis, Devon, Ruslan, et al., 2014).

A common way of visualizing a dataset consisting of multiple high-dimensional data points is to embed them into a 2- or 3-dimensional space. Existing methods like principal component analysis (PCA) (Hotelling, 1933) can be useful for revealing the linear structure of data. However, the nonlinearity of biomedical data makes analysis with PCA difficult, failing to preserve and convey the hidden structure within the data. To resolve this issue, many other methods, including Sammon mapping (Sammon Jr, 1969), curvilinear component analysis (Pierre & Jeanny, 1997), stochastic neighbor embedding (Geoffrey & Sam, 2003), isomap (Tenenbaum Joshua, Vin, & Langford John, 2000), maximum variance unfolding (Weinberger & Saul, 2006), locally linear embedding (Roweis Sam & Saul, 2000), and Laplacian Eigenmaps (Mikhail & Partha, 2003) were developed to embed and visualize nonlinear datasets. These methods perform well on artificial data, but can struggle in real high-dimensional settings due to their inability to retain local and global structure in a single map. Several methods have been proposed to overcome these problems as well. To visualize underlying structure and intrinsic transitions in high-dimensional biological data, an approach that is highly scalable both in memory and runtime, called potential of heat diffusion for affinity-based transition embedding

(PHATE), was recently introduced (Kevin, David, Zheng, et al., 2019). Other notable methods include t-distributed stochastic neighbor embedding (t-SNE) (Laurens & Geoffrey, 2008), viSNE (El-Ad, Kara, Michelle, et al., 2013), and hierarchical stochastic neighbor embedding (HSNE) (van Unen, Höllt, Pezzotti, et al., 2017). Lastly, to reduce dimensions and overcome computational restrictions, Uniform Manifold Approximation and Projection (UMAP) (Leland, John, Nathaniel, & Lukas, 2018) was proposed and has proven to be effective in the field of bioinformatics. However, all of these methods were built on the principle that the datasets are locally accessible. If the data samples were distributed across multiple sites, the sites would have to pool their data to a single site for analysis.

In this paper, we propose decentralized stochastic neighbor embedding (dSNE), an algorithm that embeds a high-dimensional, decentralized dataset into a 2D map for subsequent visualization and inspection. Our approach improves on our preliminary adopts the method of embedding multiple modalities into the same Euclidean space based on their co-occurrence statistics (Amir, Gal, Fernando, & Naftali, 2007). Since we cannot physically pool all of the data to a single local site, we use publicly available anonymized datasets as a common reference and build the overall embedding around it. This approach is most similar to the method of landmark points, previously used for improving computational efficiency (Vin & Joshua, 2004; Vin & Tenenbaum Joshua, 2003). Our approach, dSNE, significantly extends the original landmark points approach, using t-SNE as the base of our algorithm. Our method can be seen as a dynamic modification that can embed data points into a common space after capturing the relationship among samples distributed across different locations. It significantly improves on our prior work (Saha, Calhoun, Panta, & Plis, 2017) by using multiple iterations to improve the embedding. Even though dSNE provides a way to visualize data in a decentralized manner, it comes with no formal privacy guarantees. To remedy this, we also propose an (ϵ, δ) -differentially private version of dSNE (DP-dSNE). Differential privacy is a framework which quantifies the privacy risk to individuals when functions of their data are released to untrusted parties. DP-dSNE adds noise to the gradients of the private and shared data per iteration, using a method called AdaClip. We evaluate our DP-dSNE algorithm using Rényi Differential Privacy (Ilya, 2017) and present a privacy analysis using the moments accountant to keep track of the privacy loss per iteration. To evaluate and compare the performance with the centralized version in controlled settings, we demonstrate both of our algorithms (dSNE and (ϵ, δ) -DP dSNE) on multiple datasets. We also introduce a novel performance metric of overlap and roundness to quantify the quality of our embeddings. Lastly, we apply our approaches to six different multisite neuroimaging datasets, showing that our methods can capture information and perform quality control of distributed datasets producing highly pragmatic visualizations.

2 | METHODS

In the centralized problem of data embedding, we are given the task of producing a dataset of N samples $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$, where $\mathbf{y}_i \in \mathbb{R}^m$, from

a dataset $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$, where $\mathbf{x}_i \in \mathbb{R}^n$, such that $m \ll n$. For the convenience of visualization, m is usually set to $m = 2$. The goal is for the embedding \mathbf{Y} to give a “faithful” embedding of the data \mathbf{X} in the sense that similar points in \mathbf{X} will be mapped to close points in \mathbf{Y} .

2.1 | Background: t-SNE

In t-SNE, the distances between the points in \mathbf{Y} must be as close to the distances between the corresponding points in \mathbf{X} , where preserving the closeness of nearby points is weighted more heavily than far points (Laurens & Geoffrey, 2008). In the first step, t-SNE converts the high-dimensional Euclidean distances between datapoints into conditional probabilities, called pairwise affinities, that represent similarities between data points (see Algorithm 1). The algorithm takes a scalar parameter called the *perplexity* ρ . To compute similarity of a datapoint x_j to datapoint x_i , the algorithm first computes the weight of x_j given by a Gaussian kernel centered at x_i with bandwidth (variance) $\sigma_i(\rho)^2$, where it identifies the value of σ_i separately for each datapoint by performing a binary search across a range of values until it can match the user-specified perplexity. The similarity is a conditional probability distribution p_{ji} formed by renormalizing the N likelihoods into a probability mass function:

$$p_{ji} = \begin{cases} 0 & j = i \\ \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i(\rho)^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i(\rho)^2)} & j \neq i \end{cases} \quad (1)$$

These similarities can be gathered into an $N \times N$ matrix $\tilde{\mathbf{P}}$ and the affinity is the symmetrized matrix $\mathbf{P} = (\tilde{\mathbf{P}} + \tilde{\mathbf{P}}^\top) / (2N)$. This makes \mathbf{P} a matrix representing a joint distribution on pairs of data points.

For the low-dimensional representation \mathbf{Y} , we compute pairwise weights in a similar way, except that this time for the joint distribution q_{ij} we use the Student's t distribution with one degree of freedom (or a Cauchy distribution) instead of a Gaussian:

$$q_{ij} = \begin{cases} 0 & j = i \\ \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|\mathbf{y}_k - \mathbf{y}_i\|^2)^{-1}} & j \neq i \end{cases} \quad (2)$$

These can be gathered into a matrix \mathbf{Q} representing a joint distribution for the embedded points. Algorithm 2 outlines the full t-SNE

Algorithm 1 PairwiseAffinities

Input: ρ (perplexity), $\mathbf{X} \in \mathbb{R}^{N \times n}$

Output: $\mathbf{P} \in \mathbb{R}^{N \times N}$

Equation (1) to compute p_{ji} with perplexity ρ

Set $p_{ij} = (p_{ji} + p_{ij}) / (2N)$ for all i, j

procedure. To embed the data points into a low-dimensional space, t-SNE tries to minimize the mismatch between distribution \mathbf{P} and \mathbf{Q} in higher and lower dimensional spaces. The algorithm performs gradient descent on the Kullback–Leibler (KL) divergence (or relative entropy) between the joint distribution \mathbf{P} and the joint distribution \mathbf{Q} :

$$J(\mathbf{Y}) = \sum_i \sum_{j \neq i} p_{ij} \ln \frac{p_{ij}}{q_{ij}} \quad (3)$$

The gradient of the Kullback–Leibler divergence between \mathbf{P} and the Student's t based joint probability distribution \mathbf{Q} is expressed in Equation (4),

$$\frac{\partial J}{\partial \mathbf{y}_i} = 4 \sum_j (p_{ij} - q_{ij}) (\mathbf{y}_i - \mathbf{y}_j) \left(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2\right)^{-1} \quad (4)$$

Inspired by the overall satisfactory performance of t-SNE on a range of tasks, we use it as the base of our decentralized algorithm.

2.2 | Proposed method: dSNE

In the decentralized setting, the privacy and sensitivity of datasets often preclude the pooling of local data, making computation of distances among samples across different sites difficult. Without these distances (see Equation (1)), we cannot obtain a common embedding. Fortunately, in neuroimaging (and many other fields), there are now multiple large public repositories of MRI data that we can leverage to make this computation feasible (Dan, Huerta Michael, McAuliffe Matthew, & Farber, 2012; Mary, 2015; Xavier, Adriana, Cameron, Mehta Ashesh, & Milham, 2013).

In the decentralized setting we have L sites where each site ℓ has (local) data $\mathbf{X}^\ell = [\mathbf{x}_1^\ell, \mathbf{x}_2^\ell, \dots, \mathbf{x}_{N_\ell}^\ell]$ consisting of N_ℓ vectors \mathbf{x}^ℓ in \mathbb{R}^n . In

Algorithm 2 tSNE

Input:

Data: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N], \mathbf{x}_i \in \mathbb{R}^n$

Scaling parameter: ρ (perplexity)

Optimization parameters: T (number of iterations), η (learning rate), α (momentum)

Output: $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}, \mathbf{y}_i \in \mathbb{R}^m, m \ll n$

$\{p_{ij}\} = \text{PairwiseAffinities}(\rho, \mathbf{X})$

$\mathbf{Y} \propto N(0, 10^{-4}\mathbf{I}), \mathbf{I} \in \mathbb{R}^{m \times m}$ initialize from Gaussian

for $i = 1$ to T do

Equation (2) to compute low-dimensional affinities q_{ij}

Equation (4) to compute $\partial J / \partial \mathbf{y}_i(t - 1)$

$\mathbf{y}_i(t) = \mathbf{y}_i(t - 1) + \eta(\partial J / \partial \mathbf{y}_i) + \alpha(t)(\mathbf{y}_i(t - 1) - \mathbf{y}_i(t - 2))$

end for

addition, we have a shared data set $\mathbf{X}^s = [\mathbf{x}_1^s, \mathbf{x}_2^s, \dots, \mathbf{x}_{N_s}^s]$. The goal is to produce embeddings $\{\mathbf{Y}^\ell\}$ and \mathbf{Y}^s , where $\mathbf{Y}^\ell = [\mathbf{y}_1^\ell, \mathbf{y}_2^\ell, \dots, \mathbf{y}_{N_\ell}^\ell]$ for each ℓ and $\mathbf{Y}^s = [\mathbf{y}_1^s, \mathbf{y}_2^s, \dots, \mathbf{y}_{N_s}^s]$ contain vectors in \mathbb{R}^m where $m \ll n$. Typically we will consider $m = 2$ to produce 2D visualizations of the data set. As in t-SNE, we want the distances close \mathbf{x} points to remain close in the embedding of \mathbf{y} points. We assume all sites have access to the shared data \mathbf{X}^s and its embedding \mathbf{Y}^s and can modify \mathbf{Y}^s when they update locally.

We implemented three algorithms: (1) Single-shot dSNE, (2) Multi-shot dSNE, and (3) Differentially private multi-shot dSNE, which we describe in the following subsections. Detailed procedure and experimental results of Single-shot dSNE are provided in our prior work (Saha et al., 2017), in C, and Figure 14.

2.3 | Multi-shot dSNE

For multi-shot dSNE, we pass messages iteratively between the local sites and central site in rounds. At time t , the centralized site passes the reference embedding $\mathbf{Y}^s(t-1)$ from the previous iteration to each of the local sites. At this point each site ℓ has \mathbf{X}^ℓ , past values of $\mathbf{Y}^\ell(t-j)$ for $j = 1, 2, \dots, t$, the reference data set \mathbf{X}^s , and the updated embedding $\mathbf{Y}^s(t-1)$. Each local site then computes the gradient update (Algorithm 3) using a ‘‘momentum’’ approach that combines information from the past two iterations. The result are updates $\mathbf{Y}^\ell(t)$ and $\mathbf{Y}^{s,\ell}(t)$ for the local and shared data embeddings, respectively. The sites send their new embeddings of the local data $\mathbf{Y}^{s,\ell}(t)$ to the central site, which averages them to form $\hat{\mathbf{Y}}^s$ and sends that back to the local sites.

The local sites then update their local and shared embeddings and compute the average of all embedding to help recenter. They send this average to the coordinator, who averages across sites and sends back a global mean that sites use to center their local and shared embeddings to get $\mathbf{Y}^{\ell}(t)$ and $\mathbf{Y}^{s}(t)$ for the next iteration. The pseudocode and overall procedure for multi-shot dSNE are shown in Algorithm 4. Note, at each iteration, the embedded vector \mathbf{Y} for the shared dataset will be the same at all of the local sites. This ensures that the local values of different sites are influenced by the same and

common reference data at each iteration. [Correction added on 24 March 2022, after first online publication: Repeated text deleted. Text appears at end of section 2.4.]

2.4 | Differentially private multi-shot dSNE

We begin by reiterating the setup of multi-shot dSNE. In dSNE, there are n local sites (each with their own disjoint dataset) that would like to collaborate to learn a global structure among the data samples. However, the sensitivity of biomedical data prevent centralized analyses that pool all of the data to a single site. Even though the local data samples never leave the sites in dSNE, since the embeddings of the shared data are influenced by the local data points, there is room for a potential privacy leak. To resolve this, we introduce DP-dSNE, a differentially private dSNE algorithm that formally guarantees privacy. We now define differential privacy (Cynthia, Frank, Kobbi, & Adam, 2006) and the AdaClip algorithm (Venkatadheeraj, Ananda, Yu Felix, Sashank, & Sanjiv, 2019).

We say that two datasets D, D' are *neighboring* datasets if they differ by one data entry. A randomized mechanism $\mathcal{A}: \mathcal{D} \rightarrow \mathcal{R}$ is said to be (ϵ, δ) -differentially private if for all neighboring databases $D, D' \in \mathcal{D}$, and any measurable set $S \subset \mathcal{R}$, we have

$$\mathbb{P}(\mathcal{A}(D) \in S) \leq e^\epsilon \mathbb{P}(\mathcal{A}(D') \in S) + \delta, \quad (5)$$

where $\mathbb{P}(B)$ is the probability of the event B and the probability is taken over the randomness in the mechanism $\mathcal{A}(\cdot)$.

We use the shorthand notation (ϵ, δ) -DP for (ϵ, δ) -differentially private. A standard method to preserve privacy of a function is to add noise, where the variance of the noise is proportional to the sensitivity of the function. Mathematically, we define the ℓ_2 global sensitivity of a function f as

$$\Delta_2(f) = \max_{D, D'} \|f(D) - f(D')\|_2. \quad (6)$$

Given that the ℓ_2 sensitivity of a function f is Δ , one way to preserve privacy is to add Gaussian noise (Cynthia & Aaron, 2014) of variance $\Delta^2 \sigma^2$, such that $\mathcal{A}(D) = f(D) + \mathcal{N}(0, \Delta^2 \sigma^2)$. Then, if we choose σ to be $\frac{\Delta}{\epsilon} \sqrt{2 \log \frac{1.25}{\delta}}$, each iteration of the algorithm is (ϵ, δ) -DP (Cynthia & Aaron, 2014; Martin, Andy, Ian, et al., 2016). However, in practice, finding a priori bound on the size of the gradients (i.e., the sensitivity of the gradients) is difficult, and often does not exist. In literature, one way to solve this issue is to bound the gradients by clipping each gradient in ℓ_2 norm for a clipping threshold C (Martin et al., 2016). This clipping would ensure that the sensitivity of the gradients change by at most C . Although this is a plausible method, clipping all of the gradients to a fixed value of C can often add more noise than needed, as the size of the gradients generally grow smaller during training (Algorithms 5 and 6).

To add less noise per iteration, we adopt in using AdaClip (Venkatadheeraj et al., 2019), a recently proposed method to

Algorithm 3 LocalGradStep

Input:

Data embeddings: $\mathbf{Y}^\ell(t-1)$, $\mathbf{Y}^\ell(t-2)$ (local), $\mathbf{Y}^s(t-1)$,
 $\mathbf{Y}^s(t-2)$ (shared), $\mathbf{P} \in \mathbb{R}^{(N_\ell + N_s) \times (N_\ell + N_s)}$

Optimization parameters: η, α

Output: $\hat{\mathbf{Y}}^\ell(t)$ (local), $\hat{\mathbf{Y}}^s(t)$ (shared)

Equation (2) on $[\mathbf{Y}^\ell(t), \mathbf{Y}^s(t)]$ to compute low-dimensional
 affinities q_{ij}

Equation (4) to compute $\partial J / \partial \mathbf{y}_i(t-1)$

$\hat{\mathbf{y}}_i(t) = \eta(\partial J / \partial \mathbf{y}_i(t-1)) + \alpha(\mathbf{y}_i(t-1) - \mathbf{y}_i(t-2))$

Group $\{\hat{\mathbf{y}}_i(t)\}$ into $\hat{\mathbf{Y}}^\ell(t)$ (local) and $\hat{\mathbf{Y}}^s(t)$ (shared)

Algorithm 4 multishotDSNE

Input:

Objective parameters: ρ (perplexity)Optimization parameters: T, η, α Shared Data: $\mathbf{X}^s = [\mathbf{x}_1^s, \mathbf{x}_2^s, \dots, \mathbf{x}_{N_s}^s], \mathbf{x}_i^s \in \mathbb{R}^n$ Local Data: $\mathbf{X}^\ell = [\mathbf{x}_1^\ell, \mathbf{x}_2^\ell, \dots, \mathbf{x}_{N_\ell}^\ell], \mathbf{x}_i^\ell \in \mathbb{R}^n, \ell = 1, 2, \dots, L$ **Output:** $\{\mathbf{Y}^\ell: \ell = 1, 2, \dots, L\}, \mathbf{Y}^s$ Sample $\mathbf{Y}^s(0)$ i.i.d. from $\mathcal{N}(0, 10^{-4} \mathbf{I}_m)$ \triangleright Initialize from GaussianCoordinator sends $\mathbf{X}^s, \mathbf{Y}^s(0)$ to all sites**for** $\ell = 0$ to L **do** \triangleright Initialize at sites $\mathbf{P}^\ell = \text{PairwiseAffinities}(\rho, [\mathbf{X}_p, \mathbf{X}_s])$ Sample $\mathbf{Y}^\ell(0), \mathbf{Y}^\ell(-1)$ from $\mathcal{N}(0, 10^{-4} \mathbf{I}), \mathbf{I} \in \mathbb{R}^{m \times m}$ **end for****for** $t = 1$ to T **do**Coordinator sends $\mathbf{Y}^\ell(t-1)$ to all sites**for** $\ell = 1$ to L **do** $\hat{\mathbf{Y}}^\ell(t), \hat{\mathbf{Y}}^{s,\ell}(t) = \text{LocalGradStep}$ $(\mathbf{Y}^\ell(t-1), \mathbf{Y}^\ell(t-2), \mathbf{Y}^s(t-1), \mathbf{Y}^s(t-2), \eta, \alpha)$ Site ℓ sends $\hat{\mathbf{Y}}^{s,\ell}$ to Coordinator**end for** $\hat{\mathbf{Y}}^s = \frac{1}{L} \sum_{\ell=1}^L \hat{\mathbf{Y}}^{s,\ell}$ \triangleright Average local shared embeddingsCoordinator sends $\hat{\mathbf{Y}}^s$ to all sites**for** $p = 0$ to P **do** \triangleright At local sites $\tilde{\mathbf{Y}}^\ell = \mathbf{Y}^\ell(t-1) + \hat{\mathbf{Y}}^\ell$ $\tilde{\mathbf{Y}}^s = \mathbf{Y}^s(t-1) + \hat{\mathbf{Y}}^s$ $\bar{\mathbf{Y}} = \frac{1}{N_\ell + N_s} \left(\sum_{i=1}^{N_\ell} \tilde{\mathbf{Y}}_i^\ell + \sum_{i=1}^{N_s} \tilde{\mathbf{Y}}_i^s \right)$ \triangleright Mean embeddingSend $\bar{\mathbf{Y}}$ to Coordinator**end for** $\bar{\mathbf{y}} = \frac{1}{L} \sum_{\ell=1}^L \bar{\mathbf{Y}}^\ell$ Coordinator sends $\bar{\mathbf{y}}$ to all sites**for** $\ell = 1$ to L **do** \triangleright Recenter the embeddingsSet $\mathbf{y}_i^\ell(t) = \tilde{\mathbf{Y}}_i^\ell - \bar{\mathbf{y}}$ for all i Set $\mathbf{y}_i^s(t) = \tilde{\mathbf{Y}}_i^s - \bar{\mathbf{y}}$ for all i **end for****end for**

[Correction added on 24 March 2022, after first online publication: Expression has been corrected in Algorithm 4]

adaptively clip and add noise based on the size of the gradients. By introducing AdaClip into our multi-shot dSNE algorithm, we can preserve the privacy of the individuals in the dataset. The modification that we need in order to make dSNE (ϵ, δ)-DP is to replace LocalGradStep in Algorithm 4 with DP-LocalGradStep, as shown in Algorithm 7. There are two things to note about the DP algorithm: (1) DP-LocalGradStep brings two extra parameters, m and s that we use and update at every iteration and (2) even though we are computing a global mean based on each local site's mean, each step is still differentially private due to postprocessing invariance

Algorithm 5 noiseAddition

Input:

Gradients: g^t (gradient at iteration t)Noise Parameters: m^t, b^t, σ (noise scale)**Output:** \tilde{g}^t (privacy-preserving approximation of g^t)Compute transformed gradient: $w^t = \frac{g^t - m^t}{b^t}$ Clip transformed gradient: $\hat{w}^t = \frac{w^t}{\max(1, \|w^t\|_2)}$ Add noise to gradient: $\tilde{w}^t = \hat{w}^t + \mathcal{N}(0, \sigma^2 \mathbf{I})$ Rescale the gradient: $\tilde{g}^t = b^t \tilde{w}^t + m^t$ **Algorithm 6 AdaClip**

Input:

Gradients: g^t (gradient at iteration t)Noise Parameters: m^t, s^t, σ (noise scale)Parameters: $h_1, h_2, \beta_1, \beta_2$ **Output:** \tilde{g}^t (gradient at iteration t), m^{t+1}, s^{t+1} (updated noise parameters)**for** $i = 0$ to n **do** $b_i^t = \sqrt{s_i^t} \cdot \sqrt{\sum_{i=1}^n s_i^t}$ **end for**Noise addition to gradients: $\tilde{g}^t = \text{noiseAddition}(g^t, m^t, b^t, \sigma)$ Update m^t : $m^{t+1} = \beta_1 m^t + (1 - \beta_1) \tilde{g}^t$ Compute variance v^t : $v^t = \min(\max(\| \tilde{g}^t - m^t \|^2 - (b_i^t)^2 \sigma^2, h_1), h_2)$ Update s^t : $s^{t+1} = \beta_2 (s_i^t)^2 + (1 - \beta_2) v^t$ **Algorithm 7 DP-LocalGradStep**

Input:

Data embeddings: $\mathbf{Y}^\ell(t-1), \mathbf{Y}^\ell(t-2)$ (local), $\mathbf{Y}^s(t-1), \mathbf{Y}^s(t-2)$ (shared), $\mathbf{P} \in \mathbb{R}^{(N_\ell + N_s) \times (N_\ell + N_s)}$ Optimization parameters: η, α **Output:** $\hat{\mathbf{Y}}^\ell(t)$ (local), $\hat{\mathbf{Y}}^s(t)$ (shared)Equation (2) on $[\mathbf{Y}_\ell(t), \mathbf{Y}_s(t)]$ to compute low-dimensional affinities q_{ij} Equation (4) to compute $g_i^{t-1} = \partial J / \partial \mathbf{y}_i(t-1)$ $\tilde{g}^{t-1}, m^t, s^t = \text{AdaClip}(\tilde{g}^{t-1}, m^{t-1}, s^{t-1})$ $\hat{\mathbf{y}}_i(t) = \eta (\tilde{g}_i^{t-1} + \alpha (\mathbf{y}_i(t-1) - \mathbf{y}_i(t-2)))$ Group $\{\hat{\mathbf{y}}_i(t)\}$ into $\hat{\mathbf{Y}}^\ell(t)$ (local) and $\hat{\mathbf{Y}}^s(t)$ (shared)

(Cynthia & Aaron, 2014). In the next section, we provide a privacy analysis using Rényi Differential Privacy and the moments accountant to keep track of the total privacy loss during training (Martin et al., 2016).

2.5 | Privacy analysis of differentially private multi-shot dSNE

In this section, we analyze the privacy loss of our differentially private multi-shot dSNE algorithm using Rényi Differential Privacy (RDP) (Ilya, 2017). In order to define Rényi Differential Privacy, we first need to define the Rényi divergence. Let P and Q be two probability distributions over a set \mathcal{X} . Then, the Rényi divergence of order α for $\alpha > 1$ is defined as

$$D_\alpha(P\|Q) = \frac{1}{\alpha-1} \log \mathbb{E}_Q \left[\frac{P(x)}{Q(x)} \right]^\alpha. \quad (7)$$

Rényi Differential Privacy uses the Rényi divergence on the privacy loss random variable (i.e., the ratio between the distributions over the neighboring datasets) to naturally “relax” the privacy parameters. Formally, a randomized mechanism $\mathcal{A}: \mathcal{D} \rightarrow \mathcal{R}$ is said to be (α, γ) -Rényi differentially private if for all $D, D' \in \mathcal{D}$ with $D \sim D'$ and any measurable set $S \subset \mathcal{R}$ the following holds:

$$D_\alpha(P_D\|P_{D'}) \leq \gamma, \quad (8)$$

where $P_D(y)$ is the distribution of the mechanism $\mathcal{A}(D)$. We can keep track of the privacy loss of the DP-dSNE algorithm in terms of RDP and then convert the result into an (ϵ, δ) -DP bound. Recall that in our decentralized setup, each local site has data that is disjoint from the other local sites. Thus, it is sufficient to analyze the privacy loss random variable of a single site. We start our analysis of the DP-dSNE algorithm by reviewing some definitions and connections between RDP and DP.

If \mathcal{A} is a randomized algorithm satisfying (α, γ) -RDP, then it also satisfies $(\gamma + \frac{\log \frac{1}{\delta}}{\alpha-1}, \delta)$ -DP for any $0 < \delta < 1$. Further, if \mathcal{A} has \mathcal{L}_2 sensitivity Δ , then the Gaussian mechanism $\mathcal{A}(D) + Z$, where $Z \sim \mathcal{N}(0, \sigma^2)$, satisfies $(\alpha, \frac{\alpha \Delta}{2\sigma^2})$ -RDP (Ilya, 2017). We now need to derive the sensitivity of our gradients from using AdaClip to translate our RDP bound to an (ϵ, δ) -DP bound. [Correction added on 24 March 2022, after first online publication: $Z \sim \mathcal{N}(0, \sigma^2)$ has been corrected to $Z \sim \mathcal{N}(0, \sigma^2)$]

Let $g^t = [g_1^t, g_2^t, \dots, g_n^t]$ and $g'^t = [g_1'^t, g_2'^t, \dots, g_n'^t]$ be gradient vectors of two neighboring datasets at iteration t . Using AdaClip, we have two additional vectors for each dataset: $m^t = [m_1^t, m_2^t, \dots, m_n^t]$ and $b^t = [b_1^t, b_2^t, \dots, b_n^t]$. By the definition of the sensitivity of the gradients of neighboring databases, we have

$$1 = \max_{D \sim D'} \|w^t - w'^t\|_\infty, \quad (9)$$

$$= \max_{D \sim D'} \left\| \frac{g^t - m^t}{b^t} - \frac{g'^t - m^t}{b^t} \right\|_\infty, \quad (10)$$

$$= \max_{D \sim D'} \left\| \frac{g^t - g'^t}{b^t} \right\|_\infty, \quad (11)$$

$$b_{\max} = \max_{D \sim D'} \|g^t - g'^t\|_\infty, \quad (12)$$

where $\|x\|_\infty = \max_i |x_i|$ (i.e., the \mathcal{L}_∞ norm a vector x with i entries) and $b_{\max} = \|b^t\|_\infty$. Since the transformed gradient w^t is clipped at norm 1, its sensitivity is also 1. Following this analysis and from the gradient transformation of AdaClip, we obtain that the sensitivity of the gradient g^t is b_{\max} .

By using the conversion definitions stated previously, if J denotes the number of iterations, the DP-dSNE algorithm satisfies $(\frac{\alpha J}{2\sigma^2} + \frac{\log \frac{1}{\delta}}{\alpha-1}, \delta)$ -DP, where σ^2 is the variance of the noise, $0 < \delta < 1$, and α is

$$\alpha = 1 + \sqrt{\frac{2}{b_{\max} J} \sigma^2 \log \frac{1}{\delta}}. \quad (13)$$

Note that in the DP conversion, ϵ is directly proportional to α and J , which indicate that as the number of iterations increase, the overall ϵ will also increase, guaranteeing less privacy. This is an inherent trade-off that arises across all differentially private algorithms, and hence it is important to keep track of how ϵ changes throughout the course of our algorithm by using Equation (13). At each iteration, we keep track of b_{\max} in order to compute our (ϵ, δ) -DP guarantee. Note that in our privacy analysis, we defined our (ϵ, δ) -DP guarantee in terms of RDP, but we can also easily derive these values using the moments accountant (Martin et al., 2016), similar to that of the analysis done by Imtiaz et al. (Imtiaz, Mohammadi, Silva, et al., 2019), since RDP and the moments accountant are equivalent notions. In the following sections, we make comparisons between RDP and the moments accountant by demonstrating how ϵ changes for a fixed value of δ for DP-dSNE.

2.6 | Comparison metrics

To measure the performance and quality of clustering, several metrics have been proposed, including the Davies-Bouldin (DB) index (Davies & Bouldin, 1979), Dunn index (Dunn, 1974), quality index (Halkidi, Vazirgiannis, Batistakis, & Zighed Djamel, 2000), Bayesian information criterion (BIC) index (Raftery, 1986), and the silhouette coefficient (Rousseeuw, 1987). Other metrics such as F-measure, entropy, purity, and rand index can also be used as an external comparison metric.

In the distributed setting, it is important to make sure that the embedding of a data point gets clustered into its correct corresponding class. To test the cluster qualities of our decentralized algorithms, we use two a priori labeled datasets: (1) the MNIST dataset and (2) the COIL-20 dataset, whose clusters are known and described (Laurens & Geoffrey, 2008). We consider each label as the known number of clusters as our ground truth. We introduce three new validation techniques: (1) K-means ratio, (2) Intersection area, and (3) Roundness.

1. The k-means criterion (or ratio) is the ratio of intra and inter-cluster distances between clusters. Mathematically, this is defined as

$$\alpha = \frac{\sum_{d=0}^n \sum_{s \in \mathcal{X}_d} \|\mu_d - x_s^d\|_2}{\sum_{(i,j), (i>j), (i \neq j)} \|\mu_i - \mu_j\|_2}, \quad (14)$$

where n is the number of clusters. We can interpret this ratio as the following: the numerator indicates intra cluster distance, in which a smaller value shows that each distinct cluster is tightly bounded. A larger denominator value implies that the inter clusters are well separated from each other. Thus, if the α value is generally small, we can conclude that the algorithm performs well.

2. The intersection area can be seen as computing the overlap between two clusters. We first remove the outliers (see Figure 1) (Tony, Ming, & Zhi-Hua, 2012) to compute the convex hull for each group and use them to measure the overlap. For the intersection area, we find the area of each polytope(cluster) and find the total area by summing them. After that we compute the overlap area between each cluster. In specific, this is done by summing all of the polytope areas minus the area of the union of all the polytopes, normalized by the area of the union. The lower value of intersection area determines the formation of good cluster with less overlap.
3. Roundness is the ratio of the area of each polytope to the area of the circumscribed circle. To compute the roundness of each cluster, we first represent each cluster as a cloud with a convex hull. To remove the effect of the differences in perimeters, we normalize the perimeters. This normalization effectively approximates the process of making the perimeters equal. Finally, we compute the area of the polytope and use the area as our measure of roundness. The higher roundness metric value determines the good quality of the cluster. More precisely, it quantifies how the samples are distributed around the mean of a given cluster.

3 | DATA

We base our experiments on eight datasets:

1. MNIST dataset¹ (Yann & Corinna, 2010).
2. COIL-20 dataset² (Nene, Nayar, & Hiroshi, 1996).
3. Autism Brain Imaging Data Exchange (ABIDE) fMRI dataset³ (Di Martino, Yan, Li, et al., 2014).
4. Pediatric Imaging Neurocognition Genetics (PING) dataset⁴ (Jernigan, Brown, Hagler, et al., 2016).
5. Structural Magnetic Resonance Imaging (sMRI) dataset.
6. Function Biomedical Informatics Research Network (fBIRN) structural MRI dataset (Yuhui, Zening, Sui, et al., 2019).
7. Bipolar and Schizophrenia Network for Intermediate Phenotypes (BSNIP) structural MRI dataset (Du Yuhui et al., 2019).
8. Mind Research Network (MRN) fMRI dataset.

MNIST dataset was taken from a Kaggle competition which had 28,000, 28×28 gray-scale images of all 10 handwritten digits. For our experiments, we randomly chose 5,000 different samples from the dataset (while preserving class balance). For centralized t-SNE and dSNE, we preprocessed the dataset by reducing the dimensions of the data samples from 784 to 50 using Principal Component Analysis (PCA).

COIL-20 dataset contains images of $32 \times 32 = 1,024$ pixels of 20 different objects. Each object was placed on a motorized turntable against a black background. Between 0° and 360° , a picture was taken in 5° intervals. Each object was viewed from 72 equally spaced orientations, yielding a total of 1,440 images.

ABIDE fMRI dataset contains data samples of 1,153 subjects accessible through the COINS data exchange.⁵ The ABIDE dataset was preprocessed down to multiple spatial and temporal quality control (QC) measures.⁶ Since this dataset inherently had lower

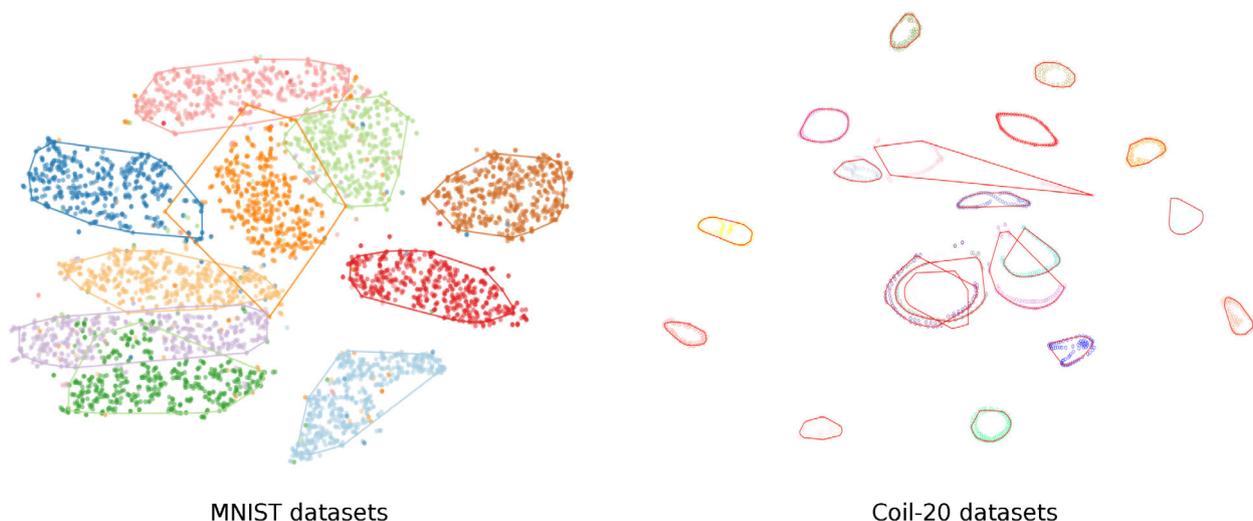


FIGURE 1 A t-SNE output on centralized MNIST and COIL-20 dataset; and outlier-free convex hull boundaries

dimensions, we ran t-SNE and dSNE directly on the dataset without a dimensionality reduction step.

PING is a multisite study containing neural developmental histories, information about developing mental and emotional functions, multimodal brain imaging data, and genotypes for well over 1,000 children and adolescents between the ages of 3–20. We take 632 subject's fMRI data from this dataset for our experiment. The data is preprocessed with SPM5 (John & Friston Karl, 2005) software. It is slice time corrected and warped to the standard MNI brain template from SPM5. This image is used for extracting the data for the experiment. For the first time point, the voxel values at each location from all the brain slices are first added across slices along the Z axis, resulting in a single row vector of size 3,339. This was done to reduce the computational load on the system, and hence improve run time of the proposed algorithm. These voxel values from each image scan served as inputs to the t-SNE and dSNE algorithms.

sMRI scans (3D T1-weighted pulse sequences) are preprocessed through the 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. Gray matter maps are extracted from segmenting the T1 weighted nifti images. The unmodulated gray matter concentration images from the VBM pipeline are registered to the SPM template. In some cases, the non-modulated maps are preferred compared to the modulated maps according to existing literature (Meda et al., 2008). The data from these unmodulated gray matter normalized images is used for this experiment. To reduce the computational load on the system, and hence improve run time of the proposed algorithm, for each scan, the voxel values at each location from all the brain slices are first added across slices, resulting in a matrix size of 91×109 . All of the voxel values from each image scan are converted into a single row vector of size 9,919 for each data point and passed as inputs to the t-SNE and dSNE algorithms.

fBIRN and **BSNIP** datasets used in this study were collected from seven and six imaging sites, respectively. Each subject was selected based on head motion ($\leq 3^\circ$ and ≤ 3 mm) and functional data providing nearly full brain normalization (Fu, Caprihan, Chen, et al., 2019). 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 (Du Yuhui et al., 2019) was adopted to extract reliable intrinsic connectivity networks (ICNs) that were replicated across independent datasets.

MRN fMRI is a multiscanner, and multistudy dataset where the harmonized imaging protocol was used during the acquisition. This dataset consists of 3,910 subjects of echo planar imaging data by following different protocols across multiple sites and studies. For preprocessing, the statistical parametric mapping (SPM5) toolbox was used for slice time correction, motion correction, and spatial normalization. For each of these images, six quality control (QC) metrics are computed from this data and are given as inputs to t-SNE algorithm. As the QC matrices values were low-dimensional, we directly run t-SNE and dSNE without reducing the dimensions.

4 | EXPERIMENTAL SETUP

In this section, we discuss the experiments used to compare centralized t-SNE to dSNE and DP-dSNE. The experiment with the fMRI dataset illustrates how we can use dSNE for outlier detection and quality control. We use the terms “reference” and “shared” data interchangeably throughout the setup. We organize this section into two parts: (1) experiments on t-SNE and dSNE and (2) experiments on t-SNE and DP-dSNE.

4.1 | Experiments with dSNE

4.1.1 | MNIST data

The objective of this experiment is to investigate the adaptability of our algorithm when there is an imbalance in the number of data samples between the local site and central node. This imitates a plausible phenomenon, where the centralized site accumulates and stores more data than the local sites (and perhaps, vice versa). In this experiment, each local site holds samples according only one digit (label), while the reference dataset contains all of the digits (0–9). We consider two cases: (1) each site contains 400 samples while each digit in the reference consists of 100 samples and (2) the inverse case, in which the local sites have only 100 samples, while the reference has 400 samples.

4.1.2 | COIL-20 data

For the COIL-20 dataset, we have two different experiments:

1. Our first experiment is very similar to the experiment of MNIST. Here, we apply the same strategy in which the centralized node holds more data than the local sites. In this experiment, each local site contains only one type of object while the reference dataset contains all of the objects (1–20). We consider two cases: (1) each site contains 52 samples of its corresponding object while each object in the reference dataset consists of 20 samples (2) and the inverse case, when each site holds 20 samples, while the reference objects are represented by 52 samples.
2. Here, we investigate the scenario in which some objects are missing from the shared dataset. Each local site out of 20 contains a single object. We run 10 experiments with different random seeds, where in each run, the reference dataset is missing objects from local sites 16–20. For each of these experiments, we have two conditions: (1) the reference dataset is small (20 samples for each object) while the sites are large (52 samples per site) and (2) the reference dataset is large (52 samples per object) and the dataset of the sites are small (only 20 samples).

4.1.3 | ABIDE fMRI data

To simulate a consortium of multiple sites, we randomly partition the ABIDE dataset into 10 local and one reference dataset. We run three

different dSNE experiments, each corresponding to one random split. We use this dataset to also demonstrate how we can perform quality control of data samples using dSNE. Lastly, we collect all of the data into one local site to simulate a centralized visualization using t-SNE to compare to our dSNE algorithm.

4.1.4 | sMRI data

The sMRI dataset consists of subjects with four different age groups: below 11, 11 to 17, 30 to 34, and above 64. We run a total of three experiments using this dataset:

1. We use each age group as a unique local site (four sites in total) and form reference samples by taking 100 samples from each local site.
2. For this experiment, we keep the local sites the same as the previous setting, but create reference data by taking 100 samples from only the first site.
3. In this case, we randomly take 50 samples from site 2 and place them in site 1 and take 100 samples from site 4 and distribute them equally between sites 2 and 3. This analyzes the effect in which each local site has data samples from the same class.

4.1.5 | PING data

We collect our PING dataset from five different data sources to run four different experiments:

1. For the first experiment, we use each data source (total of five) as a local site and form our reference samples by taking small samples from each site.
2. The second experiment also uses five local sites, but we form the reference dataset by taking 100 samples from only the second site.
3. Similar to Experiment 3 of the sMRI dataset, we randomly take 30 samples from site 2 to place in site 1, take 20 samples from site 3 to place in site 2, take 10 samples from site 2 to place in site 3, and take 10 samples from site 1 to place in site 4. The reference samples are formed by taking small samples from each local site.
4. For this experiment, we keep sites 1, 3, and 4 unchanged from the first experiment, but form reference samples by taking all of the samples from sites 2 and 5. This yields a total of 3 local sites and one reference sample, consisting of data from sites 2 and 5.

4.1.6 | fBIRN and BSNIP data

For the dSNE experiment, as we collected the fBIRN data from seven different sites, we considered each of them as a local site. The BSNIP dataset (collected from six imaging sites) served as our reference dataset. For t-SNE, we treated each dataset as its own and ran t-SNE

separately. We also ran t-SNE on combined datasets (fBIRN + BSNIP) for visual comparisons with dSNE.

4.1.7 | MRN fMRI data

We collected MRN data from three different sites (MRN, Avanto, and Boulder). For the dSNE experiment, we use each data site as its own unique local site (total of three local sites). However, we randomly picked 200 samples from site 2 and placed them in site 1. Sites 2 and 3 had its own corresponding data from Avanto and Boulder, respectively. We formed the reference dataset by picking 1,383 samples from site 1 and 622 samples from site 2. For comparison, we pooled data from all three sites and performed a centralized t-SNE analysis. We ran another experiment where the shared samples only contains the bad scans. The procedure and experimental results are provided in B.

4.2 | Experiments with DP-dSNE

For the DP-dSNE experiment, we used the MNIST and PING datasets to demonstrate the robustness of dSNE even in formal private settings. For the MNIST dataset, we manually created three local sites and one coordinator node to participate in the computation. Each local site holds two classes (digit) from the dataset and the coordinator node contains all classes (four in total). For the PING dataset, we used the same setup as stated in Experiment 1 of section 4.1.5. We compare our results to centralized t-SNE and dSNE and show that our algorithm still provides good utility. We hypothesize that our DP-dSNE algorithm will generalize well to other neuroimaging data.

5 | RESULTS

We organize this section into three parts: (1) comparison between t-SNE and dSNE, (2) comparison between t-SNE and dSNE on biomedical data, and (3) comparison between t-SNE and DP-dSNE.

5.1 | Comparison between t-SNE and dSNE

5.1.1 | MNIST data

Figure 2 represents the results of MNIST dataset. In Figure 2, the plots in each row correspond to the best and worst performing runs, respectively. The first column layout is colored by the digits, representing different clusters. The second column has layouts colored corresponding to sites. The box plots on the left show performance metrics computed by the three proposed metrics. t-SNE was computed on pooled, centralized data and SMALL and LARGE represent smaller and larger sizes of the reference dataset in dSNE runs. The

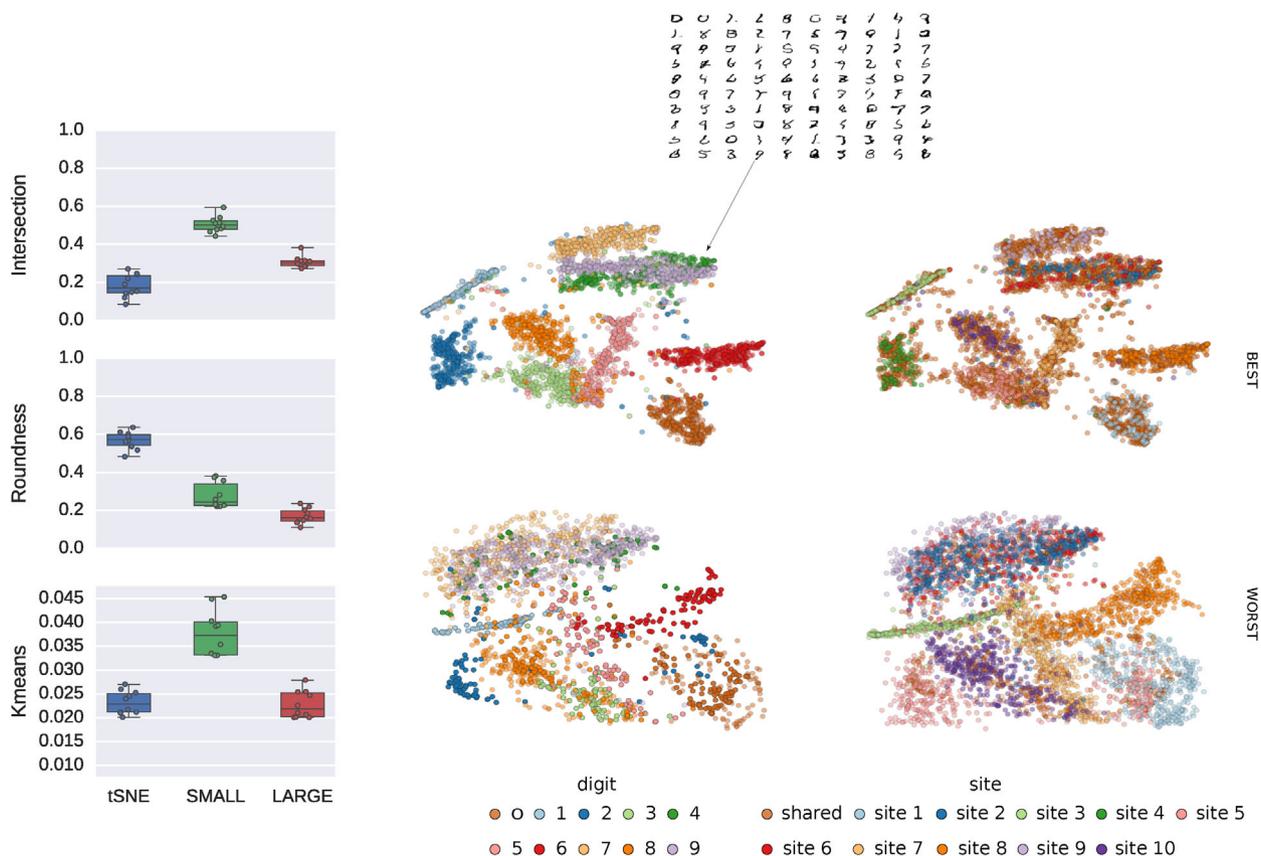


FIGURE 2 MNIST Experiment: Reference data contains samples of all of the MNIST digits, but is either a small or large amount. In the boxplots, tSNE was computed on pooled data and SMALL and LARGE represent smaller and larger reference datasets, respectively. Each row of the plots correspond to the best and worst performing runs. For each experiment, we ran the simulation 10 times with different random seeds. From the 10 experimental results, we picked the best and worst results, labeled as “best” and “worst” run. The left plots correspond to clusters labeled by digits, whereas the right plots correspond to clusters labeled by sites

comparison metrics show that performance is generally better when the shared portion of the data contains a large amount of data. However, the cluster roundness degrades with the size of the sample in the shared data. Thus, we observe this tradeoff between the roundness of the cluster and general performance. Overall, however, we observe that the dSNE clusters are less “round” compared to centralized t-SNE. Some additional results with different experimental settings for the MNIST dataset are provided in our prior work (Saha et al., 2017).

5.1.2 | COIL-20 data

Figure 3 depicts the results of Experiment 1 on the COIL-20 dataset. Similar to that of Figure 2, the rows of the plots correspond to best and worst dSNE runs. The first column has plots colored by each object, whereas the second column has clusters colored by sites. In the boxplots, t-SNE was computed on pooled data and SMALL and LARGE represent smaller and larger sizes of the reference datasets, respectively. The comparison metrics show similar results as the

MNIST experiments. For large amounts of shared data, the comparison metric shows better performance than in the case of smaller samples in the reference dataset.

Figure 4 depicts the results of Experiment 2 of the COIL-20 dataset. The comparison metrics show that we always obtain better results when the reference sample size is larger. For smaller reference samples, we observe highly overlapped clusters for which it is hard to distinguish the clusters for different objects. From this experiment, we observe that it does not affect the performances when the reference samples does not contain all type of objects, but the local sites do.

5.2 | Comparison of t-SNE and dSNE on biomedical data

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

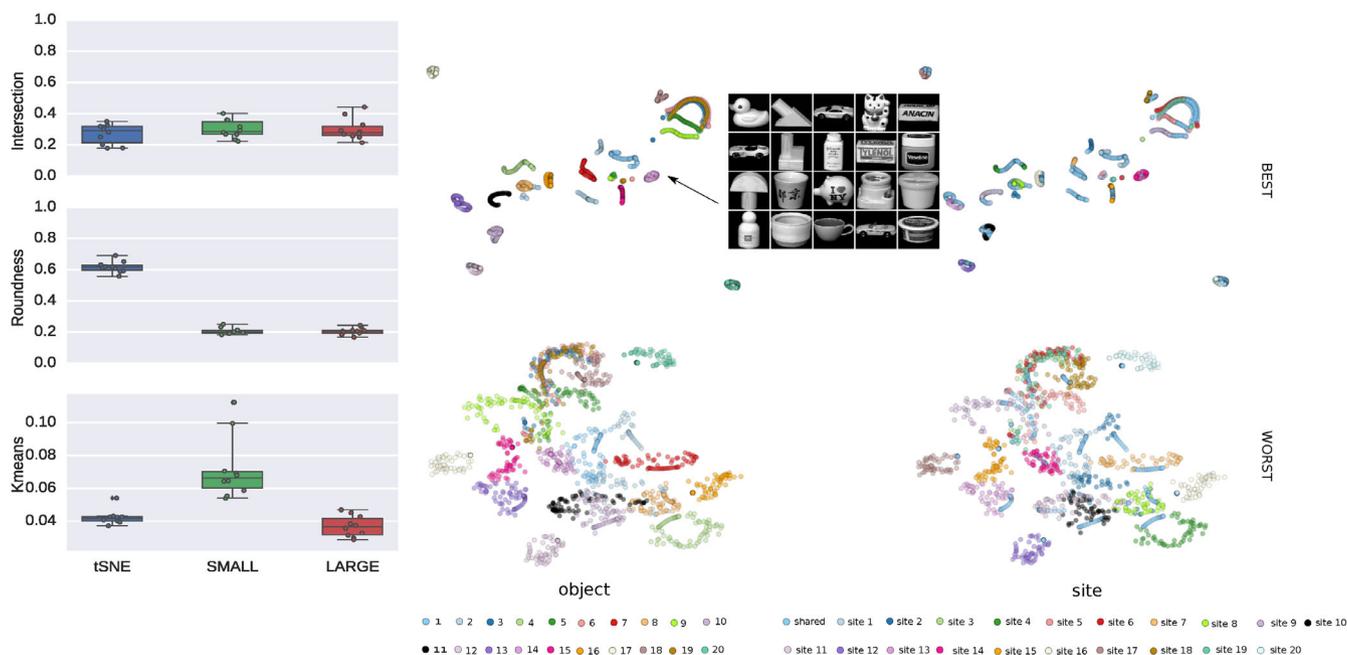


FIGURE 3 COIL-20 Experiment 1: reference data contains samples of all COIL-20 objects but is either in small or large amounts. In the boxplots, tSNE was computed on pooled data and SMALL and LARGE represent smaller and larger reference datasets, respectively. Each row of the plots correspond to the best and worst performing runs. The left plots correspond to clusters labeled by objects, whereas the right plots correspond to clusters labeled by sites. The COIL-20 dataset consist of 20 different objects which is shown in the figure. In the layout, each point represents an object from these 20 objects

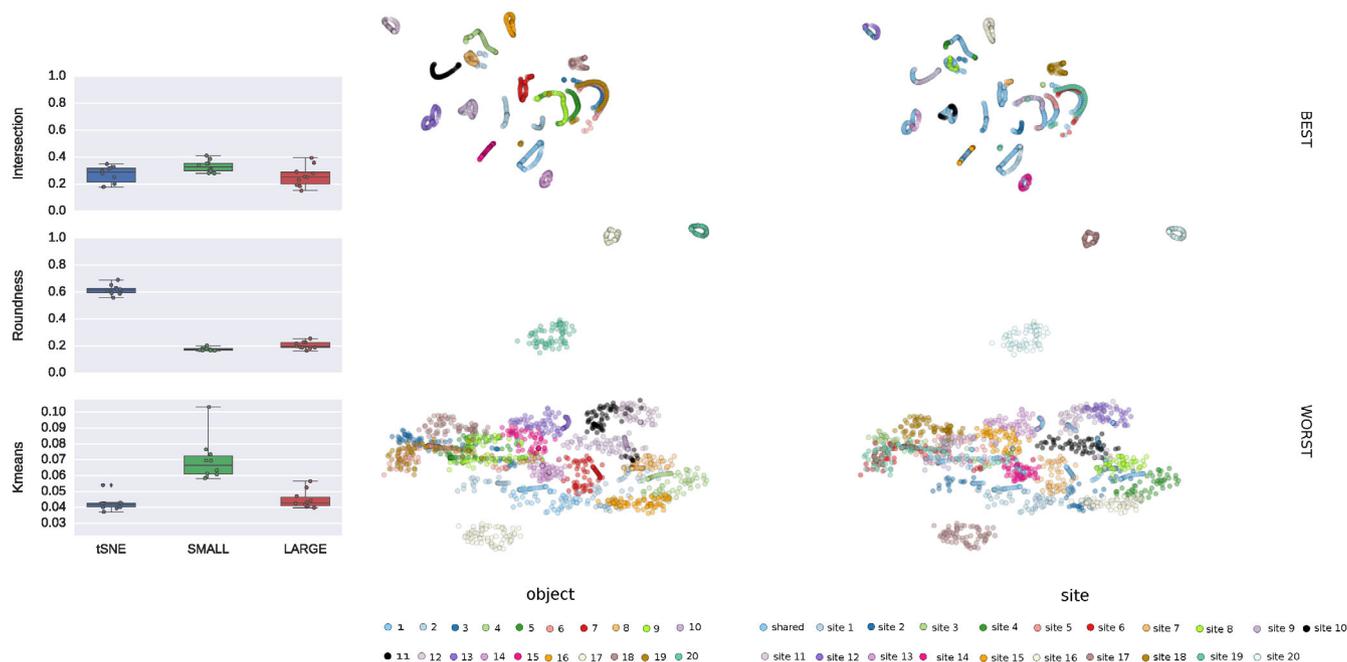


FIGURE 4 COIL-20 Experiment 2: the reference dataset is missing one unique COIL-20 object that is present at one of the local sites. In the boxplots, tSNE was computed on pooled data and SMALL and LARGE represent smaller and larger reference datasets, respectively. Each row of the plots correspond to the best and worst performing runs. The left plots correspond to clusters labeled by objects, whereas the right plots correspond to clusters labeled by sites

5.2.1 | ABIDE fMRI data

The layout of dSNE from the ABIDE dataset is shown in Figure 5. In every experiment, a total of 10 local and one remote sites are

participating in the computation. The result from a centralized t-SNE run shows 10 different clusters. For each of the three random seed experiments of our decentralized simulation, we obtain 10 different clusters as well. In the layout, the samples that belong to the same site

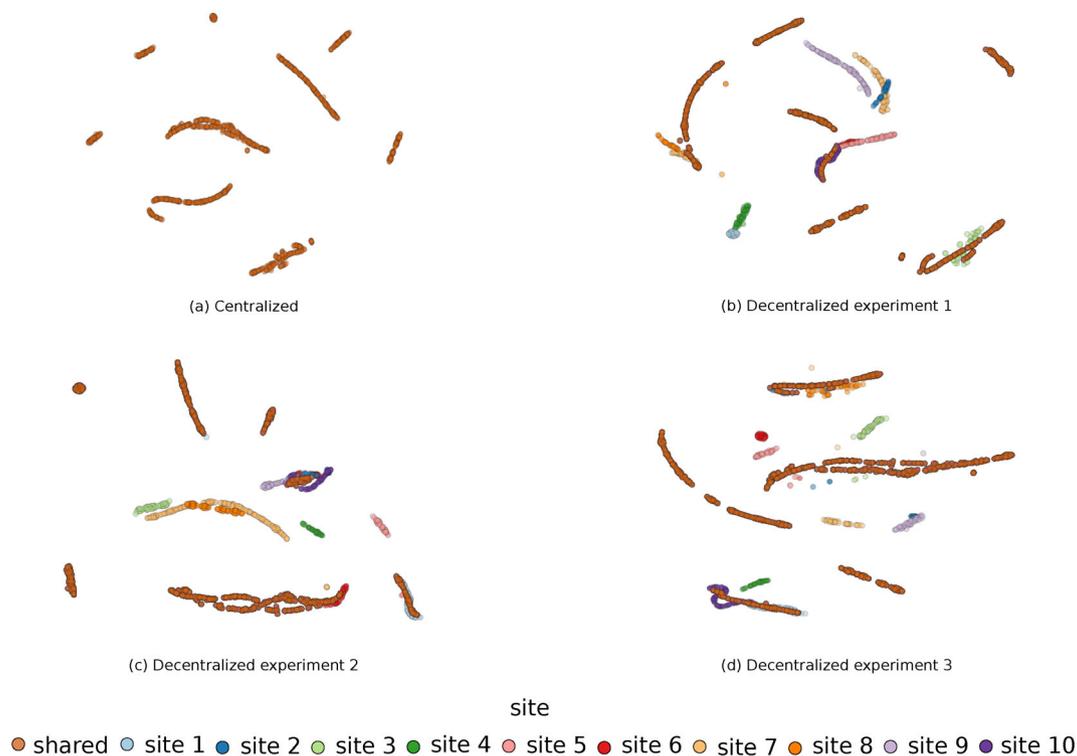


FIGURE 5 Experiment for the QC metrics of the ABIDE dataset. (a) the tSNE layout of pooled data; (b–d) are the dSNE layouts for the three different experiments. In each dSNE experiment, 10 local and a coordinator sites participate in the computation. Similar to tSNE, in the decentralized setup, we get 10 clusters, where each site is marked by unique color

are marked the same color. In Experiment 1, samples from site 1 and 4 are homogeneous. That is why they are grouped together in the final embedding. In each of the experiments, we observe good separation among the clusters and consistent grouping of homogeneous samples, which shows that dSNE produces stable embedding. The experimental results show that when homogeneous samples are distributed in different sites, dSNE can embed and group them together without the direct communication between the sites.

5.2.2 | sMRI data

Figure 6 depicts embeddings produced by dSNE on sMRI data in our three experiments specified in section 4.1.4. In Experiment 2, we obtain poor results, where all clusters are overlapped. There are two possible reasons that may cause this type of result. (1) An unrepresentative reference dataset predictably leads to such a layout. (2) In both t-SNE and dSNE, the simulation begins with a random initialization of low-dimensional Y values. If the initialization is poor (gradient descent may get stuck in a local minimum), the global minimum can sometimes not be obtained. Thus, in this case, we may not get good results. However, in Experiments 1 and 3, results closely resemble those obtained by tSNE. Note that in all successful experiments scans of children younger than 11 formed a distinct separate cluster. Meanwhile the other age groups, although connected together in a single contiguous cluster, are ordered according to the age. Although

the categories are discrete, it may be worth further investigation to inspect whether the age transition is smooth. This is an example of exploratory data analysis, where the structure of the resulting embedding may reveal some inherent regularities in a dataset. The age groups may not be as interesting, but serve as a clear demonstration that it is possible to use dSNE for discovery of data properties not already known to the researcher, similar to (Plis, Devon, Ruslan, et al., 2014). The dSNE can be used as visual and thus quick, intuitive and interpretable quality assurance, outlier detection, assessment of compatibility of datasets, and even assessment of site effects.

5.2.3 | PING data

Figure 7 depicts the experimental results of the PING dataset. In this plot, each point is colored by its respective site. For Experiments 1, 2, and 3, we used five different sites and the reference data was formed by taking samples from each of the sites. In Experiment 4, we used three different sites and formed the reference samples by combining the data from two different sources and marked them by the same color. Among the 632 subjects with age group 3–20, we get four major clusters. Like our previous experiments, here, we observe that the homogeneous samples from the different sites are grouped together. We also find that the number of clusters in the pooled scenario of t-SNE is equivalent to the number of clusters in dSNE on decentralized data.

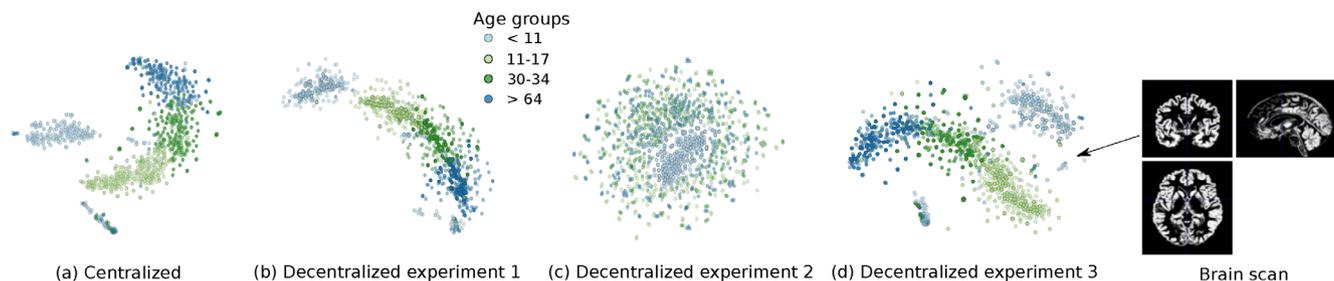


FIGURE 6 Experiment for the QC metrics of the sMRI dataset. (a) t-SNE layout of pooled data. (b–d) are the dSNE layouts for the three different experiments. In all of the experiments, there are four total classes corresponding to an age group each and each class is marked by a unique color. The sMRI dataset consists of brain scans from different age group people and one of the brain scans is shown in the figure. These scans are preprocessed before entering the dSNE algorithm. In the layout, each point represents a single individual

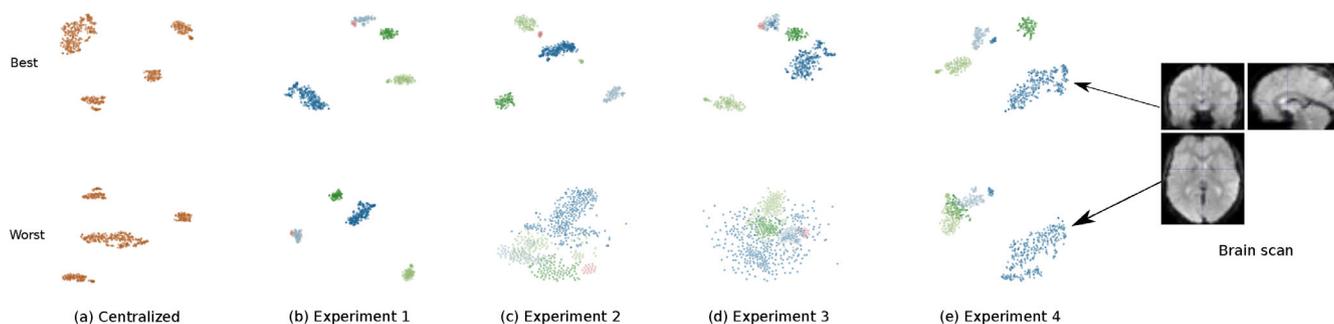


FIGURE 7 Experiment for QC metrics of the PING dataset. (a) The t-SNE layout of the pooled data. In all of the columns, the top figure presents the best performing run, while the lower one represents the worst performing run. (b–e) are the dSNE layouts for four different dSNE experiments. In all of the dSNE experiments, we get four clusters, just like the t-SNE case. The PING dataset consist of brain imaging data of children and adolescents and one of the scans is shown in the figure. These scan data are preprocessed first and give input to our algorithm. In the layout, each point represents a single individual

5.2.4 | fBIRN and BSNIIP data

Figure 8 depicts the results of the fBIRN and BSNIIP experiments. In the first column, we present the t-SNE layout, where the top and bottom plots correspond to fBIRN and BSNIIP, respectively. We obtain relatively well separated groups (HC and SZ) for fBIRN but not for BSNIIP. In the second column, we present the t-SNE layout of combined datasets (fBIRN and BSNIIP), where the top and bottom figures are colored by groups and sites, respectively. We ran t-SNE on the combined datasets, but for the plots, we only show fBIRN subjects. This is because our objective is to see how subjects from the same group of the fBIRN dataset from different sites embed together in lower dimensional space. In the third column, we present the dSNE layout of the combined datasets, where the 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 both t-SNE and dSNE on the combined datasets, we notice that subjects from the HC group are densely clustered in both plots. We obtain less dense clusters for the SZ group, but get embeddings that are similar in both t-SNE and dSNE plots. However, in the dSNE layout, we get more overlaps between clusters compared to t-SNE. We believe this is a reasonable tradeoff, as there is no direct communication between the private local sites.

It is also worth noting that during plotting, some points can be embedded in a very small dense region. To picture this better, we plotted the embeddings colored by sites. To our knowledge, we did not see any evidence of bias in which data are grouped by sites.

5.2.5 | MRN fMRI data

With the MRN data, we demonstrate how we can perform quality control using t-SNE and dSNE visualizations. Figure 9 depicts the experimental results of MRN fMRI data. In this figure, we observe four distinct clusters for both t-SNE on pooled data and dSNE on decentralized data. Having the same number of clusters in these cases highlight the potential of performing decentralized visualization with dSNE. Significantly, we can also observe the poor quality scan samples marked by the red cluster in the layout. The size of the red clusters in both plots are small and separated from the other samples. This shows that these samples are “outliers” from the other samples in the dataset. The main goal of our distributed low-dimensional embedding technique is to measure the quality control of data. From this experiment, we observe that our algorithm can successfully separate the poor quality scans from the whole dataset.

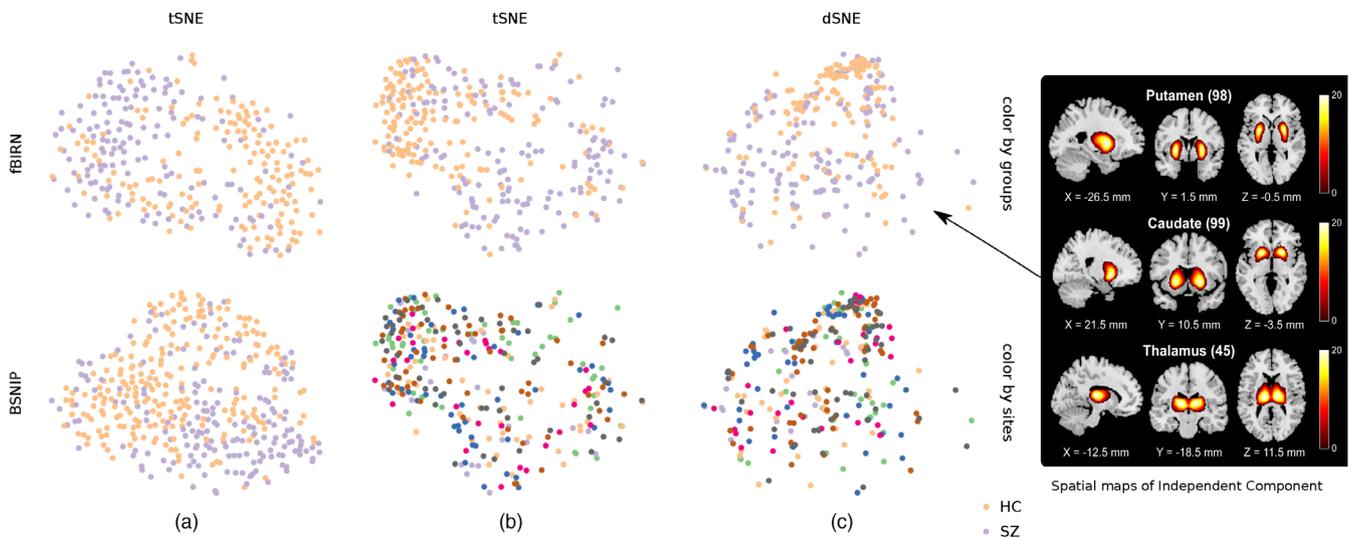


FIGURE 8 Experiment for the QC metrics of fBIRN and BSNIP datasets. (a) Top and bottom plots represent the t-SNE layout of fBIRN and BSNIP, respectively. (b) Top and bottom plots represent t-SNE layouts of combined (fBIRN+BSNIP) datasets but colored by groups and sites, respectively. (c) Top and bottom plots represent dSNE layouts of combined (fBIRN + BSNIP) datasets but colored by groups and sites, respectively. The fBIRN and BSNIP are the brain imaging data of healthy control and Schizophrenia. From these data intrinsic connectivity networks (ICNs) were extracted and used as input to our algorithm. One of the spatial maps of ICNs is shown in the figure

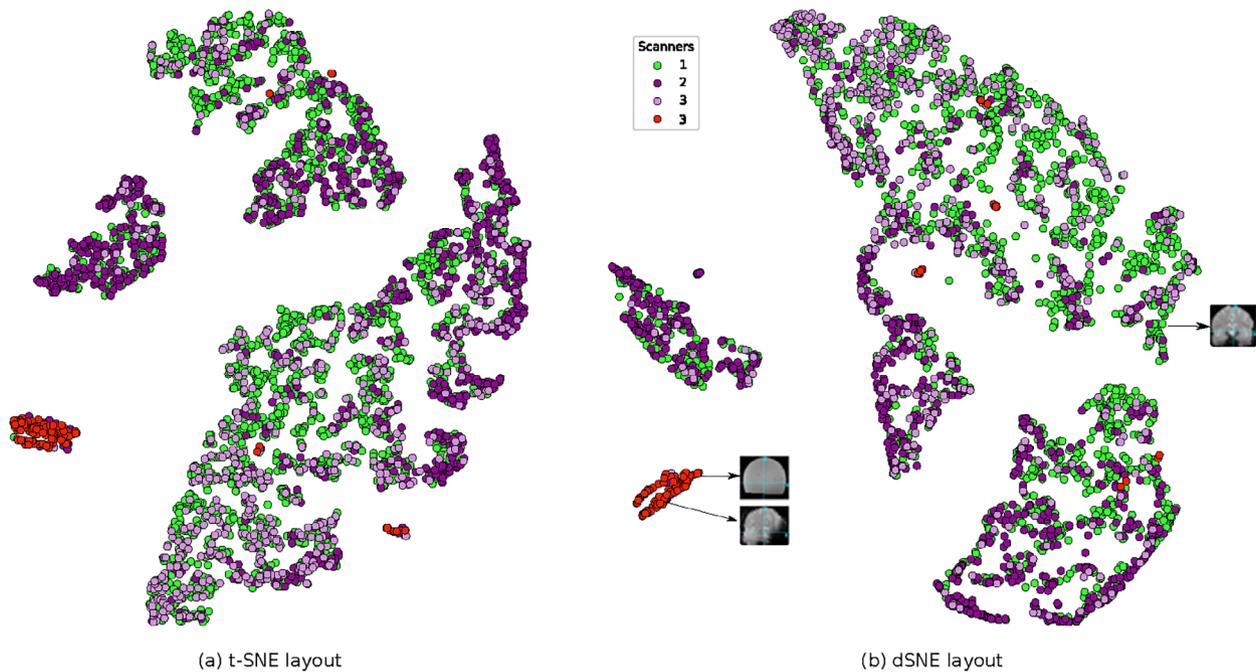


FIGURE 9 Experiment for the QC metrics of the MRN fMRI dataset. (a, b) The layouts colored by Scanners for t-SNE and dSNE, respectively. In both experiments, we get four distinct clusters. Here, we can identify poor quality scan samples marked by the red cluster. In this experiment, three local and one remote sites participated in the computation. In the layout, each point represents a brain scan of an individual

5.3 | Comparison between dSNE and DP-dSNE

Lastly, we make a comparison between t-SNE, dSNE, and DP-dSNE on the MNIST and PING datasets. Figure 10 depicts the experimental results of DP-dSNE after 1,000 iterations. In this figure, the left, middle, and right columns correspond to centralized t-SNE, dSNE and DP-

dSNE, respectively. For all experiments and in all plots, we see four different clusters corresponding to each class. Each cluster is distinct and well-separated among the other clusters. This proves to show the potential of our DP-dSNE algorithm, in which we can provide high utility while providing privacy of sensitive datasets in distributed locations.

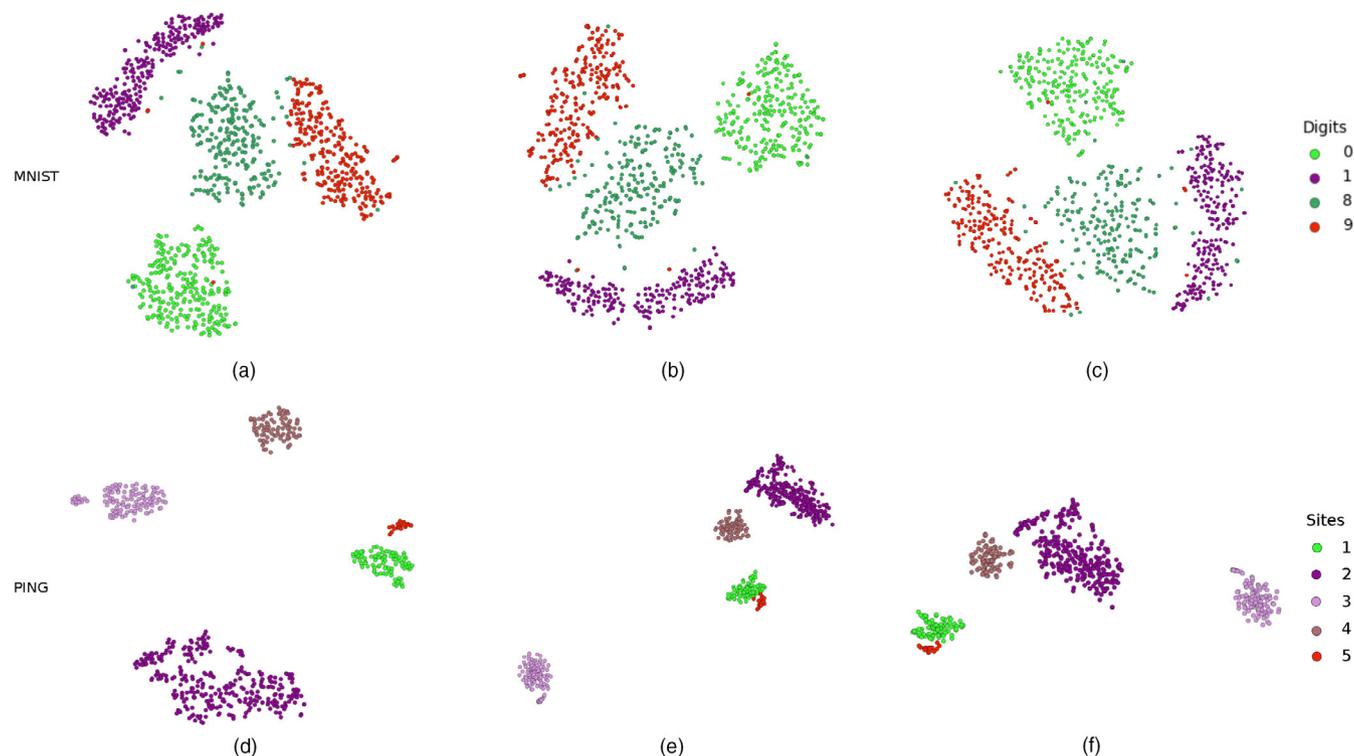


FIGURE 10 Experiment for DP-dSNE of MNIST and PING dataset with $\sigma^2 = 0.001$. (a–c) The t-SNE, dSNE, and DP-dSNE layout for the MNIST dataset; (d–f) the t-SNE, dSNE, and DP-dSNE output for the PING dataset, respectively. We observe that DP-dSNE gives overall close results to dSNE and centralized t-SNE. In the MNIST layout, each class is marked by a unique color and in PING layout, each site is marked by a unique color

Figure 11 shows a plot of ϵ for a fixed δ for the PING dataset as we increase the number of iterations for convergence. We fix δ to be a value $\delta \ll \frac{1}{N}$, where N denotes the number of samples used in the computation. Again, since each site has disjoint data samples from the other sites, we only observe b_{\max} of the first site to compute ϵ . We also compare the (ϵ, δ) pairs of the RDP analysis compared to the strong composition (Cynthia, Rothblum Guy, & Salil, 2010). Both the moments accountant and the RDP method are shown to give tighter, stronger privacy bounds (Ilya, 2017; Martin et al., 2016), showing the benefits of defining the guarantees in terms of RDP. After 1,000 iterations, for the PING dataset, given $\delta = 10^{-5}$ and $\sigma^2 = 0.001$, we obtain $(1.39, 10^{-5})$ -DP using the moments accountant and RDP method and $(5.85, 10^{-5})$ -DP using strong composition. These (ϵ, δ) pairs are similar for the MNIST dataset.

6 | DISCUSSION

The current practice of data sharing and pooling face great challenges as privacy concerns such as subject de-identification becomes more apparent. Previous studies have showed that in some cases it is possible to identify specific subjects from a dataset consisting of patients with rare diseases (Latanya, 2013; Latanya, Merce, & Michael, 2015). The inability to combine datasets from different research groups and data sites can be devastating, as individual sites rarely contain enough

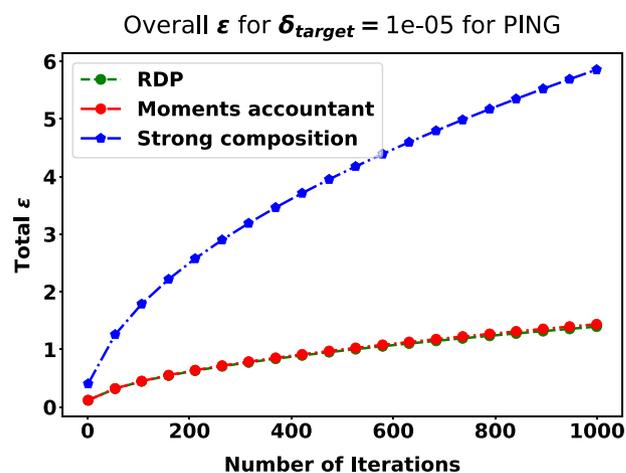


FIGURE 11 Plot of the number of iterations (J) versus the total ϵ given $\delta = 10^{-5}$ and $\sigma^2 = 0.001$. The RDP and moments accountant gives smaller values of ϵ over the strong composition method

data to answer the questions of foremost importance in biomedical research. There have been many notable methods that address the problem of data scarcity at individual sites in decentralized settings. For example, Virtual Pooling and Analysis of Research Data (ViPAR) (Carter Kim et al., 2015) is a framework proposed in which a secure and trusted coordinator node (or server) synchronizes with the remote

sites involved in the computation. At each iteration, each remote site sends data via an encrypted channel to the secure server. Data are then stored in RAM on the VIPAR coordinator server, where the data samples are analyzed and subsequently removed without ever being permanently stored. However, this process still relies on sending data outside of the original site. In addition, sending data via an encrypted channel incurs severe bandwidth and traffic overhead that ultimately increases computational load. Another promising approach is the Enhancing Neuroimaging Genetics through Meta-Analysis (ENIGMA) consortium (Thompson Paul et al., 2014), which is a community approach that requires the local sites to upload or email the summary statistics of the data following implementation of shared analysis scripts. ENIGMA uses both mega (if data pooling is possible) and meta-analysis. In meta-analysis, each local site runs the same analysis (e.g., regression) using the same measurements of the brain to aggregate summary statistics from all of the sites. The ENIGMA model has been widely embraced by the community. To run a meta-analysis, the leading site has to coordinate with all of the local sites before starting and after the completion of computation. Meanwhile, the ENIGMA meta-analysis approach does not support multivariate or multi-shot computation, that is, computing results in an iterative manner (Anand, Sergey, Jessica, Mohammad, & Vince, 2014). This is key as close to 50% of the data in some work groups, based on internal polling, cannot be centralized and thus much rely on meta-analysis. In addition, a standard meta-analysis approach does not provide any formal guarantees that it will prevent the re-identification of individuals. In many machine learning problems, there are many cases in which statistics exchange must be done in a multi-shot manner, as single-shot is not enough to obtain an optimal solution (Saha et al., 2017).

Some recent research combining federated learning, differential privacy and encrypted computing is described in a recent whitepaper (Emma, 2019). The Intel corporation has started a collaboration with the University of Pennsylvania and 19 other institutions to advance real world medical research using federated learning. Their work showed that a deep learning model trained by the traditional federated learning approach can reach up to 99% training accuracy (Sheller, Anthony, Brandon, Jason, & Spyridon, 2018).

Several notable tools and algorithms were introduced to handle federated computing efficiently. PySyft (OpenMined, 2019) is one of OpenMined's Python code libraries that integrated cryptography and distributed technology with PyTorch and Tensorflow. This was mainly developed to train AI models in a secured way by ensuring patient privacy using distributed data. Our platform COINSTAC (Plis et al., 2016) is another example of an open source platform addressing these tasks. Researchers at Google Inc. introduced a model of federated learning using distributed data of user's mobile devices (Brendan, Eider, Daniel, & Agüera, 2016). In this model, a mobile device downloads and trains the model by accessing the data of the user's device. It summarizes the changes and sends them as an update to the cloud using encrypted communication. Finally, the updates coming from all of the devices are averaged in the cloud and improves the shared model.

Federated Averaging (FedAvg) is a computation technique introduced in 2016 to fit a global model in the decentralized setting

(Brendan et al., 2016). In this model, the parameters are initialized on the server and distributed to the local clients. After training the model on each local dataset over multiple iterations, the trained parameters are delivered to the server, which computes the average to send the weights back to the local sites. We adopted a very similar approach to Federated Averaging in dSNE (Saha et al., 2017) before proposing the proxy data sharing technique (Yue et al., 2018). In dSNE, each local site accesses a publicly available dataset and updates its model using the combination of the shared data and its own local data. Similar to the communication round in Zhao et al. (Yue et al., 2018), each site runs the operations over a fixed number of iterations to reach the optimal solution. We also applied the averaging technique in which the local model is averaged after each iteration and transferred to by the coordinator node.

None of these existing methods fully address how we can perform data assessment and quality control in a decentralized manner. Data centers and institutions may not be willing or able to share their data due to a need to preserve the privacy of their subjects, precluding analyses that pool the data to a single site. To address these issues, we introduced a way to visualize federated datasets in a single display: dSNE (Saha et al., 2017) and its differentially private counterpart, DP-dSNE. In both algorithms, one coordinator node communicates with all local sites during one computation period. Our multi-shot approach follows from the averaging strategy similar to that of (Anand et al., 2014). The performances of t-SNE and dSNE are presented in Figures 2–9. In the best case scenarios, dSNE almost replicates t-SNE and shows great performance in terms of the comparison metrics. We showed that the performance increases when the reference data contains a large amount of samples as shown in Figure 2 for the MNIST dataset. We observe the similar type of behavior for the COIL-20 dataset, shown in Figures 3 and 4. Our results in the influence of large reference samples reflect the results also shown in djICA (Baker, Silva, Calhoun, Sarwate, & Plis, 2015).

In Figures 5–9, we were also able to observe significant results for six different biomedical datasets (ABIDE, sMRI, PING, fBIRN, BSNIP, and MRN). Data assessment and quality control plays a vital role during data acquisition from multiple data sources, especially to keep consistency or adjust parameters across various studies. We designed 4.1.7 to check the effectiveness of our algorithm in detecting outliers in multisite consortia. In Experiment 4.1.7, we collected 705 samples from Boulder site, where 120 of the samples were acquired from the second 3T dataset that was from a specific study with a very different acquisition protocol. Most importantly, these 120 samples were examples of a poor quality scan. In Figure 9, we see that these specific samples are clustered together (marked by red color) as outliers in both t-SNE and dSNE plots. Our results are similar to one of our earlier works, which used t-SNE to detect outliers of the same MRN dataset (Panta Sandeep et al., 2016).

In Figures 10 and 11, we can see the performance of DP-dSNE with a noise variance of $\sigma^2 = 0.001$ on the MNIST and PING datasets. From Figure 10, we see that DP-dSNE gives very similar results compared to t-SNE and dSNE. We implemented DP-dSNE as dSNE does not provide any formal privacy guarantees. Even though the data

samples in each local site never leaves the sites, sending the gradient values from each local site could potentially leak privacy. For datasets that are sensitive and have stricter protocols, we propose that DP-dSNE can provide utility similar to that of dSNE with strong privacy guarantees.

Lastly, we also implemented single-shot dSNE, but the results were not as promising as multi-shot dSNE (dSNE), as shown in 14. In single-shot dSNE, there is no way for the local sites to communicate iteratively. Additionally, a major problem of averaging arises when different sites are widely varied in terms of size and population (Plis et al., 2016). This problem is analogous to the size of the reference dataset in multi-shot dSNE. The reference dataset should provide variability and be large in size, in which otherwise, there is a high chance of not obtaining the optimal embedding. To obtain an optimal embedding, the algorithm should be run multiple times with a different initialization.

The Collaborative Informatics and Neuroimaging Suite Toolkit for Anonymous Computation (COINSTAC⁷) (Harshvardhan, Ross, Javier, et al., 2020; Plis et al., 2016), a dynamic and decentralized platform,

was introduced to address the difficulty of data sharing. This platform gives scope to perform distributed computation by using the commonly used algorithm in privacy-preserving mode. Our dSNE algorithm is currently deployed within the COINSTAC framework.⁸ Figure 12 shows the different computation phases of dSNE experiment in COINSTAC using MNIST dataset. In this experiment, three local and one remote site participate in the computation, where the remote site contains 200 samples of each digit (0 to 9) and each local site contains 20 samples of each digit. The results shown in Figure 12 demonstrate the robust capability of the COINSTAC framework. Future work consists of running simulations using all of the datasets used in this paper. This serves to demonstrate that COINSTAC ensures the real time applicability of our algorithm in the biomedical domain.

In our algorithm, the shared dataset can be provided by the master node, to be distributed across the participants. A number of multi-center neuroimaging datasets, currently are publicly available data and can be utilized to enable dSNE. There are many options providing data which can be of use for this including: OpenNEURO,⁹ Human

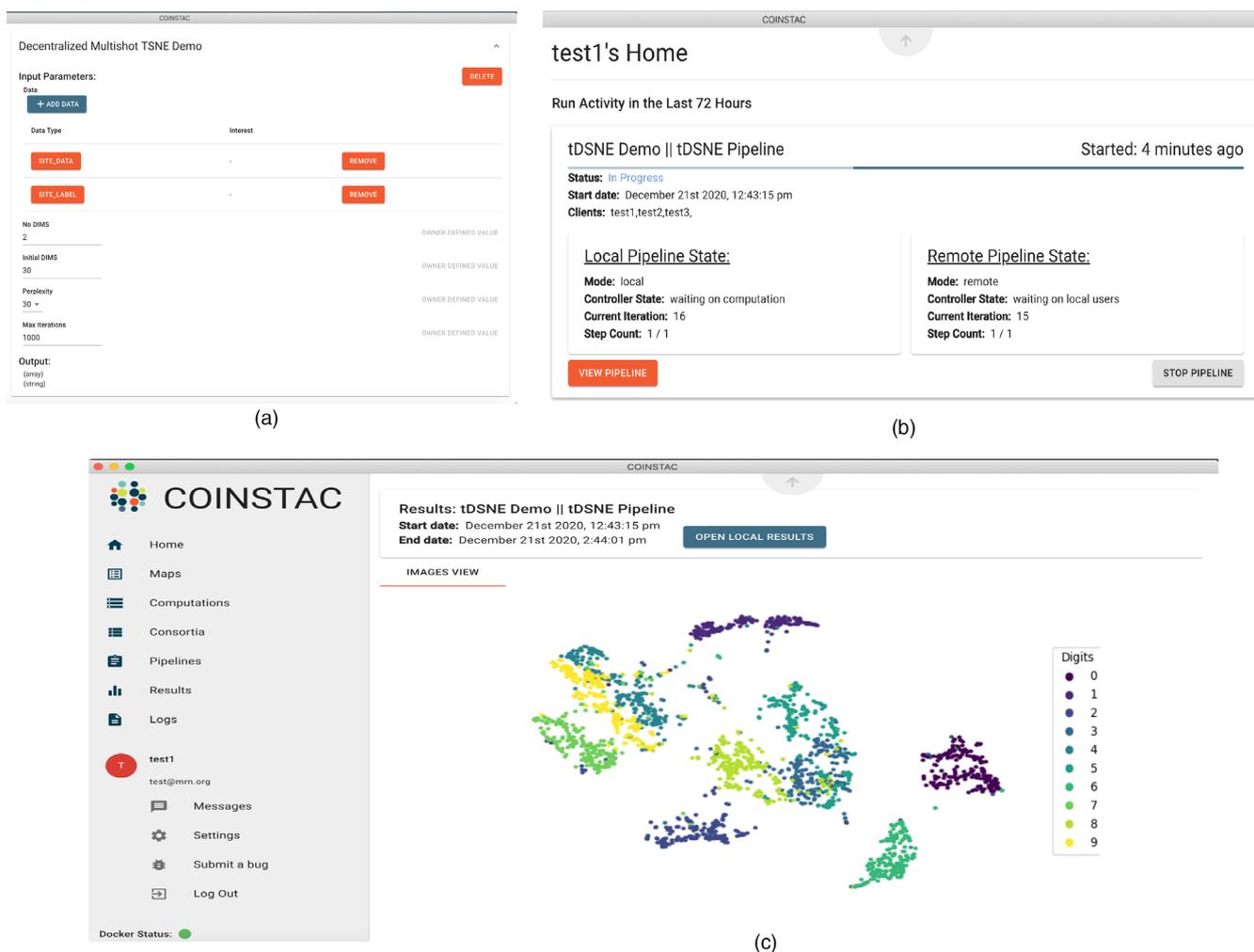


FIGURE 12 An run time demo of dSNE algorithm in coinstac simulator. (a–c) The computation phase of dSNE at the beginning, middle, and at the end of the simulation

Connectome Project,¹⁰ COINS,¹¹ NITRC, various datasets available through DataLad¹² (Halchenko, Kyle, Benjamin, et al., 2021) or kaggle competitions, and many more. We provide an extended list in the Appendix A.

Throughout this paper, we empirically compared our algorithm with t-SNE on different datasets. In terms of computational complexity, dSNE also outperforms the t-SNE algorithm. In dSNE, recall that one needs to compute the pairwise distance between all co-located samples. This amounts to m reference samples and k local samples. In this case, each site creates $(m + k) \times (m + k)$ matrices to compute the pairwise affinities. If the data were pooled, this pairwise matrix would necessarily be $(m + sk) \times (m + sk)$, for s sites. Note that this is significantly larger in comparison with the parallel decentralized creation of $s(m + k) \times (m + k)$ matrices. As our algorithm is an iterative approach, and we do not consider any random subsets of samples in each iteration, our algorithm does not allow stochastic gradient descent (SGD). The computational and the memory complexity of dSNE are $\mathcal{O}(s(m+k)^2)$. In dSNE, to obtain the optimal results, it may need to run the same experiment several times. In dSNE, the simulation begins based on the randomly initialized low-dimensional Y values. If the initialization happens to be poor, it may face some local minimas during the gradient descent computation. In that case, the global minima may not be obtained and the simulation should be ran again to get the optimal solution.

7 | CONCLUSIONS

In this paper, we have proposed two algorithms: decentralized stochastic neighbor embedding (dSNE) and differentially private decentralized stochastic neighbor embedding (DP-dSNE). Our dSNE algorithm enables the embedding of high-dimensional private neuroimaging data spread over multiple sites into low-dimensional space for visualization. This visualization allows us to perform quality control of poor data samples and also helps learn a global interrelation structure among brain volumes or feature vectors. Throughout the dSNE computation, no data samples leave their respective local sites, and only minimal gradient information from the embedding space is transferred across the sites. The clusters in the output embeddings are formed by samples belonging to classes, possibly present across many locations. Of course, the algorithm is neither explicitly aware nor requires the prior existence of any classes. The main idea of this iterative method is to share only the parts related to the publicly available reference dataset. As our results show, this is enough to co-orient classes that are spread across multiple locations. Extensive validation of eight datasets (two toy and six multisite neuroimaging) demonstrate the utility of our approaches. Our results showed that although multi-shot dSNE is robust to various conditions and settings (e.g., changes in the number of sites, and rare or missing data), and highest performance is achieved when the reference dataset is dense. Even though the data samples never leave the local sites in dSNE, there is still room for a potential privacy leak, as we are sending the gradients over to a coordinator node. DP-dSNE tackles this by introducing formal privacy guarantees within the gradients. Our

results show that both dSNE and DP-dSNE provide good utility for decentralized visualization while preserving privacy of the local samples. The implementation and integration of the algorithm with an existing neuroimaging platform for federated neuroimaging COINSTAC provides our methods as ready to use tools.

For future work, we believe that an alternative solution to our decentralized setting can be to use an average of the gradients weighted by the quality of the respective local t-SNE embeddings. However, it is not immediately clear how one should approach this. Using clustering measures on location specific data to weigh each Y may bias the results toward good local groupings over poor ones. Our novel metrics are also not quite able to convey each site's contribution to give it proper weight. In most general settings, we do not know a priori what type of data samples each site contains, as each local site has private data. Given these difficulties, we leave the problem for future work, noting that it could be an exciting research direction. Finally, we conclude with that we believe dSNE is a valuable quality control tool for virtual consortia when working with private data in decentralized settings.

ACKNOWLEDGMENTS

Autism data was provided by ABIDE. We acknowledge primary support for the work by Adriana Di Martino provided by the NIMH (K23MH087770) and the Leon Levy Foundation and primary support for the work by Michael P. Milham and the INDI team was provided by gifts from Joseph P. Healy and the Stavros Niarchos Foundation to the Child Mind Institute, as well as by an NIMH award to MPM (NIMH R03MH096321). Data for Schizophrenia classification used in this study were downloaded from the Function BIRN Data Repository (<http://bdr.birncommunity.org:8080/BDR/>), supported by grants to the Function BIRN (U24-RR021992) Testbed funded by the National Center for Research Resources at the National Institutes of Health, USA and from the COllaborative Informatics and Neuroimaging Suite Data Exchange tool (COINS; <http://coins.trendscenter.org>) and data collection was performed at the Mind Research Network, and funded by a Center of Biomedical Research Excellence (COBRE) grant 5P20RR021938/P20GM103472 from the NIH to Dr. Vince Calhoun.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

ENDNOTES

¹ <https://www.kaggle.com/c/digit-recognizer>

² <http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>

³ http://fcon_1000.projects.nitrc.org/indi/abide/

⁴ <http://pingstudy.ucsd.edu/Data.php>

⁵ <https://coins.trendscenter.org/>

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

⁷ <https://coinstac.org/>

⁸ The code is available at: <https://github.com/trendscenter/coinstac-dsne-multishot>

- ⁹ <https://openneuro.org>
- ¹⁰ <http://www.humanconnectomeproject.org/>
- ¹¹ <http://coins.trendscenter.org>
- ¹² <http://www.data-lad.org/>
- ¹³ <https://www.kaggle.com/c/mlsp-2014-mri/data>

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions. The experimental code will be available upon request from the corresponding author.

ORCID

Debbbrata K. Saha  <https://orcid.org/0000-0003-0754-7570>

Sergey M. Plis  <https://orcid.org/0000-0003-0040-0365>

REFERENCES

- Amadou, G., Yannick, M., Julia, I., Philippe, L., Andrew, T., & Elinor, M.J. (2014). DataSHIELD: Taking the analysis to the data, not the data to the analysis. *International Journal of Epidemiology*, 43(6), 1929–1944.
- Amir, G., Gal, C., Fernando, P., & Naftali, T. (2007). Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8, 2265–2295.
- Anand, S., Sergey, P., Jessica, T., Mohammad, A., & Vince, C. (2014). Sharing privacy-sensitive access to neuroimaging and genetics data: A review and preliminary validation. *Frontiers in Neuroinformatics*, 8, 35.
- Baker, B. T., Silva, R., Calhoun, V., Sarwate, A., & Plis, S. (2015). Large scale collaboration with autonomy: Decentralized data ICA. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1–6). Boston, MA: IEEE.
- McMahan H. Brendan, Moore Eider, Ramage Daniel, Arcas Blaise Agüera. (2016) Federated learning of deep networks using model averaging. CoRR.2016;abs/1602.05629.
- Carter Kim, W., Francis Richard, W., Autism Registry Epidemiology, et al. (2015). ViPAR: A software platform for the virtual pooling and analysis of research data. *International Journal of Epidemiology*, 45(2), 408–416.
- Cynthia, D., & Aaron, R. (2014). The algorithmic foundations of differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4), 211–407.
- Cynthia, D., Frank, M. S., Kobbi, N., & Adam, S. (2006). Calibrating noise to sensitivity in private data analysis. In H. Shai & R. Tal (Eds.), *Theory of cryptography* (pp. 265–284). Berlin-Heidelberg: Springer.
- Cynthia, D., Rothblum Guy, N., & Salil, V. (2010). *Boosting and differential privacy* (pp. 51–60). Washington, DC: IEEE.
- Dan, H., Huerta Michael, F., McAuliffe Matthew, J., & Farber, G. K. (2012). Sharing heterogeneous data: The national database for autism research. *Neuroinformatics*, 10(4), 331–339.
- Davies, D. I., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2), 224–227.
- Di Martino, A., Yan, C.-G., Li, Q., Denio, E., Castellanos, F. X., Alaerts, K., ... Milham, M. P. (2014). The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular Psychiatry*, 19(6), 659–667. <https://doi.org/10.1038/mp.2013.78>
- Dunn, J. C. (1974). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3), 32–57.
- El-Ad, A., Kara, D., Michelle, T., Simonds, E.F., Levine, J.H., & Bendall, S.C. (2013). VISNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature Biotechnology*, 31(6), 545–552.
- Emma, B. (2019). Privacy-preserving AI in medical imaging: Federated learning, differential privacy, and encrypted computation. Retrieved from <https://blog.openmined.org/federated-learning-differential-privacy-and-encrypted-computation-for-medical-imaging/>.
- Fu, Z., Caprihan, A., Chen, J., Du, Y., Adair, J. C., Sui, J., ... Calhoun, V. D. (2019). 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), 3203–3221. <https://doi.org/10.1002/hbm.24591>
- Geoffrey, H., & Sam, R. (2003). Stochastic neighbor embedding. *Advances in Neural Information Processing Systems*, 15, 833–840.
- Gunter, S., Vivek, B., Chunshui, Y., Tao, S., Jernigan, T., & Heinz, A. (2019). Precision medicine and global mental health. *Lancet Global Health*, 7(1), e32.
- Halchenko, Y. O., Kyle, M., Benjamin, P., Solanky, D.S., Wagner, A.S., & Gors, J. (2021). DataLad: Distributed system for joint management of code, data, and their relationship. *Journal of Open Source Software*, 6(63), 3262.
- Halkidi, M., Vazirgiannis, M., Batistakis, Y.. Quality scheme assessment in the clustering process. Zighed Djamel, A., Komorowski Jan, Żytkow Jan, *Principles of data mining and knowledge discovery*. 265–276 Springer 2000; Berlin-Heidelberg.
- Harshvardhan, G., Ross, K., Javier, R., Verner, E., Baker, B.T., & Silva, R.F. (2020). COINSTAC: Collaborative informatics and neuroimaging suite toolkit for anonymous computation. *Journal of Open Source Software*, 5(54), 2166.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24, 498–520.
- Ilya, M. (2017). Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*. Santa Barbara, CA: IEEE.
- Imtiaz, H., Mohammadi, J., Silva, R., Baker, B., Plis, S.M., Sarwate, A.D. & Calhoun, V. (2019). Improved differentially private decentralized source separation for fMRI data. ArXiv.2019;abs/1910.12913.
- Jernigan, T. L., Brown, T. T., Hagler, D. J., Akshoomoff, N., Bartsch, H., & Newman, E. (2016). The pediatric imaging, neurocognition, and genetics (PING) data repository. *NeuroImage*, 124, 1149–1154.
- John, A., & Friston Karl, J. (2005). Unified segmentation. *NeuroImage*, 26(3), 839–851.
- Kevin, M., David, D., Zheng, W., Gigante, S., Burkhardt, D.B., & Chen, W.S. (2019). Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37, 1482–1492.
- Latanya, S. (2013). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5), 557–570.
- Latanya, S., Merce, C., & Michael, B.-S. (2015). Sharing sensitive data with confidence: The datatags system. *Technology Science*. Available from <http://techscience.org/a/2015101601>
- Laurens, M., & Geoffrey, H. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.
- LeCun Yann, Cortes Corinna. (2010) MNIST handwritten digit database.
- Leland, M. I., John, H., Nathaniel, S., & Lukas, G. (2018). UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3, 861.
- Martin, A., Andy, C., Ian, G., Brendan McMahan, H., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In *CCS '16* (pp. 308–318). New York, NY: Association for Computing Machinery.
- Mary, I. (2015). *Federal Interagency Traumatic Brain Injury Research (FITBIR) bioinformatics platform for the advancement of collaborative traumatic brain injury research and analysis*. Chicago, IL: 143rd APHA Annual Meeting and Exposition.

- Meda, S. A., Giuliani, N. R., Calhoun, V. D., Kanchana, J., Schretlen, D. J., Anne, P., ... Pearlson, G. D. (2008). A large scale ($N = 400$) investigation of gray matter differences in schizophrenia using optimized voxel-based morphometry. *Schizophrenia Research*, *101*(1), 95–105.
- Mikhail, B., & Partha, N. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*(6), 1373–1396.
- Nene, S. A., Nayar, S. K., & Hiroshi, M. (1996). *Columbia Object Image Library (COIL-20)*. CUCS-005-96. New York, NY: Department of Computer Science, Columbia University.
- OpenMined. (2019). Privacy-preserving AI in medical imaging: Federated learning, differential privacy, and encrypted computation.
- Panta, S. R., Wang, R., Fries, J., Kalyanam, R., Speer, N., Banich, M., ... Calhoun, V. D. (2016). A tool for interactive data visualization: application to over 10,000 brain imaging and phantom MRI data sets. *Frontiers in Neuroinformatics*, *10*. <https://doi.org/10.3389/fninf.2016.00009>
- Pichapati, V., Suresh, A., Yu, F. X., Reddi, S., Kumar, S. (2019). AdaClip: Adaptive clipping for private SGD. CoRR.2019;abs/1908.07643.
- Pierre, D., & Jeanny, H. (1997). Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, *8*(1), 148–154.
- Plis, S. M., Sarwate, A. D., Wood, D., Dieringer, C., Landis, D., Reed, C., ... Calhoun, V. D. (2016). COINSTAC: A privacy enabled model and prototype for leveraging and processing decentralized brain imaging data. *Frontiers in Neuroscience*, *10*. <https://doi.org/10.3389/fnins.2016.00365>
- Plis, S. M., Hjelm, D. R., Salakhutdinov, R., Allen, E. A., Bockholt, H. J., Long, J. D., ... Calhoun, V. D. (2014). Deep learning for neuroimaging: a validation study. *Frontiers in Neuroscience*, *8*. <https://doi.org/10.3389/fnins.2014.00229>
- Raftery, A. E. (1986). A note on Bayes factors for log-linear contingency table models with vague prior information. *Journal of the Royal Statistical Society*, *48*(2), 249–250.
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.
- Roweis Sam, T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *5500*, 2323–2326.
- Saha, D. K., Calhoun, V. D., Panta, S. R., & Plis, S. M. (2017). See without looking: Joint visualization of sensitive multi-site datasets. In *JCAI '17* (pp. 2672–2678). Melbourne, Australia: AAAI Press.
- Sammon, J. W., Jr. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, *100*(5), 401–409.
- Scott, H. (2014). *Functional magnetic resonance imaging*. Sunderland, MA: Sinauer Associates.
- Sheller, Micah J., Reina G. Anthony, Edwards Brandon, Martin Jason, Bakas Spyridon. (2018) Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. CoRR.2018;abs/1810.04304.
- Shuang, W., Jiang Xiaoqian, W., Yuan, C. L., Samuel, C., & Lucila, O.-M. (2013). EXpectation propagation LOGistic REgression (EXPLORER): Distributed privacy-preserving online model learning. *Journal of Biomedical Informatics*, *46*(3), 480–496.
- Tenenbaum Joshua, B., Vin, D. S., & Langford John, C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*(5500), 2319–2323.
- Thompson, P. M., Stein, J. L., Medland, S. E., Hibar, D. P., Vasquez, A. A., Renteria, M. E., ... Drevets, W. (2014). The ENIGMA Consortium: large-scale collaborative analyses of neuroimaging and genetic data. *Brain Imaging and Behavior*, *8*(2), 153–182. <https://doi.org/10.1007/s11682-013-9269-5>
- Tony, L. F., Ming, T. K., & Zhi-Hua, Z. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *6*(1), 3.
- Vin, D. S., & Tenenbaum Joshua, B. (2003). Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems*, *15*, 721–728.
- Vin, S., & Joshua, T. (2004). Sparse multidimensional scaling using landmark points. *Technology*, *120*,
- van Unen, V., Höllt, T., Pezzotti, N., Li, N., Reinders, M. J. T., Eisemann, E., ... Lelieveldt, B. P. F. (2017). Visual analysis of mass cytometry data by hierarchical stochastic neighbour embedding reveals rare cell types. *Nature Communications*, *8*(1), 1–10. <https://doi.org/10.1038/s41467-017-01689-9>
- Weinberger, K. Q., & Saul, L. (2006). *An introduction to nonlinear dimensionality reduction by maximum variance unfolding*. (pp. 1683–1686). Menlo Park, CA: American Association for Artificial Intelligence.
- Xavier, C. F., Adriana, D. M., Cameron, C. R., Mehta Ashesh, D., & Milham, M. P. (2013). Clinical applications of the functional connectome. *NeuroImage*, *80*, 527–540.
- Yuhui, D., Zening, F., Sui, J., Gao, S., Xing, Y. & Lin, D. (2019). NeuroMark: An adaptive independent component analysis framework for estimating reproducible and comparable fMRI biomarkers among brain disorders. *medRxiv*.
- Zhao Yue, Li Meng, Lai Liangzhen, Suda Naveen, Civin Damon, Chandra, V. (2018) Federated learning with non-IID data. ArXiv.2018; abs/1806.00582.

How to cite this article: Saha, D. K., Calhoun, V. D., Du, Y., Fu, Z., Kwon, S. M., Sarwate, A. D., Panta, S. R., & Plis, S. M. (2022). Privacy-preserving quality control of neuroimaging datasets in federated environments. *Human Brain Mapping*, *43*(7), 2289–2310. <https://doi.org/10.1002/hbm.25788>

APPENDIX A: PUBLIC SHARED DATA

OpenNEURO is one of the data sources to access the public data and utilize it for our experiments. For the sMRI dataset, we have filtered based on the MRI scans of human subjects (no of subjects ≥ 100) and used the same type of preprocessing measurements that we have used for our dataset. Here are some of the versions which can be used for our experiments:

- <https://openneuro.org/datasets/ds000243/versions/00001>
- <https://openneuro.org/datasets/ds003643/versions/1.0.4>
- <https://openneuro.org/datasets/ds003481/versions/1.0.3>
- <https://openneuro.org/datasets/ds003949/versions/1.0.0>
- <https://openneuro.org/datasets/ds000158/versions/1.0.0>
- <https://openneuro.org/datasets/ds003097/versions/1.2.1>
- <https://openneuro.org/datasets/ds002790/versions/2.0.0>

For the fMRI dataset, we have filtered based on the resting state data of human subjects (no of subjects ≥ 100) and used the same type of preprocessing measurements that we have used for our dataset. Here are some of the versions which can be used for our experiments:

- <https://openneuro.org/datasets/ds000243/versions/00001>
- <https://openneuro.org/datasets/ds003346/versions/1.1.2>
- <https://openneuro.org/datasets/ds003481/versions/1.0.3>
- <https://openneuro.org/datasets/ds002843/versions/1.0.1>

Some public MRI datasets for the healthy controls and Schizophrenia patients are available in the Kaggle site.¹³ In the repository, two modalities of MRI (functional and structural) are available. From this resource, both functional networks connectivity values, and source based morphometry (SBM) features can be utilized.

The resting state fMRI data of healthy controls can be accessed from the Human Connectome Project repository.

APPENDIX B: MRN fMRI DATA

In this experiment, we check the scenario when the shared dataset only contains poor-quality scans. We have run an experiment on the MRN dataset to test this special case. In our experiment, there are a total three local and a remote site participating in the computation. We place 125 poor-quality fMRI samples to the remote site and each local site contains 200 good fMRI scans. Additionally, we ran a t-SNE experiment on the accumulated data for comparison. We represent our t-SNE and dSNE plot in Figure 13. We observe that our algorithm shows robust behavior even when all the shared samples are bad scans. This suggests the main benefit of the shared data is to serve as a common reference for the sites, rather than as a source of high quality data (Figure 14).

APPENDIX C: SINGLE-SHOT dSNE

For single shot dSNE (Algorithm 8), we first pass the reference data from centralized site C to each local site.

Now each local site has data consisting of two portions: (1) its local dataset, for which we need to preserve privacy, and (2) the

Algorithm 8 SingleshotDSNE

Input:

Objective parameters: ρ (perplexity)

Optimization parameters: T, η, α

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

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

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

$\mathbf{Y}_s \leftarrow \text{tSNE}(\mathbf{X}_s, \rho, T, \eta, \alpha) \triangleright$ At the coordinator node

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

\mathbf{Y}_s^{-p}

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

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

end for

$\mathbf{Y} \leftarrow [] \triangleright$ At the coordinator

for $p = 0$ to P **do** \triangleright At the coordinator

\mathbf{Y}_p^{-p}

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

end for

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

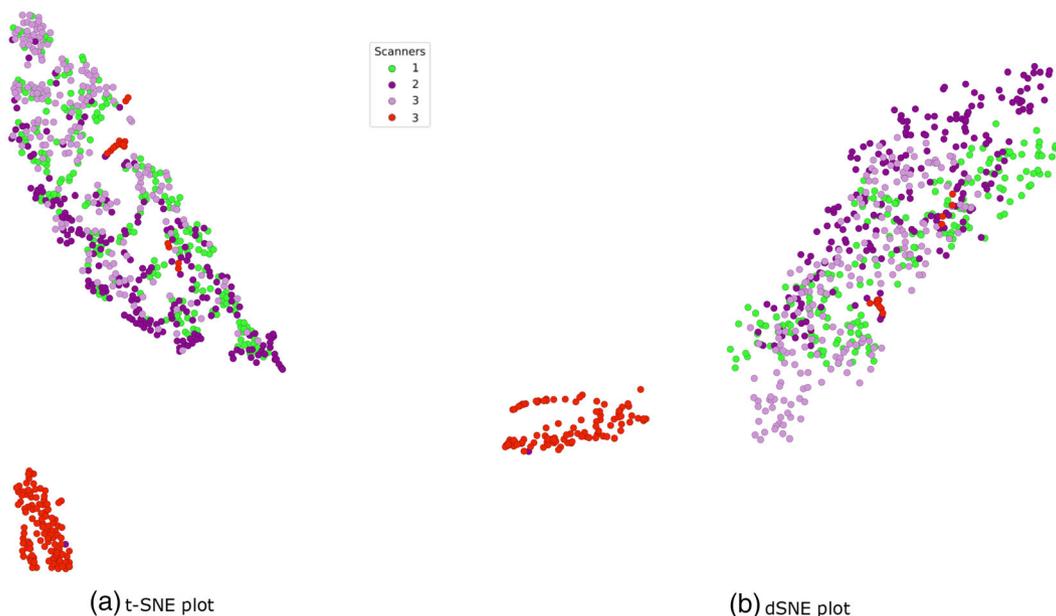


FIGURE 13 Experiment for outlier detection of the MRN fMRI dataset. In this experiment, the shared sample only contains the bad scans. In both t-SNE and dSNE, we can successfully identify poor quality scans which is marked by the red color. In this experiment, three local and one remote sites participated in the computation. In the layout, each point represents a brain scan of an individual

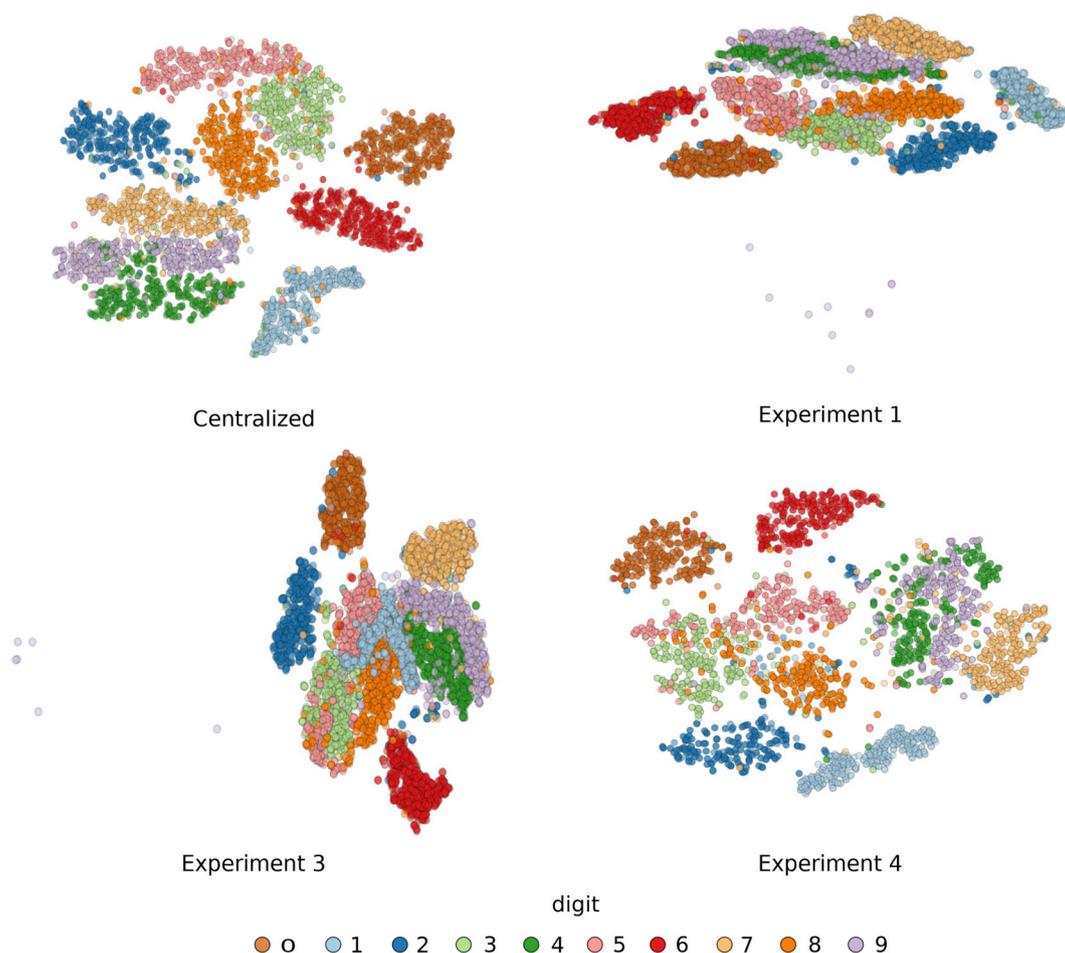


FIGURE 14 Single-shot dSNE layout of MNIST data (Saha et al., 2017). Single-shot was run for the experiment of 1, 3 and 4 of MNIST dataset. For all experiments, we are able to embed and group same digits from different sites without passing any site info to others. Here every digit is marked by a unique color. Centralized is the original tSNE solution for locally grouped data. Digits are correctly grouped into clusters but these clusters tend to heavily overlapped

shared reference dataset. Each local site runs the t-SNE algorithm on this combined data (local and reference) 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 \mathbf{y} for the local dataset. The embedding for the shared data has been precomputed at the coordinator node and shared with each local site. Similar to the landmark points approach (Vin & Joshua, 2004), 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 and the embedding of the shared dataset. Since the embedding points of the shared dataset do not change, all local embeddings are easily combined by aligning the points representing the shared data.