Bioinformatics Advances, 2023, **00**, vbad097
https://doi.org/10.1093/bioadv/vbad097
Advance Access Publication Date: 3 August 2023
Original Article

OXFORD

Data visualization

Toward computing attributions for dimensionality reduction techniques

Matthew Scicluna[1,2], Jean-Christophe Grenier[1], Raphaël Poujol[1], Sébastien Lemieux [2,3], Julie G. Hussin [1,2,3,4,*]

[1]Montreal Heart Institute, Research Center, Montreal, Quebec H1T 1C8, Canada
[2]Département de Biochimie et Medecine Moleculaire, Université de Montréal, Montreal, Quebec H3C 3J7, Canada
[3]Mila—Quebec AI institute, Montreal, Quebec H2S 3H1, Canada
[4]Département de Medecine, Université de Montréal, Montreal, Quebec H3C 3A7, Canada

*Corresponding author. Montreal Heart Institute, Research Center, 5000 Rue Bélanger, Montreal, Quebec, H1T 1C8, Canada. E-mail: julie.hussin@umontreal.ca

Associate Editor: Shaun Mahony

Abstract

Summary: We describe the problem of computing local feature attributions for dimensionality reduction methods. We use one such method that is well established within the context of supervised classification—using the gradients of target outputs with respect to the inputs—on the popular dimensionality reduction technique t-SNE, widely used in analyses of biological data. We provide an efficient implementation for the gradient computation for this dimensionality reduction technique. We show that our explanations identify significant features using novel validation methodology; using synthetic datasets and the popular MNIST benchmark dataset. We then demonstrate the practical utility of our algorithm by showing that it can produce explanations that agree with domain knowledge on a SARS-CoV-2 sequence dataset. Throughout, we provide a road map so that similar explanation methods could be applied to other dimensionality reduction techniques to rigorously analyze biological datasets.

Availability and implementation: We have created a Python package that can be installed using the following command: `pip install interpretable_tsne`. All code used can be found at github.com/MattScicluna/interpretable_tsne.

1 Introduction

Dimensionality reduction techniques, such as t-distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton 2008), Uniform Manifold Approximation and Projection (McInnes *et al.* 2020), and Potential of Heat-diffusion for Affinity-based Trajectory Embedding (Moon *et al.* 2019), have become widespread tools in the data analyst's toolbox, achieving popularity in the Machine Learning (ML) community and particularly in Bioinformatics. Such techniques can identify structure in high-dimensional data by projecting it onto a lower dimensional manifold. When the manifold is 2 or 3 dimensions, the structure can be easily interrogated using ordinary scatterplots. While these methods have informed many data analysis projects, they suffer from an overlooked limitation: there is no obvious way to attribute a datapoints' embedding to its corresponding input features. Currently, practitioners rely on checking for enrichment of features within groups of points of interest. This is often *ad hoc*, and can potentially miss significant features due to cognitive tendencies, such as confirmation bias.

We propose a method that can produce such attributions for the t-SNE algorithm. Our methodology is conceptually simple, being based on the well-established practice of using model gradients to compute feature attributions (Simonyan *et al.* 2014). Our algorithm can be added to any implementation of t-SNE, with comparable complexity to the original t-SNE fitting procedure.

In the next section, we describe interpretability methods in more detail, contextualizing ours. We then introduce the constituent parts of our framework: the gradients attribution method and the t-SNE dimensionality reduction algorithm. Then, we propose our method to apply gradients computation to the t-SNE algorithm. Then, we describe the methods for validating our attributions, describing the results we get when applying our validation methods on the MNIST dataset. Finally, we utilize our attributions to analyze a SARS-CoV-2 dataset—a case study that represents a realistic bioinformatic application. We also performed an additional attribution experiment on the 20 newsgroups dataset that can be found in Supplementary Appendix F. In summary, this work makes the following contributions:

1) Derives the equations to compute the gradient of t-SNE embeddings with respect to each input.
2) Produces an algorithm which returns these gradients and is compatible with the Barnes–Hut t-SNE approximation.
3) Introduces a novel metric to evaluate dimensionality reduction attribution performance.
4) Demonstrates empirical evidence for the methodology on MNIST and SARS-CoV-2 datasets.

Received: 20 April 2023; **Revised:** 21 June 2023; **Editorial Decision:** 7 July 2023; **Accepted:** 1 August 2023

2 Background

Previous literature suggests that interpretability is not a monolithic concept, but in fact reflects several distinct ideas (Lipton 2018). For the purposes of contextualizing this work, we define interpretability as the ability to extract human-understandable insights from ML models (we include dimensionality reduction techniques, such as t-SNE, into our definition of ML models). One way to ensure interpretability is to use a model, which admits a simple explanation by design. Within the supervised learning framework, algorithms have been designed to produce models, which are simple enough to be interpretable. These range from classic algorithms like Decision Trees and Sparse Lasso regression (Tibshirani 1996) to interpretable versions of modern deep learning architectures like BagNets (Brendel and Bethge 2019). The limitation of these approaches is that the increased interpretability comes at the expense of model performance.

Practitioners can instead apply *post-hoc* interpretability methods: which we define as methods that produce explanations of model behavior after training. There exists many such methods, which can be separated by the kind of explanation they provide: some are "local", providing explanations specific to each datapoint [e.g. LIME (Ribeiro *et al.* 2016), Vanilla Gradients (Simonyan *et al.* 2014)], while others produce "global" explanations of a models activity (Tan *et al.* 2018, Plumb *et al.* 2020). These methods can be further grouped based on whether they produce feature attributions, which we define as a score for each input feature, which represents the features relative influence on the models behavior.

Many *post-hoc*, local, feature attribution methods have been proposed. We can divide these into perturbation and gradient-based approaches. Perturbation-based approaches like LIME (Ribeiro *et al.* 2016) and SHAP (Lundberg and Lee 2017) change parts of the input and observe the impact on the output of the model. The downside of such methods is that they are computationally infeasible when model inference is slow since they require many model evaluations. Gradient-based approaches use the gradient (or a modification), to compute feature attributions [e.g. Layerwise Relevance Propagation (Bach *et al.* 2015) and DeConvNet (Zeiler and Fergus 2014)]. These techniques tend to be much more computationally efficient, but can be insensitive to either data or model (Adebayo *et al.* 2018).

2.1 Gradient attributions

Within the context of supervised learning of neural networks on classification tasks, techniques have been developed for computing (local) feature attributions. Let $S_c(x) \in \mathbb{R}$ be the score function of class c given by our classification model, when $x \in \mathbb{R}^d$ is the input data. Feature attribution methods assign a value to each feature $A_c(x) = \{A_{c,i}(x)\}_{i=1}^d$. $A_{c,i}(x)$ represents how much feature i of x contributed to the model's prediction of class c.

For this work, we use an attribution method commonly referred to as the vanilla gradient (Simonyan *et al.* 2014). For our purposes:

$$A_{c,i}(x) = \left[\frac{\partial S_c(x)}{\partial x}\right]_i. \tag{1}$$

The argument for using the gradients as attribution values provided in Simonyan *et al.* (2014) is that the above gradients correspond to the weights of the first order Taylor approximation of S_c at x. These weights would have a direct correspondence to attributions since the approximation is linear (intuitively, the larger the attribution, the less you have to change the corresponding input to achieve a fixed change in output).

In practice, many gradient-based attribution methods have been proposed and validated including Integrated Gradients (Sundararajan *et al.* 2017), DeconvNets (Zeiler and Fergus 2014), and "Guided Backpropagation" (Springenberg *et al.* 2015). While such techniques have well known limitations (Adebayo *et al.* 2018, Hooker *et al.* 2019), they nonetheless continue to be used all throughout the interpretable ML literature.

2.2 The t-SNE algorithm

The t-SNE algorithm is among the oldest and most influential dimensionality reduction techniques still in widespread use. We provide a sketch of the t-SNE algorithm here. For a detailed discussion of the t-SNE paper, we refer the reader to the original paper (van der Maaten and Hinton 2008).

Suppose, we have input data $x_1, \ldots, x_n \in \mathbb{R}^d$. Denote $y_i^t \in \mathbb{R}^{d'}$ as the embedding for x_i to be produced by the t-SNE algorithm at step t in the embedding space with dimension d' (usually two or three). The t-SNE algorithm updates the y_i^t's to minimize the Kullback–Leibler (KL) divergence, a measure of the difference between the probability distributions $p_{ij} := p(x_i, x_j)$ and $q_{ij}^{t-1} := q(y_i^{t-1}, y_j^{t-1})$:

$$KL(p_{ij}, q_{ij}) = \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{2}$$

Note that this represents distances between pairs of points in input and embedded space, respectively. The intuition is that we want the embeddings in the low dimensional space to recapitulate the distances between points in the high dimensional space. Ignoring optimization hyperparameters, our embeddings are updated using the following equations:

$$y_i^t = y_i^{t-1} + dy_i^t, \tag{3}$$

where:

$$\begin{aligned} dy_i^t &= 4 \textstyle\sum_{j \neq i} (p_{ij} - q_{ij}^{t-1}) \phi_{ij}^{t-1}, \\ \phi_{ij}^{t-1} &= (y_i^{t-1} - y_j^{t-1})(1 + ||y_i^{t-1} - y_j^{t-1}||^2)^{-1}. \end{aligned} \tag{4}$$

In t-SNE, we update the embedding of each datapoint using (4) until convergence.

3 Algorithm

The reasoning behind the use of the gradient as a feature attribution method can be used if we consider our score function $S_c(x)$ to be the output of a dimensionality reduction technique (for embedding dimension c) rather than the score of class c of a parametric classifier.

Furthermore, the t-SNE update formula (4) is the gradient of an objective function [Equation (2)] with respect to embeddings $y_1, \ldots, y_n$, and so each y_i is essentially receiving a Stochastic Gradient Descent (SGD) update. We propose inspecting the gradients of t-SNE in the same manner as one

would look at gradients with respect to their inputs in relation to supervised classifiers trained also via SGD.

3.1 Computing t-SNE attributions

In the supervised classification context, computing the gradient with respect to the input is usually very simple, but doing so for t-SNE is more involved, since the relationship between inputs $x_1, \ldots, x_n$ and outputs $y_1, \ldots, y_n$ is less clear. In the following section, we will derive the gradient of each component of a t-SNE embedded point with respect to its input:

$$\frac{\partial y_i^t}{\partial x_i}. \tag{5}$$

We do not use this gradient directly since we would end up with a set of feature attributions per t-SNE component. This is undesirable since (i) we want only one set of attributions and (ii) the t-SNE components themselves do not have any clear meaning. Instead, we return $A_c(x) = \frac{\partial ||y||^2}{\partial x}$. We found that this modification produced attributions that had an easy interpretation: they inform us of how the features of x_i contributed to the overall placement of y_i.

3.2 Computing the gradient of the t-SNE algorithm

Hereafter, we discuss applying the gradient attribution method to the t-SNE algorithm. We chose this algorithm since it is fairly easy to implement and analyze, and has become widely used within both of the ML and bioinformatics communities. We emphasize that our technique could be extended to other dimensionality reduction techniques, provided that they consist of no non-differentiable operations.

We can compute (5) since each of the steps of the t-SNE algorithm are differentiable (we assume that the Euclidean distance is used in the computation of p_{ij}). If we assume that $\frac{\partial y_i^t}{\partial x_j} = 0 \, \forall i \neq j$, we can compute (5) efficiently using dynamic programming:

$$\frac{\partial y_i^t}{\partial x_i} = \frac{\partial y_i^{t-1}}{\partial x_i} + \frac{\partial dy_i^t}{\partial x_i}, \tag{6}$$

where:

$$\frac{\partial dy_i^t}{\partial x_i} = 4 \sum_{j \neq i} \left\{ \left(\frac{\partial p_{ij}}{\partial x_i} - \frac{\partial q_{ij}^{t-1}}{\partial x_i} \right) \phi_{ij}^{t-1} + (p_{ij} - q_{ij}) \frac{\partial \phi_{ij}^{t-1}}{\partial x_i} \right\}. \tag{7}$$

At step t, we store $\frac{\partial y_i^t}{\partial x_i}$ so it can be accessed at step $t+1$. This allows us to compute the following:

$$\frac{\partial q_{ij}^{t-1}}{\partial x_i} = \frac{\partial q_{ij}^{t-1}}{\partial y_i^{t-1}} \frac{\partial y_i^{t-1}}{\partial x_i}, \tag{8}$$

$$\frac{\partial \phi_{ij}^{t-1}}{\partial x_i} = \frac{\partial \phi_{ij}^{t-1}}{\partial y_i^{t-1}} \frac{\partial y_i^{t-1}}{\partial x_i}. \tag{9}$$

We note that this can be implemented within any implementation of the standard t-SNE algorithm by the addition of a few lines of code. We provide pseudo-code in Algorithm 1. See Supplementary Appendix A for the formulas for $\frac{\partial q_{ij}^{t-1}}{\partial y_i^{t-1}}$, $\frac{\partial \phi_{ij}^{t-1}}{\partial y_i^{t-1}}$, $\frac{\partial p_{ij}}{\partial x_i}$, and for the full derivations.

Algorithm 1: Gradients for t-SNE

Require: data $x_1, x_2, \ldots, x_n$

Ensure: $\frac{\partial y_i^T}{\partial x_i}$

Initialize $y_1, \ldots, y_n$ as specified by t-SNE
Set $\frac{\partial y_i^0}{\partial x_i} = 0$
Compute $\{p_{ij}\}$, $\left\{\frac{\partial p_{ij}}{\partial x_i}\right\}$
for $t = 1$ **to** T **do**
    Compute $\{q_{ij}^{t-1}\}$, $\{\phi_{ij}^{t-1}\}$ using y_i^{t-1}
    Update y_i^t using Equation (3)
    Compute $\left\{\frac{\partial q_{ij}^{t-1}}{\partial y_i^{t-1}}\right\}$, $\left\{\frac{\partial \phi_{ij}^{t-1}}{\partial y_i^{t-1}}\right\}$
    Compute $\left\{\frac{\partial q_{ij}^{t-1}}{\partial x_i}\right\}$, $\left\{\frac{\partial \phi_{ij}^{t-1}}{\partial x_i}\right\}$ via Equation (8)
    Update $\left\{\frac{\partial y_i^t}{\partial x_i}\right\}$ using Equations (16) and (17)
end for

3.3 Barnes–Hut approximation

Most implementations of the t-SNE algorithm use the Barnes–Hut approximation to speed up computation time from $O(n^2)$ to $O(n \log n)$ (van der Maaten 2014). We show in Supplementary Appendix B how to derive gradients using the Barnes–Hut variant of t-SNE. We note that all experiments reported in this article were done using gradients of the Barnes–Hut approximation of t-SNE.

4 Methods

It is generally very difficult to assess the validity of feature attribution methods, even in their usual supervised classification context (Lipton 2018). In order to determine whether our attributions were identifying significant features, we performed a series of experiments on synthetic data as well as on the MNIST benchmark dataset. To show real-world applicability of our method, we used our method to identify the mutations driving SARS-CoV-2 evolution using publicly available sequence data. Please refer to Supplementary Appendix G for details regarding t-SNE hyperparameters, attribution processing, and performance on benchmarking experiments.

4.1 Simulated data experiments

We generated several datasets such that they would have a hierarchical cluster structure whose structure was attributed to a small subset of features. For each datapoint of a cluster, we translated a small subset of features by a fixed amount. Each cluster was designed such that a small subset of features was translated by a given amount. This set of features differed per cluster, and one cluster did not have any translated feature. We fixed the cluster structure and ground truth feature dependencies, but varied the amount of feature translation that defined the clusters. The details of the data generating procedure can be found in Supplementary Appendix C. After fitting our t-SNE and computing attributions for each synthetic dataset, we took the absolute value of the average of the attributions of all the points in each cluster, and found that, for each simulated dataset, these class-averaged attributions were significantly higher for the ground truth features versus the rest. This was observed even for the cluster that

contained no translated features. See Supplementary Appendix C for details of the results.

4.2 MNIST validation experiments

We performed a series of experiments using the MNIST dataset. The main idea was to corrupt features based on their attribution values, and then compute the t-SNE embeddings of this corrupted data. If the attributions had detected significant features, then the t-SNE of the corrupted data should be significantly different, then the t-SNE fit on the uncorrupted data. We used three separate metrics to quantify the extent of t-SNE structure degradation caused by the data corruption, adapted from metrics used to measure t-SNE quality (Lee and Verleysen 2009, Kobak and Berens 2019). These metrics are:

1) Spearman correlation. The correlation between distances of pairs of embedded points before and after feature corruption. This is a measure of the change of global structure.
2) Adjusted rand index (ARI). We computed the ARI between clusters generated using K-means clustering ($K = 10$) before and after corruption. This is a measure of the change of cluster structure.
3) 10 Nearest-neighbor preservation. The average of the 10 nearest neighbors retained by each point before and after corruption. This is a measure of the change of local structure.

We performed our validation experiments on a random subset of 10 000 MNIST digits. For each experiment, we computed 10 different t-SNEs (random seeds). We varied the percentage of features corrupted from 2% to 18% (in increments of 2%) and report the average over the percentage corrupted.

4.2.1 Local, class-level, and global attribution validation

On MNIST, we noticed that individual attributions highlighted idiosyncrasies of each digit (see Fig. 1A). We noticed that these attributions could be aggregated on a class level, and these saliency maps appeared to be visually meaningful (see Fig. 1B). This led us to investigate the validity of these attributions on three distinct levels:

1) "Local": Attributions produced for each individual digit
2) "Class": Attributions for each digit class
3) "Global": Attributions of each feature across all digits.

4.2.2 Selecting features to corrupt using the attributions

On the local level, we corrupted k% of features by corrupting the features within the top k percentile of attribution values (in absolute value). On the global level, we corrupted the features that appeared in the k percentile of attribution values most often. On the class level, we did the same but for all the points in each digit class separately. Note that on the local level, each digit had a different set of features to be corrupted. On the global level, the same features were corrupted for all digits. On the class-based level, each class had its own set of features to be corrupted, and every digit within a class had the same features corrupted.

Taking inspiration from previous work in the local feature attribution literature (Springenberg *et al.* 2015), we experimented with corrupting features based on attributions

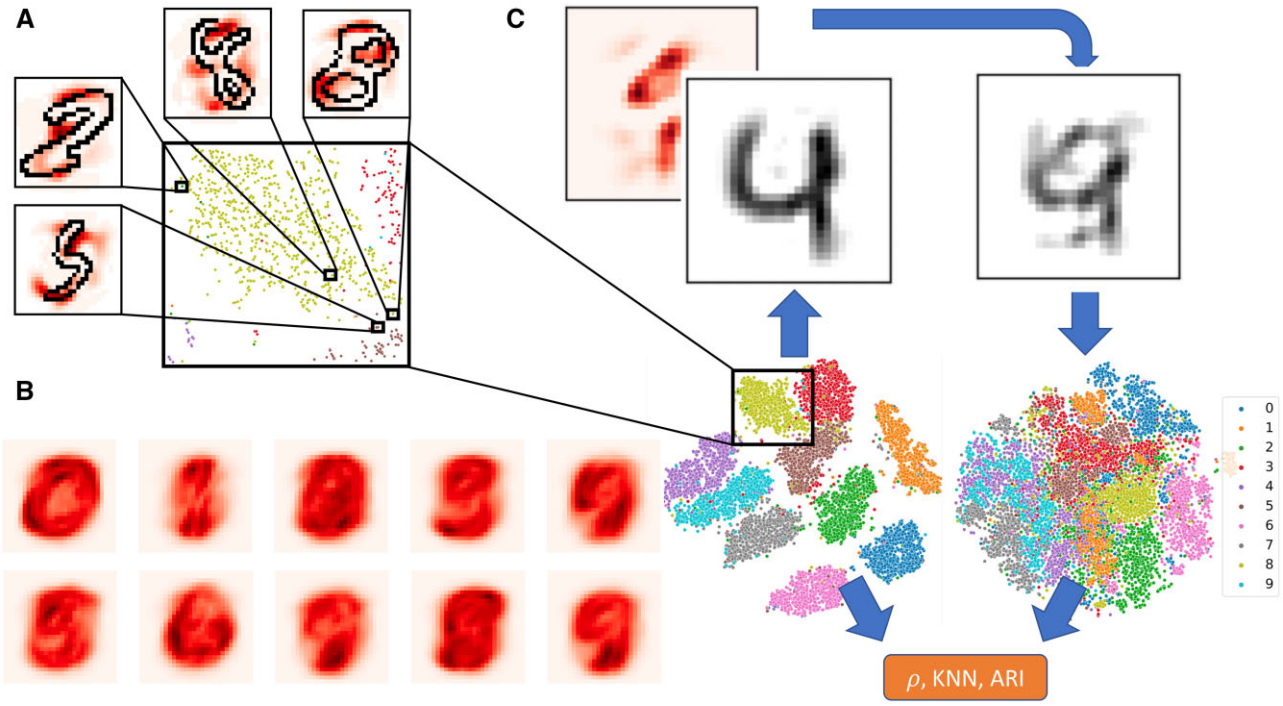

Figure 1. Overall description of method and schematic of validation experiment on MNIST dataset. (A) We display local attributions superimposed onto t-SNE embedded digits. (B) Attributions aggregated (via averaging) within each class. (C) We computed t-SNE embeddings and their corresponding attributions using the PCA transformed MNIST digits (left t-SNE plot). We then corrupted the digits based on their attributions (the heatmap and the digit 4 before and after corruption). Note that both digits are projected from PC space using the inverse of the PCA transformation. We then recomputed the t-SNE using this corrupted data as input (right t-SNE plot). We computed metrics, such as the Spearman correlation (ρ) of t-SNE embedded distances before and after the feature corruption. For (A), (B), and (C), we projected the attributions from PC space into pixel space by multiplying them by their corresponding PC loadings.

produced by positive gradients only and by multiplying the gradients by the inputs (and taking absolute value).

4.2.3 Methods of feature corruption

For our local and class level attributions, we corrupted each feature by setting all the values to be corrupted by the mean of those values. For the global-level attribution validation, we corrupted each feature by removing it from the dataset entirely. To ensure that our results were not biased by our corruption method, we experimented with an additional method of corruption: randomly permuting the values to be corrupted. We replicated all experiments using this permutation corruption method and present the results in Supplementary Appendix D.

4.2.4 Baselines

For each level of analysis and each percentage of features to be corrupted, we randomly sampled 10 subsets of features to be corrupted, and computed the change in correlation/10-KNN preservation/ARI to be used as our random baseline. For the individual level attributions, we corrupted a different random subset of features per sample. For class level validations, we corrupted a different random subset per class. For the global validations, we corrupted the same random subset of features for all samples.

For the global-level validation, we compared our method to the Laplace score, a popular unsupervised feature importance (He *et al.* 2005) method. We computed the Laplace score with respect to both P (matrix of p_{ij}'s) and Q (matrix of q_{ij}'s) used by t-SNE. In addition, we compared the method to the Fischer score, which can be seen as the supervised version of the Laplace score (He *et al.* 2005). We also compared the method to the top principal components, representing a variance-based control. For the class level validation, we computed a "class-based" Laplace score by re-computing the P and Q matrices on each class subset and then computing the Laplace scores. To compute the Fisher and Laplace scores, we used the python package `scikit-feature`.

At all levels, our final baseline was to select features using the absolute values of those features. For the class-based and global experiments, we selected features in an analogous manner as was done with our attribution-based methods, except that we substituted the feature values in place of the attributions. Refer to Fig. 1C for a schematic of the validation experiment.

4.3 SARS-CoV-2 case study

In order to demonstrate the practical utility of our method, we used it to investigate SARS-CoV-2 sequence data. The project has ethical approval from the Ethics Board of the Montreal Heart Institute, Project 2021-2868. We downloaded a globally representative sampling of 3064 SARS-CoV-2 via Nextstrain (Hadfield *et al.* 2018) accessed 26 January 2023. The sampling was done between December 2019 and January 2023. We intersect these with the codon-based alignment of GISAID (Elbe and Buckland-Merrett 2017) from 15 March 2023 resulting in a final dataset of size 2374 (EPI_SET ID EPI_SET_230418kp). The down sampling is due to the filtering perform by GISAID on missing data during the alignment process. We then recode as missing data any deletion >12 nt. We note that our dataset may be biased due to the sampling done by NextStrain. We derived the allele states from the Wuhan ancestral sequence (Gisaid ID: EPI_ISL_402124). The multiple sequence alignment (MSA) was performed using an optimized MSA procedure made by GISAID using MAFFT (Katoh *et al.* 2002). Each observed mutation or deletion at each position was encoded as a 1 if that mutation or deletion was present in the sequence and 0 otherwise. We ignored mutations or deletions that only occurred once in our dataset. Finally, we ignored any mutations occurring in the first or last 100 positions as these are less covered by the sequencing and thus of low quality. This left us with 33 250 mutations and 3359 when removing the reference allele.

For each sequence, we obtained Pangolin annotations (Rambaut *et al.* 2020, O'Toole *et al.* 2021) from GISAID, and used these to classify each sequence as belonging to either "Alpha," "Beta," "Delta," "Gamma," "Omicron": BA.1, BA.2, BA.4, BA.5, and BQ as designated by the World Health Organization (WHO). We labeled recombinant lineages, such as "XBB" separately.

We downloaded representative genetic markers for each lineage from outbreak.info (Tsueng *et al.* 2022). We removed markers containing deletions, since we were unable to identify the exact genetic positions of them.

5 Results

5.1 Qualitative results on MNIST dataset

We found that on the local level, our t-SNE attributions highlighted digit idiosyncrasies (see Fig. 1A). On the class-based level, we found that the digits highlighted pixels that varied within classes, but also seemed to suggest which digit classes would cluster together in the resulting t-SNE. For example, looking at the class-averaged attributions in Fig. 1B, we see that the averaged attributions of the 4's look very similar to those of the 7's and 9's, and indeed these three clusters appear next to each other in t-SNE space (almost forming their own "super cluster"). We observe the same pattern between the 3's, 5's, and 8's.

5.2 Local, class-level, and global attribution validation results

For each of the local, class, and global level, we found that our methods significantly outperformed the random baseline and were on par with or superior to the other baselines.

For the individual level baseline, we experimented with using only positive attributions. We found that these performed worse than just using the attributions themselves, and so we ignored them in subsequent experiments. We found that multiplying the attribution by the absolute feature value yielded the best 10-NN preservation (averaged across corruption %) at 0.20 ± 0.0024 versus the second-highest value of 0.28 ± 0.0035. Similarly, the ARI was 0.36 ± 0.0240 versus second best value of 0.38 ± 0.0129. The Spearman correlation was a close second to the feature value baseline: 0.35 ± 0.0361 versus 0.32 ± 0.0510.

For the class level experiments, we found that the attribution alone either outperformed or were on par with all other baselines (Spearman 0.49 ± 0.0992 versus 0.50 ± 0.0657, KNN Preservation 0.49 ± 0.0034 versus 0.50 ± 0.0019). For the global-level experiments, we found that both our gradient attribution-based methods either outperformed or performed on par with the other baselines. The full results can be found in Fig. 2 and in Supplementary Tables S2–S4. We highlight that our method is on par with other well-established methods

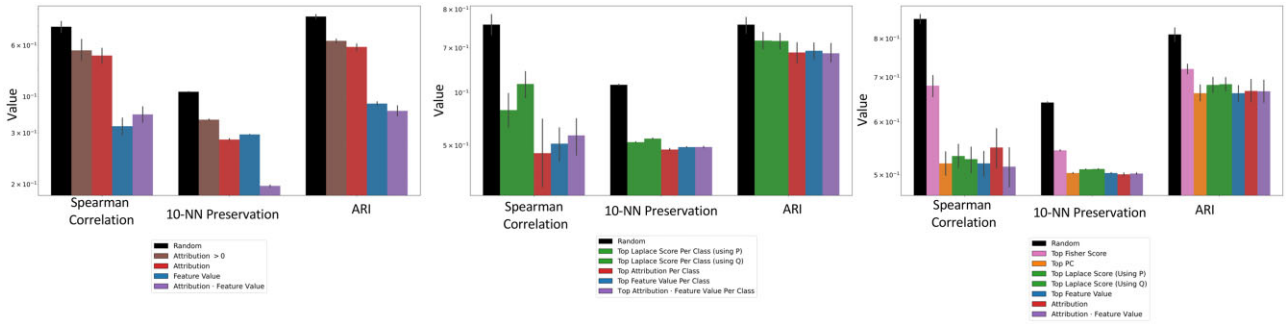

Figure 2. Individual, class, and global-level attribution validation experiments performed on MNIST. (Left=Local) We corrupted each feature using the mean of the sampled features to be corrupted. At each level, we compared the t-SNE embeddings before and after feature corruption using three metrics: the Spearman correlation, 10-nearest-neighbor preservation, and ARI (the *y*-axis). Note that lower values of each metric means that the corruption affected the embeddings more. Our baselines are (from left to right) random corruption, using only positive attributions, using only the attribution, using only the absolute feature values, and multiplying the attribution by the absolute feature values. (Middle=Class-based) We corrupted each feature using the mean of the sampled features to be corrupted. (from left to right) Random corruption, using the class-based Laplace score on matrices P and Q, using the attribution, using the feature, or multiplying the attribution by the feature. (Right=Global) We corrupted each feature by removing the features to be corrupted. Our controls are (from left to right) random corruption, using the Fisher score (supervised feature importance control), using the top principal components (variance-based control), using the Laplace score on matrices P and Q (unsupervised feature importance control), using the absolute value of the feature, the attribution, or multiplying the attribution by the absolute feature value. The error bars are 95% bootstrap CIs over the random seeds (and over sampling for our random baselines) computed using seaborn.barplot.

from the feature importance literature, despite being developed from the local feature attribution framework.

5.3 SARS-CoV-2 case study

We wanted to see if our t-SNE attribution method would assign high attribution to the mutations or deletions that we expected to be lineage defining. In order to do this, we needed to ensure that our t-SNE recapitulated the relevant lineage structure. We did this by inspecting a scatterplot of the t-SNE embeddings.

5.3.1 Using t-SNE attributions for quality control

Our initial SARS-Cov-2 encoding scheme did not yield t-SNE embeddings that clustered based on the WHO designations. This led us to perform an analysis of the t-SNE embeddings using our proposed attribution method. When we compared the attributions averaged within clusters generated by DBSCAN, we found that for several of the clusters, the attribution score was positively correlated with the missingness frequency. Given that the attributions were identifying missing values as the cause of certain clustering patters, we chose to impute this missing data as the reference genotype. For full details of our attribution-based QC, see Supplementary Appendix E.

When we computed a t-SNE of our imputed SARS-CoV-2 sequence dataset, we found that the sequences did generally cluster based on their WHO designation. As can be seen in the t-SNE scatterplot of Fig. 3A, most clusters correspond to a single lineage, with sub-lineages appearing as nearby subclusters. Note that there are some deviations in the observed scatterplot. For example, the clusters corresponding to sublineages of BA.5 (BA.5.1 and BA.5.2) do appear on opposite sides of the t-SNE scatterplot.

5.3.2 Identifying genetic markers from lineage-averaged attributions

Motivated by the apparent utility of class-averaged attributions when used with MNIST, we averaged the attributions of each mutation/deletion per lineage and compared this to the mutation/deletion frequency. Note that the mutation/deletion frequency is a feature average since we encoded each mutation/deletion as a binary variable.

We chose the 90th percentile to be our threshold of significance when identifying mutations/deletions based on attribution scores or mutation/deletion frequency. Of the 267 markers, we found that 251 could be identified by having significantly high mutation frequency, while 229 could be identified by having high attribution. However, three markers were identified using attributions that had low frequency. Thirteen markers could not be identified using either the attributions or mutation frequency. This can be seen in Fig. 3C, where the markers detected by attributions and not feature means appear in the top left quadrant, and the 13 markers not detected by either method appears in the bottom left quadrant.

The attribution-based method uniquely identified the Omicron BA.1 marker Spike: G142D and the Alpha and BA.2 marker ORF8: L84S. Both methods missed Spike: N440K (BA.2, BA.4, and BA.5 marker) as well as Spike: N679K (BA.1, BA.2, BA.4, and BA.5) and ORF8: L84S (for Beta, Delta, Gamma, BA.1, BA.4, and BA.5). Of the 25 markers missed by our attribution-based method, 21 of them were markers of Gamma, 2 from Beta, and 2 from Omicron BA.4. We suspect that our approach had difficulty identifying these markers because their lineages were the least frequent within our dataset (among the sequences that had markers). In fact, the dataset contained only 34, 49, and 58 sequences of Gamma, Beta, and BA.4, respectively.

Finally, we note that our highly attributed mutations were corroborated in the literature. For example, a previous study (Mostefai *et al.* 2022) identified 25 "Haplotype defining" mutations (highly predictive of SARS-CoV-2 evolutionary structure). Twenty-four of these positions were highly attributed by our method.

6 Discussion

To the best of our knowledge, this is the first application of a feature attribution method to any dimensionality reduction algorithm. Furthermore, we develop a novel validation method, and provide a biologically relevant demonstration. We note that the algorithm presented provides feature attributions

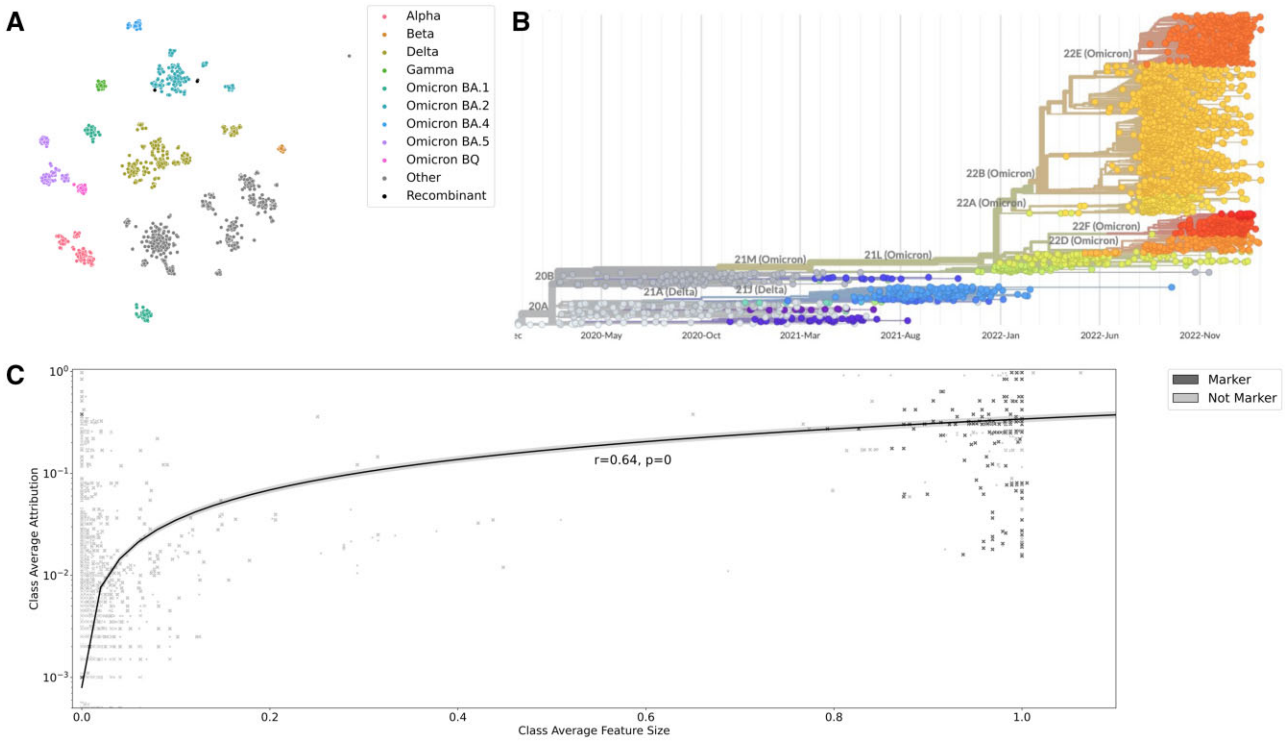

Figure 3. Finding genetic markers for SARS-CoV-2 lineages. (A) The t-SNE embeddings of our SARS-CoV-2 dataset. We added the additional "Omicron BQ" and "Recombinant" categories. (B) Phylogenetic tree fit via Nextstrain on sequences used in this study (Hadfield *et al.* 2018). (C) We averaged the attributions per mutation for each lineage, and plotted them against the mutation/deletion frequency. We colored the points based on whether they were a marker gene as determined by outbreak.info. Points marked as * are synonymous, and points marked as X are non-synonymous. The dashed lines on the *x* and *y* axes indicate the 90th percentile for the mutation/deletion frequency and averaged attribution, respectively.

with respect to a given t-SNE embedding. Therefore, any insights yielded by the attribution scores only represent "true signal" from the data insofar as the t-SNE embedding has modeled the data appropriately. This is demonstrated in Fig. 1C, where the t-SNE embeddings for the three- and five-digit classes are very similar, and indeed the t-SNE embedding has both digit classes adjoined, and not fully resolved on their own. Our method can identify such algorithmic artifacts, which can be useful for practitioners who want to understand why their embeddings appear a certain way, without having to do *ad-hoc* feature enrichment analysis.

In practice, we suggest that users analyze the attributions of high-quality t-SNE embeddings. There exist metrics that quantify t-SNE embedding quality (Lee and Verleysen 2009, Kobak and Berens 2019). We suggest that practitioners use them to filter out potentially problematic t-SNE embeddings prior to attribution analysis.

Our MNIST data exists in a human understandable space, and so we can visualize our attributions at each level, and this provides a sanity check for our method. Qualitatively, we found that our attributions yielded human understandable insights about the variation of individual digits and the defining characteristics of each MNIST digit class that were recapitulated by the t-SNE embedding.

On all levels, we found that the attributions produced by our methods significantly outperformed random feature corruption. We are not surprised that our method did not always outperform baselines, particularly at the class-based and global level, given that our method is a local feature attribution method. We hope that the development of this method could inspire future research, to eventually develop less noisy variations of our approach.

We note that throughout this work, we implicitly assume that the ground truth feature dependencies are somewhat sparse (i.e. only a few features driving the structures recapitulated by t-SNE). This assumption appears to hold for the datasets used here. In cases where the data exhibits complex relationships between features and structures, it is not clear if one should use feature attribution methods since such relationships may not be well represented by per-sample per-feature scores.

In the SARS-CoV-2 application, we further found that aggregating lineage-averaged feature attribution scores identified significant variations within SARS-CoV-2 lineages. We note that other methods exist for finding markers mutations for SARS-CoV-2 variants, and these have been used extensively to analyze SARS-CoV-2 data in the last 3 years. This is precisely the information that we wanted to leverage to confirm the validity of our attributions in a biological application. In contrast, the ground truth of attributions in other biological modalities, such as transcriptomics, metagenomics, or metabolomics can be harder to establish, making the evaluation of attributions trickier. Our approach is not meant as a replacement for other methods, but the ample domain expertise in this field made it appropriate a point of reference to assess our method. Nonetheless, our method could be used on sequence datasets from future waves to identify quickly new sub-lineages arising and to identify outlier sequences to be removed.

We anticipate that this work can be extended in multiple ways. First, we would like to see this method applied to more real-world biological data science applications (including gene expression, protein interaction, metagenomics, and metabolomics). We are particularly intrigued by applications

in the single-cell RNA transcriptomics domain, where t-SNE analysis is particularly popular (Kobak and Berens 2019). However, since ground truth is generally missing in these applications, simulation work will be needed to validate the approach (Zappia *et al.* 2017). The algorithmic complexity of our method scales roughly linearly in terms of the number of input features when compared to the usual t-SNE. This is due to additional computations of large, multidimensional arrays. Increasing the efficiency of these computations is a second promising extension. Permutation-based attribution methods, such as SHAP (Lundberg and Lee 2017) have nice mathematical guarantees, but a naive application of such methods would require an infeasible number of model evaluations. Being able to adapt such methods to this problem setting represents a third possible future direction for this research.

7 Conclusion

We propose a feature attribution method designed for t-SNE. To the best of our knowledge, this represents the first such attempt for any dimensionality reduction algorithm. In fact, this is also the first attempt to do attribution of a non-parametric ML algorithm. We argue that since both methods are optimized via SGD, the gradient with respect to inputs represent the same thing.

We developed a method that evaluates the validity of our approach. Our method quantifies the feature attribution performance by comparing the extent of degradation of t-SNE embeddings post-corruption. We chose baselines from the unsupervised feature importance literature. We also compared our method with feature enrichment baselines, and with appropriate random baselines.

We demonstrated our algorithms correctness using synthetic data, where we knew the significant features available. We then evaluated our algorithm on MNIST. Here, we did not have the significant features known in advance, but were able to provide evidence for our approach using our validation method. Finally, we demonstrate the utility of our method via a SARS-CoV-2 case study, finding that in all cases our approach yielded unique insights that could help a data scientist better understand their t-SNE plot. We hope that this work can serve as the foundation for other works investigating the use of feature attributions for dimensionality reduction algorithms.

Acknowledgements

We would like to thank all laboratories that contributed to GISAID sequences.

Author contributions

Matthew Crispin Scicluna (Conceptualization [lead], Formal analysis [lead], Investigation [lead], Methodology [lead], Project administration [lead], Software [lead], Validation [lead], Visualization [lead], Writing—original draft [lead]), Jean-Christophe Grenier (Data curation [lead], Methodology [supporting], Software [supporting], Visualization [supporting], Writing—review & editing [supporting]), Raphael Poujol (Data curation [lead], Methodology [supporting], Software [supporting], Writing—review & editing [supporting]), Sebastien Lemieux (Conceptualization [supporting], Supervision [supporting]), and Julie Hussin (Conceptualization [supporting], Funding acquisition [lead], Project administration [supporting], Resources [lead], Supervision [lead], Writing—review & editing [lead])

Supplementary data

Supplementary data are available at *Bioinformatics Advances* online.

Conflict of interest

None declared.

Funding

J.G.H. is a Fonds de la Recherche du Québec en Santé (FRQS) Junior 2 Scholar [FRQS 311067]. This work was supported by grants from the Natural Sciences and Engineering Research Council (NSERC) [RGPIN-2022-04262 to J.G.H.]; the Institute for Data Valorization IVADO [CVD19-030 to J.G.H.]. This study was also supported by the Canadian Institute of Health Research (CIHR) operating grant to the Coronavirus Variants Rapid Response Network (CoVaRR-Net).

Data availability

The SARS-CoV-2 data underlying this article are available in GISAID (https://gisaid.org). The other datasets were derived from sources in the public domain (https://www.kaggle.com/datasets/hojjatk/mnist-dataset, https://www.kaggle.com/datasets/crawford/20-newsgroups).

References

Adebayo J, Gilmer J, Muelly M *et al.* Sanity checks for saliency maps. In: Bengio S, Wallach H, Larochelle H *et al.* (eds), *Advances in Neural Information Processing Systems*, NeurIPS 2018. Montreal, QC, Canada, 3-8 December 2018.

Bach S, Binder A, Montavon G *et al.* On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One* 2015;**10**:e0130140.

Brendel W, Bethge M. Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet. In: *7th International Conference on Learning Representations, ICLR 2019. New Orleans, LA, USA, 6–9 May 2019*. OpenReview.net, 2019.

Elbe S, Buckland-Merrett G. Data, disease and diplomacy: GISAID's innovative contribution to global health. *Glob Chall* 2017;**1**:33–46.

Hadfield J, Megill C, Bell SM *et al.* Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics* 2018;**34**:4121–3.

He X, Cai D, Niyogi P. Laplacian score for feature selection. In: Weiss Y, Schölkopf B, Platt J (eds), *Advances in Neural Information Processing Systems*, Vol. **18**. MIT Press, 2005.

Hooker S, Erhan D, Kindermans P *et al.* A benchmark for interpretability methods in deep neural networks. In: Wallach HM, Larochelle H, Beygelzimer A *et al.* (eds), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8–14 December 2019, Vancouver, BC, Canada*. 9734–45. 2019.

Katoh K, Misawa K, Kuma K *et al.* MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res* 2002;**30**:3059–66.

Kobak D, Berens P. The art of using t-SNE for single-cell transcriptomics. *Nat Commun* 2019;**10**:5416.

Lee JA, Verleysen M. Quality assessment of dimensionality reduction: rank-based criteria. *Neurocomputing* 2009;**72**:1431–43.

Lipton ZC. The mythos of model interpretability. *Commun ACM* 2018; **61**:36–43.

Lundberg SM, Lee S. A unified approach to interpreting model predictions. In: Guyon I, von Luxburg U, BengioS *et al.* (eds), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA*, 4765–74. 2017.

McInnes L, Healy J, Saul N *et al.* UMAP: Uniform manifold approximation and projection. *JOSS* **3**:861.

Moon KR, Van Dijk D, Wang Z *et al.* Visualizing structure and transitions in high-dimensional biological data. *Nat Biotechnol* 2019;**37**: 1482–92.

Mostefai F, Gamache I, N'Guessan A *et al.* Population genomics approaches for genetic characterization of SARS-CoV-2 lineages. *Front Med (Lausanne)* 2022;**9**:826746.

O'Toole I, Scher E, Underwood A *et al.* Assignment of epidemiological lineages in an emerging pandemic using the pangolin tool. *Virus Evol* 2021;**7**:veab064.

Plumb G, Terhorst J, Sankararaman S *et al.* Explaining groups of points in low-dimensional representations. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13–18 July 2020, Virtual Event, Vol. 119 of Proceedings of Machine Learning Research*. 7762–71. PMLR, 2020.

Rambaut A, Holmes EC, O'Toole Á *et al.* A dynamic nomenclature proposal for SARS-CoV-2 lineages to assist genomic epidemiology. *Nat Microbiol* 2020;**5**:1403–7.

Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?": explaining the predictions of any classifier. In: Krishnapuram B, Shah M, Smola AJ *et al.* (eds), *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016*. 1135–44. ACM, 2016.

Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. In: BengioY, LeCun Y (eds), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014, Workshop Track Proceedings*. 2014.

Springenberg JT, Dosovitskiy A, Brox T *et al.* Striving for simplicity: the all convolutional net. In: Bengio Y, LeCun Y (eds), *3rd International Conference on Learning Representations, ICLR 2015, Sn Diego, CA, USA, 7–9 May 2015, Workshop Track Proceedings*. 2015.

Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: Precup D, Teh YW (eds), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, Vol. 70 of Proceedings of Machine Learning Research*. 3319–28. PMLR, 2017.

Tan S, Hooker G, Koch P *et al.* Considerations when learning additive explanations for black-box models. *Mach Learn* 2023;**112**: 3333–59.

Tibshirani R. Regression shrinkage and selection via the lasso. *J R Stat Soc Series B Methodol* 1996;**58**:267–88.

Tsueng G, Mullen JL, Alkuzweny M *et al.* Outbreak.info research library: A standardized, searchable platform to discover and explore COVID-19 resources. *Nat Methods* 2023;**20**:536–40.

van der Maaten L. Accelerating t-SNE using tree-based algorithms. *J Mach Learn Res* 2014;**15**:3221–45.

van der Maaten L, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008;**9**:2579–605.

Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol* 2017;**18**:174.

Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: Fleet DJ, Pajdla T, Schiele B *et al.* (eds), *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, 6–12 September 2014, Proceedings, Part I, Vol. 8689 of Lecture Notes in Computer Science*. Springer. 2014, 818–33.