Novel Parameter-Free and Parametric Same Degree Distribution-based Dimensionality Reduction Algorithms for Trustworthy Data Structure Preserving

Laureta Hajderanj, Daqing Chen, Sandra Dudley, Guillaume Gilloppe, and Baptiste Sivy

London South Bank University, SE1 0AA, London, UK.
{hajderal,chend,dudleyms}@lsbu.ac.uk,
{guillaume.gilloppe, baptiste.sivy}@etu.univ-nantes.fr

Abstract. As an effective dimensionality reduction method, Same Degree Distribution (SDD) has been demonstrated to be able to maintain better data structure than other dimensionality reduction methods, including Principal Component Analysis (PCA), Multidimensional Scaling (MDS), Isomap, Locally Linear Embedding (LLE), Laplacian Eigenmaps (LE), Uniform Manifold Approximation and Projection (UMAP) and t-Stochastic Neighbor Embedding (t-SNE). In addition, SDD does not require tuning the number of neighbours or perplexity to scale the structure capturing performance. Instead, it requires tuning the degree of degree-distribution ranging in e certain interval. Hence, tuning the degree of degree-distribution makes SDD a less costly method than other methods that require tuning the number of neighbours or perplexity. Although these advantages, SDD is still an expensive method compared with parameter-free methods such as PCA and MDS. A parameter-free SDD is proposed based on standard SDD, with two main differences: 1) it does not require tuning the degree of degree-distribution in the entire range from 1 to 15, but only uses degree 1; and 2) it re-scales the pairwise distances in the range [0, 2] instead of range [0, 1]. A theoretical analysis is presented to prove the better performance of parameter-free SDD. In addition, the performances of the proposed parameter-free SDD and the standard SDD have been experimentally compared in terms of structure capturing and computational time. This paper also proposes a parametric version of SDD using a deep neural network approach to learn the mapping based on the samples of the original data and their corresponding embedded representations in a low dimensional space. Comparative experiments have been undertaken with SDD and other methods such as Isomap, t-SNE and UMAP to demonstrate the effectiveness of the proposed parametric SDD with several popular synthetic and real datasets such as Churn, SEER Breast Cancer, AVletters (LIPS Reading) and MNIST.

Keywords: high dimensional data, dimensionality reduction techniques, structure capturing, computational time

1 Introduction

Analysing and exploring high-dimensional data remains a challenging task since a high number of dimensions has made it difficult to understand and interpret data. Researchers have been seeking effective ways to facilitate exploring high dimensional data for information of interest, and dimensionality reduction (DR) has been considered an effective means for visualising high dimensional data and handling the problem of the curse of dimensionality. In terms of visualisation, a good dimensionality reduction technique is a technique that can capture the best data structure [25].

Maintaining data structure means that close data points in the high dimensional space to be embedded closely in the low dimensional space, and distant data points in the original high dimensional space to be embedded distant in the low dimensional space. The methods that favour global structure capturing are Principal Component Analysis(PCA)[1], Multidimensional Scaling (MDS)[2], and Isomap[8], whereas local structure capturing is supported by Locally Linear Embedding (LLE)[9], Laplacian Eigenmaps (LE)[10], t-Stochastic Neighbour Embedding (t-SNE)[19], Uniform Manifold Approximation and Projection (UMAP)[24], and Same Degree Distribution (SDD)[25]. Although they favour different data structures, most methods are constrained to the type of data to be considered. The performance of a dimensionality reduction method also depends on tuning parameters, such as the number of neighbours (k), perplexity (pr), which is also a costly and time-consuming process. Methods that require parameter tuning are Isomap (k), LLE (k), LE (k), t-SNE (pr), and they are the state-of-the-art methods in terms of their ability into capturing the data structure. SDD is another state-of-the-art method, which, despite its good performance, still requires tuning the degree (deg), and, although it is less expensive than tuning the number of neighbours (k) or perplexity (pr), it still requires more time than parameter-free [1] methods such as MDS and PCA. In summary, the existing dimensionality reduction approaches suffer from the following problems: 1) Only work well with linear data, 2) The scale of maintained data structure is related to the number of data samples considered.

To deal with all the above-mentioned problems, this paper has developed parameter-free SDD approache that 1) Saves computational time because it does not require tuning parameter to achieve the best data structure, and 2) Produces more trustworthy visualisation by using degree-distribution with $deg = 1$ that is smooth enough to capture local and global data structure. Also, parameter-free SDD does not suffer from tear and false neighbours problems due to using degree-distribution with $deg = 1$ in high and low dimensional spaces to calculate the similarities between data samples.

Parameter-free SDD is an extension of SDD that manages to capture the same data structure as SDD due to using degree distribution with $deg = 1$,

[1] Parameter-free methods refer to as those that do not require tuning of parameters such as number of neighbours, perplexity etc., but it does not include optimisation parameters such as learning rate and so on.

and re-scaling pairwise distances in the interval [0, 2]. The reason for re-scaling pairwise distances in the range [0, 2] is that degree-distribution with $deg = 1$ provides a broader range of similarity to short distances, which is a problem that degree-distribution with $deg = 1$ has in case of re-scaling pairwise distances in the range [0, 1]. The relevant theoretical analyses are provided along with comparative experiments to demonstrate the super performance of the proposed method.

The main contributions of this paper are: 1) a parameter-free SDD which achieves the same results as SDD in significantly less computational time and resources, and 2) a parametric SDD that can reduce the dimensionality of new interest data points without requiring extra computational time and resources.

The rest of this paper is structured as follows; Section 2 presents a comprehensive related work, which presents a detailed view on the strengths and limitations of most dimensionality reduction techniques. Section 3 presents the proposed approaches, followed by parametric SDD in Section 4 and conclusions are presented in Section 5.

2 Related Works

2.1 Dimensionality Reduction

This Section presents a wide range of dimensionality reduction methods, emphasising their drawbacks and advantages. Methods have been compared considering the type of data and parameters [2]. The type of data is *linear* or *nonlinear* data. Note that nonlinear data are much more complex, and their low dimensional representative lies on a nonlinear manifold. There are also some data that their low dimensional representation lies on more than one manifold.

From a practical viewpoint, high dimensional data is not necessarily always high dimensional, as the data analysis community have agreed that high dimensional data points reside on low-dimensional manifolds [26]. In other words, a large number of variables can be represented by a smaller set of new variables, with no or less redundancy [3]. From a theoretical perspective, all the difficulties that arise when faced with high dimensional data are related to *curse of dimensionality*. The *curse of dimensionality* is a phenomenon occurred when working with high dimensional data, which refers to the situation where the number of samples required increases exponentially with the number of variables to estimate a function of several variables to give a certain accuracy [26]. To address the issues associated with the curse of dimensionality, two main approaches and techniques have been explored: *Feature Selection* and *Feature Extraction*. *Feature Selection* is closely related to keeping some original variables highly correlated with the label variable, and the rest can be eliminated. *Feature Extraction* transforms an original set of variables into a set of new variables with a reduced

[2] Parameter include all parameters that each technique has to tune to achieve the best by excluding optimisation parameters

[3] i.e. variables are independent of each other.

number of variables by preserving their inherent characteristics. This presented research focuses on feature extraction, which will be referred to as dimensionality reduction in the rest of the paper.

The dimensionality reduction methods presented in Table 1 will be compared based on the two main criteria:

1. *Parameters*[4] refers to the parameters that impact the performance type of the dimensionality reduction method.
2. *Type of data* refers to the shape of a manifold (linear or nonlinear) that contains the low dimensional representation of the original data.

Table 1: DIMENSIONALITY REDUCTION METHODS

Year	DR algorithm	Parameters	Type of Data	References
1901	Principal Component Analysis (PCA)	none	linear	[1]
1962	Multidimensional Scaling (MDS)	none	linear	[2]
1969	Sammon Mapping	none	nonlinear	[3]
1997	Curvlinear Commponent Analysis (CCA)	λ	nonlinear	[4]
1997	Curvelinear Distance Analysis (CDA)	λ	nonlinear	[4]
1997	Generative Topographic Embedding (GTE)	$K(.,.)$	nonlinear	[5]
1998	Kernel PCA (KPCA)	$K(.,.)$	nonlinear	[6]
1998	Self-organizing Maps (SOM)	σ ,v_λ	nonlinear	[7]
2000	Isomap	k	nonlinear	[8]
2000	Locally Linear Embedding (LLE)	k	nonlinear	[9]
2001	Laplacian Eigenmaps (LE)	k, σ	nonlinear	[10]
2003	Hessian Locally-Linear Embedding (HLLE)	k	nonlinear	[11]
2004	Maximum Variance Unifolding (MVU)	k	nonlinear	[12]
2005	Nonlinear PCA	$NetSize$	nonlinear	[13]
2005	Local Tangent Space Alignment (LTSA)	k	nonlinear	[14]
2006	Diffusion Maps	σ, t	nonlinear	[15]
2006	Autoencoders	$NetSize$	nonlinear	[16]
2007	Modified Locally Linear Embedding (MLLE)	k	nonlinear	[17]
2007	Data-Driven High-Dimensional Scaling (DD-HDS)	$\lambda 1$	nonlinear	[18]
2008	t-Stochastic Neighbor Embedding (t-SNE)	pr	nonlinear	[19]
2008	Manifold Sculpting	k	nonlinear	[20]
2009	RankVisu	k	nonlinear	[21]
2010	Topologically Constrained Isometric Embedding (TCIE)	k	nonlinear	[22]
2018	Trimap	k	nonlinear	[23]
2018	Uniform Manifold Approximation and Projection (UMAP)	k	nonlinear	[24]
2020	Multi Same Degree Distributions (MSDD)	deg	nonlinear	[25]

Some symbols and parameters used throught this paper are clarified as following:

[4] Parameter include all parameters that each technique has to tune to achieve the best by excluding optimisation parameters.

N	Number of samples
n	Number of degree-distributions
D	Number of dimensions of the original data
d	Number of dimensions in low dimensions space data
$X^{N\times D}$	Original space dataset
$Y^{N\times d}$	Low dimensional space dataset
k	Number of neighbours
pr	Perplexity
$dis(a,b)$	Euclidean distance between data samples a and b
τ	Kendall's Tau
λ	User-defined parameter
λ_1	User-defined parameter in DD-HDS method
$K(.,.)$	Kernel function
σ	Density in Gaussian distribution
t	Timestep in Diffusion Maps

PCA and MDS are two linear dimensionality reduction methods that do not require parameter tuning. Although they can save computational time, they neglect the maintenance of the local data information.

Sammon's mapping has been considered a nonlinear version of MDS, which unfolds non-heavy manifolds; however, Sammon's mapping boosts the contribution of very close data points in the cost function employed to the method and neglects the preservation of the global data structure. It also fails to unfold heavy manifolds (complex manifolds).

CCA is another variant of MDS by modifying the cost function with a parameter F_λ, where

$$F_\lambda = H(\lambda - dis(y_i, y_j)) \tag{1}$$

where y_i and y_j are two data points in a dataset and $H(u)$ is defined as follows:

$$H(u) = \begin{cases} 0 & if\, u \leq 0 \\ 1 & if\, u \geq 0 \end{cases} \tag{2}$$

The performance of CCA depends on the hard border of λ, which is a user-defined parameter. Because F_λ depends on the distances on the embedded space, CCA is prone to *tearing*.

A different variant of MDS is Isomap, which uses Geodesic distance instead of Euclidean distance for calculating the distance between high dimensional data samples. However, Isomap can only be successfully used for developable manifolds, not for nondevelopable manifolds such as a sphere or a hollow piece, or if any hole exists in manifolds [26]. Furthermore, the performance of Isomap depends on tuning the number of neighbours, which is a very time-consuming process.

Geodesic Sammon's mapping and CDA are two variants of Sammon's mapping and CCA, employing Geodesic distances instead of Euclidean distances to calculate distances between high dimensional data points. However, they have

the same limitations as Sammon's Mapping and CCA, and also they are limited to only developable manifolds. Kernel PCA suffers from a tedious process in selecting a right kernel and its parameters. Some existing kernels include

1. Polinomial Kernel :

$$\kappa(u, v) = (\langle u \cdot v \rangle + 1)^{int} \tag{3}$$

2. Gaussian kernels:

$$\kappa(u, v) = exp\left(-\frac{\|u - v\|^2}{2\sigma^2}\right) \tag{4}$$

3. MLP Kernel :

$$\kappa(u, v) = tanh(\langle u \cdot v \rangle + b) \tag{5}$$

MVU suffers from the short-circuiting problem, and MVU captures global data structure, although its main aim is to preserve local data structure.

SOM and GTM are two methods aiming to preserve the topology of data. However, the topology to be considered is pre-defined, and as a result, these methods do not perform well if the topology data manifold is not the same with the pre-defined topology. Furthermore, these methods depend on parameter tuning, which is a tedious task.

LLE, LTSA, HLLE and LE perform well with smooth manifolds, and their performance is related to the number of neighbours.

Diffusion map requires tuning of parameters time and and number of neighbours (k), and it performs better in capturing global structure.

DD-HDS is a variant of MDS, but their performance is related to a user-defined parameter with a positive value ranging between 0.1 and 0.9. It should be noted that this method is not good in preserving the global structure of data due to the sharptails of the Gaussian distribution employed. Furthermore, employing the Gaussian distribution makes this method more expensive. RankVisu is similar to DD-HDS; however, it considers the rank of neighbours instead of distances and, the performance of this method depends on the number of neighbours.

All the above-mentioned methods work well with linear, simple smooth, or developable manifolds. However, suppose the low dimensional representation is located on heavily curved manifolds. In that case, a group of methods such as SNE, t-SNE, UMAP and TriMap are able to capture the structure of data. Furthermore, in t-SNE, UMAP, and TriMap, Gaussian and Student-t distributions are introduced to provide a softer border between local and global structure maintenance. However, as previously discussed, these methods require tuning the number of neighbours (perplexity) to generate the best low dimensional representation in maintaining the data structure. Multiscale approaches such as Multiscale-SNE attempted to overcome this shortcoming; however, it is still a costly method because both the multiscale calculations and the utilisation

of Gaussian distribution consumes much more computational time than using Student-t distribution. Note that, employing different distributions in high and low dimensional spaces, respectively, can lead to *tears* and *false neighbours* problems [26].

Considering the problems of t-SNE, UMAP, and TriMap, the Same Degree Distribution (SDD) method has been proposed to capture a better geometry of data by employing the same degree-distribution(s) in the high and the low dimensional spaces. SDD also saves significant computational time by employing degree-distribution, which does not need to calculate perplexity or number of neighbours but the degree of degree-distribution. Also, using degree-distribution to calculate high and low similarities is less computational since a degree-distribution does not include exponential in the formula as Gaussian distribution does. In summary, SDD takes all advantages of t-SNE, UMAP, and TriMap over all other dimensionality reduction methods, including its ability to capture the data structure, having their low dimensional representation located on non-smooth, non-developable manifolds, and even in more than one manifold. On the other hand, SDD overcomes t-SNE, t-SNE, UMAP, and TriMap performances by maintaining a better data structure and in less computational time.

2.2 SAME DEGREE DISTRIBUTION (SDD) APPROACH

Kullback-Leibler is the loss function used in SDD to approximate the degree-distribution in the low dimensional space with the degree-distribution in the high dimensional space:

$$C_1 = \sum_{i \neq j} (p_{deg_m})_{ij} \log(\frac{(p_{deg_m})_{ij}}{(q_{deg_m})_{ij}} \tag{6}$$

where deg_m is the degree of degree-distribution m, $m = 1 : n$. SDD intends to minimize the cost function C_1 as (6):

$$loss_1 = \min(C_1) \tag{7}$$

where

$$(p_{deg_{ij}})_{ij} = \frac{(1 + dis(x_i, x_j))^{-deg_m}}{\sum\limits_{k \neq l} (1 + dis(x_k, x_l))^{-deg_m}} \tag{8}$$

$$(q_{deg_{ij}})_{ij} = \frac{(1 + dis(y_i, y_j))^{-deg_m}}{\sum\limits_{k \neq l} (1 + dis(y_k, y_l))^{-deg_m}} \tag{9}$$

Despite all the benefits of using SDD, there exist some limitations: 1) SDD still requires tuning the degree of degree-distribution, and it is costly since it consumes significant computational resources; and 2) SDD is a non-parametric

method, and as such, the data embedding process needs to re-start whenever a new data point becomes available and needs to be considered. In the following section, a parameter-free method is proposed and discussed in detail, aiming to better capture data structure, particularly improving local data structure capturing by increasing the range of distances. In addition, a parametric SDD has also been proposed in Section 4.

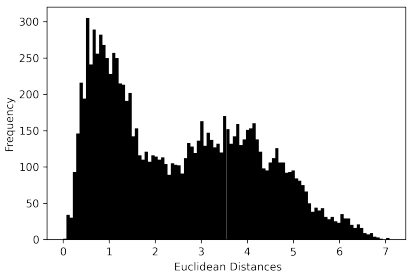
3 Proposed Parameter-free SDD Approach

The all benefits using the SDD comes from the degree-distribution with degree 1 ($deg = 1$), which is equivalent to Student-t distribution with $deg = 1$. Student-t distribution generates higher similarity values to short distances, and values decrease smoothly as distance increases. When distance increases infinitely, the Student-t distribution with $deg = 1$ approaches to zero, making Student- t distribution unfeasible to capture of data with large distances. To deal with this, Hajderanj et al.[25] proposed re-scaling the pairwise distances of the original data into the range [0, 1]. It has been demonstrated that the original data X re-scaled by a single value (maximum value of pairwise distances), then the distribution of pairwise distances of the re-scaled data is the same as the distribution of pairwise distances of the original data X, as shown in Figs 1. (a), (b), and (c).

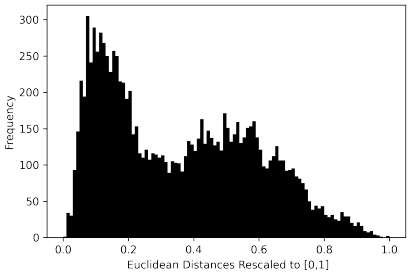
Re-scaling the pairwise distances of the original data is the key to the success of SDD, which has been demonstrated to be capable to capture a good structure of the data; however, SDD still requires tuning the degree (deg) of degree-distribution, which normally ranges from 1 to 15. As seen in Fig. 2, degree-distributions with degrees $deg = 20$ and $deg = 5$ are not sensitive to distances between 0.5 and 1, and that means they can not captures the neighbourhood structure of far away data points.

To evaluate if close (far away) neighbours in the original high-dimensional space are kept close (far away) in the embedded low dimensional space, two metrics are commonly used: *Trustworthiness* and *Continuity*. Trustworthiness measures the far away data points that embedd close in the low dimensional space, whereas Continuity measures close data points that embedd far away in the low dimensional space. As shown in Figs 3. (a) and (b), measured by Trustworthiness and Continuity, degree-distribution with $deg = 1$, demonstrated a poorer performance in maintaining the local data structure than degree-distribution with $deg = 7$ and $deg = 15$ (where under consideration is a small number of neighbours). However, in situations where the number of neighbours under consideration is large, degree-distribution with $deg = 1$ shows a better performance in capturing the global data structure than the degree-distribution with $deg = 7$, shown in Fig. 4.

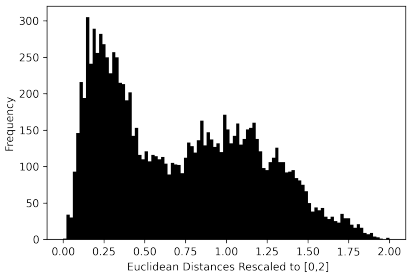
Overall, using the degree-distribution with $deg = 1$ is similar to using the degree-distribution with best degree ($deg = 7$). However, it can be shown that degree-distribution with $deg = 1$ does not perform as good as degree-distribution with $deg = 7$ in short distances (a small number of neighbours under consid-



(a)



(b)



(c)

Fig. 1: Euclidean Distance (a), re-scaled Euclidean distance to [0, 1] (b) and re-scaled Euclidean distance to [0, 2] (c).

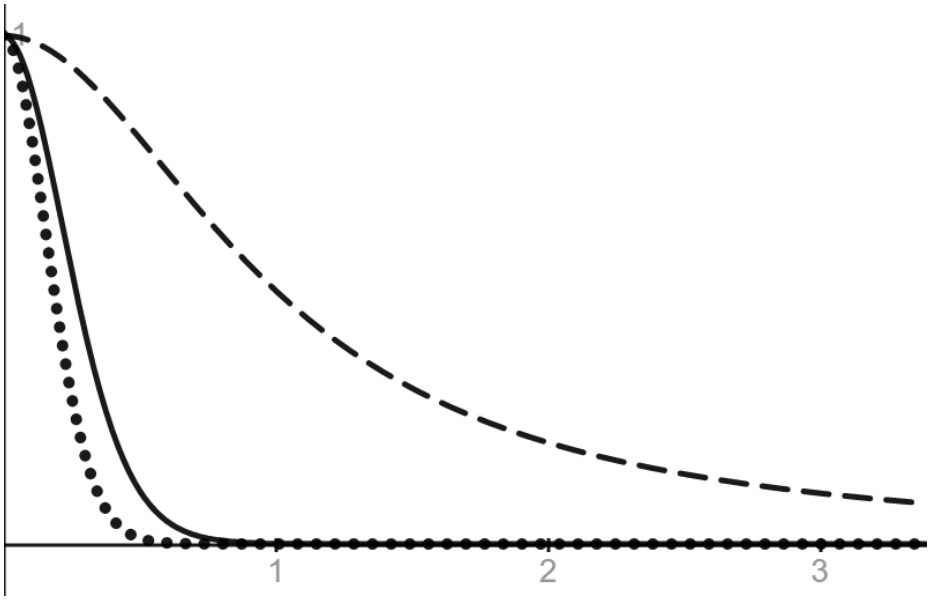


Fig. 2: Three degree-distributions with $deg = 1$, $deg = 5$ and $deg = 20$.

eration). To deal with that issue, it is proposed in the research to increase the range of pairwise distance of the original data in [0, 2].

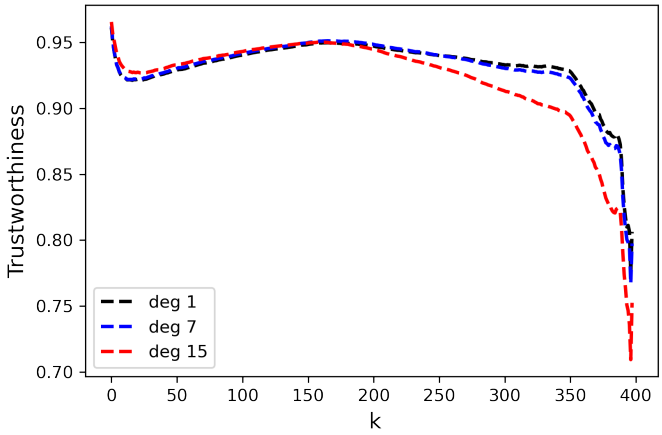
3.1 Idea and Theoretical Proof

Re-scaling pairwise distances in the range [0, 2] can generate a wider range of similarity, as shown in Fig. 5. The range of similarity generated by degree-distribution with $deg = 1$ has pairwise distances in [0, 1], ranges in the interval [1, 0.5], whereas the similarity generated by degree-distribution with $deg = 1$ has pairwise distances in [0, 2], ranges in a wider interval [1, 0. 2]. Moreover, it can be theoretically proven that re-scaling the pairwise distances of the original data in the range [0, 2] generates a wider range of similarity produced by degree distribution with $deg = 1$, as in Proposition 1 below.

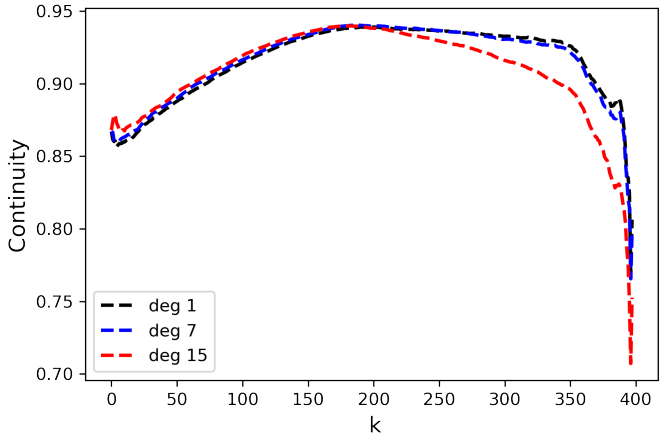
Proposition 1 By increasing the rescaled distance range interval, the similarity range of degree-distribution $p_{ij} = \frac{(1+dis(x_i,x_j))^{-1}}{\sum_{k \neq l}(1+dis(x_k,x_l))^{-1}}$ also increases.

Proof

Let's define with $d_1(x_i, x_j)$ the rescaled distance of $dis(x_i, x_j)$ in the interval $[0, 1]$, and $d_2(x_i, x_j)$ the rescaled distance of $dis(x_i, x_j)$ in the interval $[0, 2]$ and $d_2(x_i, x_j) = 2 \times d_1(x_i, x_j)$, where $d_{10}(x_i, x_j) = 0$, $d_{11}(x_i, x_j) = 1$, $d_{20}(x_i, x_j) = 0$, and $d_{22}(x_i, x_j) = 2$. Let's also define with $[L_1, U_1]$ and $[L_2, U_2]$ the the similarity



(a)



(b)

Fig. 3: Trustworthiness (a), and Continuity (b) for SDD with *deg* 1, 7 and 15 .

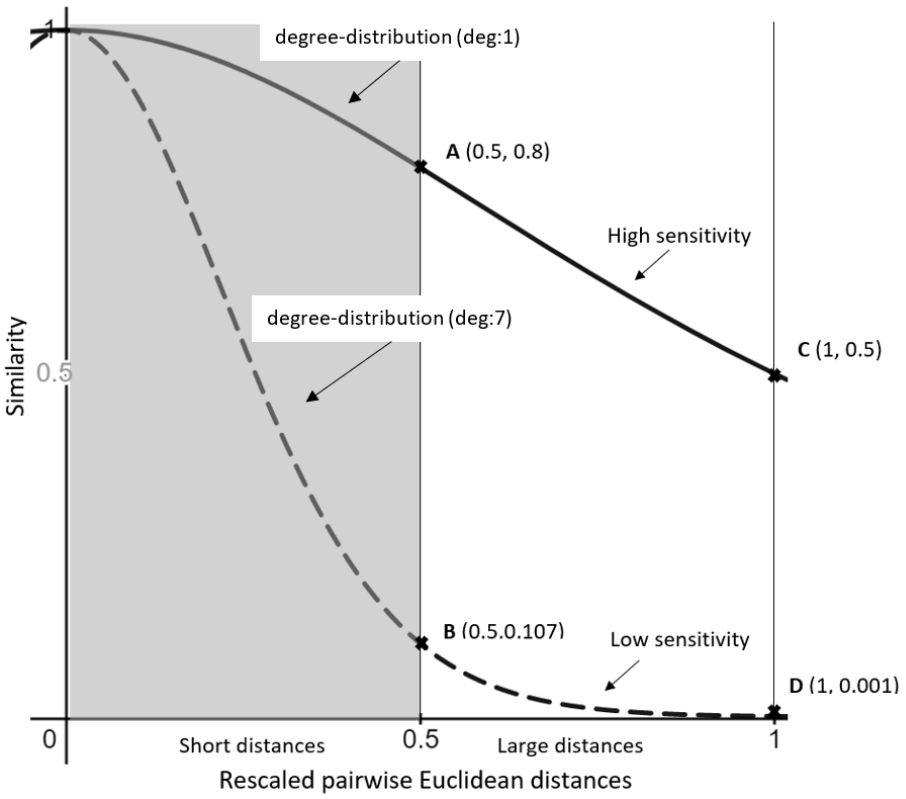


Fig. 4: Two degree-distributions and the sensitivity to large pairwise distances.

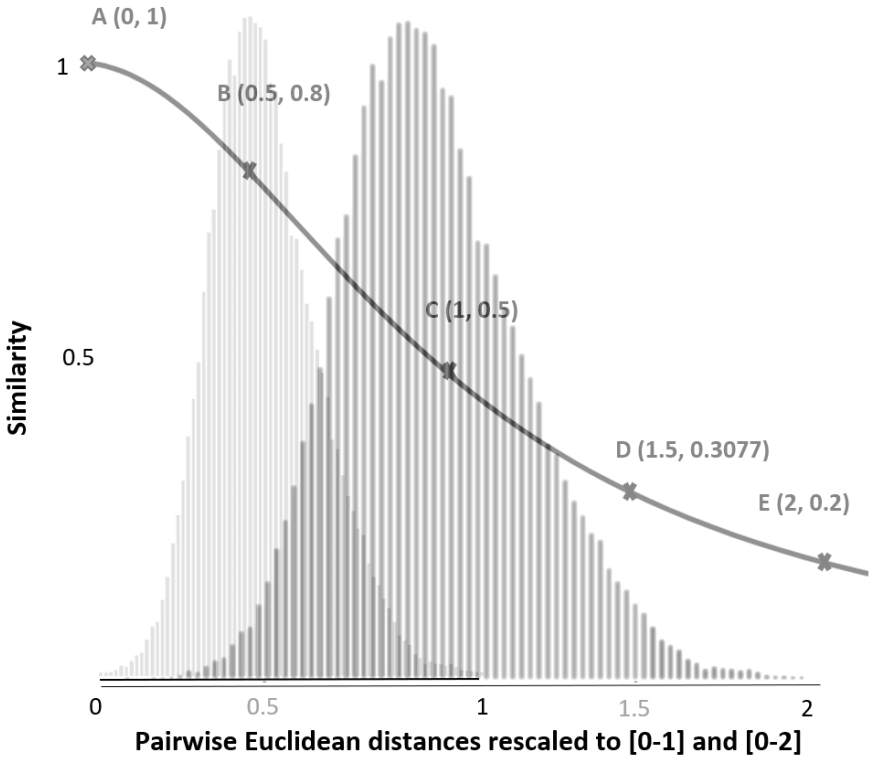


Fig. 5: Degree-distribution ($deg = 1$) in the pairwise distances re-scaled in [0, 2].

ranges of $[0,1]$ and $[0,2]$, respectively.

$$L_1 = \frac{(1+d_{10}(x_i,x_j))^{-1}}{S_1} = \frac{(1+0)^{-1}}{S_1} = \frac{1}{S_1} = \frac{1}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} \quad (10)$$

where $S_1 = \sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}$

$$U_1 = \frac{(1+d_{11}(x_i,x_j))^{-1}}{S_1} = \frac{(1+1)^{-1}}{S_1} = \frac{1}{2(S_1)} = \frac{1}{2\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} \quad (11)$$

$$L_2 = \frac{(1+d_{20}(x_i,x_j))^{-1}}{S_2} = \frac{(1+0)^{-1}}{S_2} = \frac{1}{S_2} = \frac{1}{\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \quad (12)$$

where $S_2 = \sum\limits_{k\neq l}(1+d_2(x_k,x_l))^{-1} = \sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}$

$$U_2 = \frac{(1+d_{22}(x_i,x_j))^{-1}}{S_2} = \frac{(1+2)^{-1}}{S_2} = \frac{1}{3(S_2)} = \frac{1}{3\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \quad (13)$$

Based on Eqs. (10) and (11), the interval of similarity is:
$[L_1,U_1] = [\sum\limits_{k\neq l}(1+d_1(x_k,x_l)), \frac{1}{2\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}}]$,
and based on Eqs. (12) and (13) the interval of similarity is:
$[L_2,U_2] = [\frac{1}{\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}}, \frac{1}{3\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}}]$. As such, the length of the interval is

$$L_1 - U_1 = \frac{1}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} - \frac{1}{2\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} = \frac{1}{2\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} \quad (14)$$

$$L_2 - U_2 = \frac{1}{\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{3\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} = \frac{2}{3\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \quad (15)$$

To proof that $[L_2,U_2]$ is wider than $[L_1,U_1]$, then based on Eqs. (14) and (15) it has to be proven that

$$\frac{2}{3\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} > \frac{1}{2\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} \Longrightarrow$$

$$\frac{2}{3\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{2\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} > 0$$

$$
\begin{aligned}
(L_2 - U_2) - (L_1 - U_1) = \\
&= \frac{2}{3\sum_{k\neq l}\frac{1}{(1+2d_1)}} - \frac{1}{2\sum_{k\neq l}\frac{1}{(1+d_1)}} \\
&= \frac{4\sum_{k\neq l}\frac{1}{(1+d_1)} - 3\Big(\sum_{k\neq l}\frac{1}{(1+2d_1(x_k,x_l))}\Big)}{6\sum_{k\neq l}\frac{1}{(1+2d_1(x_k,x_l))}\sum_{k\neq l}\frac{1}{(1+d_1(x_k,x_l))}} \\
&= \frac{\sum_{k\neq l}\frac{4}{(1+d_1(x_k,x_l))} - \Big(\frac{3}{(1+2d_1)}\Big)}{6\sum_{k\neq l}\frac{1}{(1+2d_1)}\frac{1}{(1+d_1)}} \\
&= \frac{\sum_{k\neq l}\frac{4((1+2d_1))-3((1+d_1))}{(1+d_1)(1+2d_1(x_k,x_l))}}{6\sum_{k\neq l}\frac{1}{(1+2d_1(x_k,x_l))(1+d_1(x_k,x_l))}} \\
&= \frac{\sum_{k\neq l}\frac{1}{(1+d_1(x_k,x_l))(1+2d_1(x_k,x_l))}\sum_{k\neq l}4((1+2d_1(x_k,x_l))) - 3((1+d_1(x_k,x_l)))}{6\sum_{k\neq l}\frac{1}{(1+2d_1(x_k,x_l))(1+d_1(x_k,x_l))}} \\
&= \frac{\sum_{k\neq l}4(1+2d_1(x_k,x_l)) - 3((1+d_1(x_k,x_l)))}{6} \\
&= \frac{\sum_{k\neq l}(1+5d_1(x_k,x_l))}{6}
\end{aligned}
$$

Since $\frac{\sum_{k\neq l}(1+5d_1(x_k,x_l))}{6} \geq 0$, then the similarity range provided by pairwise distances rescaled in the range [0, 2] is wider than the similarity range provided by pairwise distances rescaled in the range [0, 1]. ■

However, a further question arises: are the similarity ranges of both short and large distances expanded the same as the range of pairwise distances increases? To examine the question, consider short distances $d_{1\,short} \in [L_1, H_1]$, $d_{1\,large} \in]H_1, U_1]$, $d_{2\,short} \in [L_2, H_2]$, and $d_{1\,large} \in]H_2, U_2]$ as in Fig. 6.

To evaluate whether short and large distances have been affected mainly by increasing the range of pairwise distances, has defined and proven in Proposition 2.

Proposition 2 By increasing the range of rescaled pairwise distances, the similarity range of short distances increases more than the similarity range of large distances.

Proof

Since L_1, U_1, L_2, U_2 have been calculated in the Proposition 1, then let calculates the H_1 and H_2 which are the middle samples of $[L_1, U_1]$ and $[L_2, U_2]$, respectively.

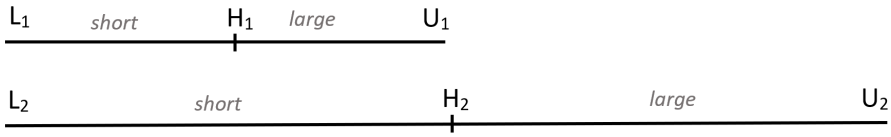


Fig. 6: Two data segments.

$$H_1 = \frac{(1 + \frac{(d_{10}(x_i,x_j))+(d_{11}(x_i,x_j))}{2})^{-1}}{S_1} = \frac{2}{3(S_1)} = \frac{2}{3\sum\limits_{k\neq l}(1 + d_1(x_k, x_l))^{-1}} \quad (16)$$

$$H_2 = \frac{(1 + \frac{(d_{20}(x_i,x_j))+(d_{22}(x_i,x_j))}{2})^{-1}}{S_2} = \frac{1}{2(S_2)} = \frac{1}{2\sum\limits_{k\neq l}(1 + 2d_1(x_k, x_l))^{-1}} \quad (17)$$

Finally,
$\Big[L_1, H_1\Big] = \Big[\frac{1}{\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}}, \frac{2}{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}}\Big]$, and
$\Big[L_2, H_2\Big] = \Big[\frac{1}{\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}}, \frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}}\Big]$
Then,

$$L_1 - H_1 = \frac{1}{\sum\limits_{k\neq l}(1 + d_1(x_k, x_l))^{-1}} - \frac{2}{3\sum\limits_{k\neq l}(1 + d_1(x_k, x_l))^{-1}} = \frac{1}{3\sum\limits_{k\neq l}(1 + d_1(x_k, x_l))^{-1}} \quad (18)$$

$$L_2 - H_2 = \frac{1}{\sum\limits_{k\neq l}(1 + 2d_1(x_k, x_l))^{-1}} - \frac{1}{2\sum\limits_{k\neq l}(1 + 2d_1(x_k, x_l))^{-1}} = \frac{1}{2\sum\limits_{k\neq l}(1 + 2d_1(x_k, x_l))^{-1}} \quad (19)$$

To proof that $[L_2, H_2]$ is wider than $[L_1, H_1]$, then based on Eqs. (18) and (19) have to be checked if $\frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} > \frac{1}{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}}$ which is equivalent with $\frac{1}{2\sum\limits_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{3\sum\limits_{k\neq l}(1+d_1(x_k,x_l))^{-1}} > 0$

$$
\begin{aligned}
&(L_2 - H_2) - (L_1 - H_1) \\
&\qquad = \frac{1}{2\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{3\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}} \\
&\qquad = \frac{3\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1} - 2(\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1})}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&\qquad = \frac{3\frac{1}{\sum_{k\neq l}(1+d_1(x_k,x_l))} - 2\frac{1}{\sum_{k\neq l}(1+2d_1(x_k,x_l))}}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&\qquad = \frac{\frac{3\sum_{k\neq l}(1+2d_1(x_k,x_l)) - 2\sum_{k\neq l}(1+d_1(x_k,x_l))}{\sum_{k\neq l}(1+d_1(x_k,x_l))\sum_{k\neq l}(d_1(x_k,x_l))}}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&\qquad = \frac{\frac{\sum_{k\neq l}(1) + 4\sum_{k\neq l}(d_1(x_k,x_l))}{\sum_{k\neq l}(1+d_1(x_k,x_l))\sum_{k\neq l}(1+2d_1(x_k,x_l))}}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&\qquad = \frac{\sum_{k\neq l}(1) + 4\sum_{k\neq l}(d_1(x_k,x_l))\frac{1}{\sum_{k\neq l}(1+d_1(x_k,x_l))\sum_{k\neq l}(1+2d_1(x_k,x_l))}}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&\qquad = \frac{\sum_{k\neq l}(1) + 4\sum_{k\neq l}(d_1(x_k,x_l))}{6}
\end{aligned}
$$

Also, based on Eqs. (16) and (11) $\Big[H_1, U_1\Big] = \Big[\frac{2}{3\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}}, \frac{1}{2\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}}\Big]$, and based on Eqs. (17) and (13) $\Big[H_2, U_2\Big] = \Big[\frac{1}{2\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}}, \frac{1}{3\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}}\Big]$ then,

$$
H_1 - U_1 = \frac{2}{3\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}} - \frac{1}{2\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}} = \frac{1}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}} \tag{20}
$$

$$
H_2 - U_2 = \frac{1}{2\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{3\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} = \frac{1}{6\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \tag{21}
$$

To proof that $[H_2, U_2]$ is wider than $[H_1, U_1]$, then based on Eqs. (20) and (21) has to be checked if $\frac{1}{2\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} > \frac{1}{3\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}}$ which is equivalent with $\frac{1}{6\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}} > 0$

$$
\begin{aligned}
(H_2 - U_2) - (H_1 - U_1) = \\
&= \frac{1}{6\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} - \frac{1}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}} \\
&= \frac{\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1} - (\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1})}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\frac{1}{\sum_{k\neq l}(1+d_1(x_k,x_l))} - \frac{1}{\sum_{k\neq l}(1+2d_1(x_k,x_l))}}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\frac{\sum_{k\neq l}(1+2d_1(x_k,x_l)) - \sum_{k\neq l}(1+d_1(x_k,x_l))}{\sum_{k\neq l}(1+d_1(x_k,x_l))\sum_{k\neq l}(d_1(x_k,x_l))}}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\sum_{k\neq l}(d_1(x_k,x_l))\frac{1}{\sum_{k\neq l}(1+d_1(x_k,x_l))\sum_{k\neq l}(1+2d_1(x_k,x_l))}}{6\sum_{k\neq l}(1+d_1(x_k,x_l))^{-1}\sum_{k\neq l}(1+2d_1(x_k,x_l))^{-1}} \\
&= \frac{\sum_{k\neq l}(d_1(x_k,x_l))}{6}
\end{aligned}
$$

As proved above, when increasing the pairwise distances ranges from [0, 1] to [0, 2], the similarity range of short distances is increased by $\frac{\sum_{k\neq l}(1)+4\sum_{k\neq l}(d_1(x_k,x_l))}{6}$ and similarity range of large distances increased by $\frac{\sum_{k\neq l}(d_1(x_k,x_l))}{6}$. Having a wider interval of similarity means a small change in distance derives a bigger change in similarity. As such, data samples $A_1, A_2, ..., A_m$ have pairwise distances in the interval $[L_1 = 0, H_1 = 0.5]$ if pairwise distances are rescaled in range $[0, 1]$ and in and in interval $[L_0 = 0, H_2 = 1]$ if pairwise distances are rescaled in range $[0, 2]$. ■

Based on the proof of Proposition 2, the interval of similarity between data samples $A_1, A_2, ..., A_m$ with pairwise distances rescaled in the interval $[L_0 = 0, H_2 = 1]$ is $\frac{\sum_{k\neq l}(1)+4\sum_{k\neq l}(d_1(x_k,x_l))}{6}$ and wider than the interval of similarity between data samples $A_1, A_2, ..., A_m$ with pairwise distances rescaled in the interval $[L_1 = 0, H_1 = 0.5]$. So, if data samples A_1 and A_2 are close and A_1 and A_3 are far away, with pairwise distances $d_1(A_1, A_2) = 0$, $d_1(A_1, A_3) = 5$ and $d_2(A_1, A_2) = 0$, $d_2(A_1, A_3) = 1$, then is more possible that data points A_1, A_2

and A_3 will maintain their structure using pairwise distances rescaled to the interval [0, 2] rather than scaled in the interval [0, 1], due to the huge difference provided between small distances similarities and large distance similarities.

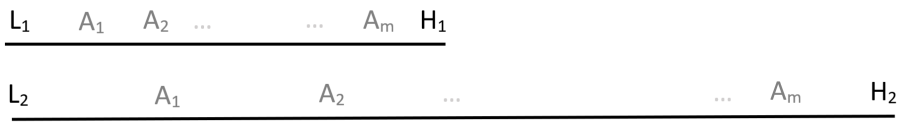


Fig. 7: Data samples $A_1, A_2, ..., A_m$ whose distances is in in the range $[L_1, H_1]$ and $[L_2, H_2]$.

In other words, a wider similarity interval means a better-maintained structure. It is also visible that increasing the pairwise distance range has an impact more on short distances than on large distances. Overall, the increase of the rescaled range has a negative impact on capturing local data structure, which is one of the disadvantages of using degree-distribution with $deg = 1$ in the rescaled distance range [0, 1].

Additionally, based on Proposition 2, if the range of rescaled pairwise distances increases to [0, 2], the global structure destroys. Consequently, rescaling the pairwise distances in the interval [0, 3] or [0, 4] may improve the local structure maintenance. However, it destroys the maintenance of the global data structure because degree-distribution converges to zero when the pairwise distances increase. In conclusion, this research proposes rescaling original data in the interval [0, 2] due to the sensitivity that degree-distribution with $deg = 1$ has in this interval.

As such, it is proposed to use SDD with degree ($deg = 1$) in the rescaled pairwise range in [0, 2]. Using SDD with $deg = 1$ in the rescaled range [0, 2] is named as parameter-free SDD as shown in Algorithm 3, and like SDD, it uses Kullback-Leibler to approximate the degree-distribution in the low dimensional space with the degree-distribution in the high dimensional space:

$$C_1 = \sum_{i \neq j} p_{ij} \log(\frac{p_{ij}}{q_{ij}}) \tag{22}$$

Parameter-free SDD intends to minimize the cost function C_1 as :

$$loss_1 = \min(C_1) \tag{23}$$

where

$$p_{ij} = \frac{(1 + dis(x_i, x_j))^{-1}}{\sum_{k \neq l}(1 + dis(x_k, x_l))^{-1}} \tag{24}$$

Algorithm 1 Parameter-free SDD

Require: Input :

Let be X a dataset with N number of samples and D number of dimensions (features per each sample), calculate matrix DIS of pairwise distance of X and rescale into the range $[0, 2]$, number of iterations H, learning rate η, momentum α, initial low dimensional data $Y^0 = y_1, ..., y_N \in N(0, 10^{-4}I)$, ϵ.

Step 1 :

Compute the high dimensional space similarities (p_{ij}) using (24) and store them in P.

Step 2 :

Compute the low dimensional space similarities (q_{ij}) using (25) and store them in Q.

Step 3 :

Compute the gradient $\frac{\delta C_1}{\delta\ yi}$ where C_1 is defined in (22).

Step 4 :

Minimize the objective function using the Gradient Descent optimisation algorithm: $Y^h = Y^{h-1} + \eta\frac{\delta C_1}{\delta y_i} + \alpha(Y^{h-1} - Y^{h-2})$.

The optimisation algorithm will stop either achieves the maximum number of iterations H or the Kullback-Leibler value is lower than the minimum threshhold ϵ.

Output :

Low dimensional space represenation Y.

$$q_{ij} = \frac{(1 + dis(y_i, y_j))^{-1}}{\sum_{k \neq l}(1 + dis(y_k, y_l))^{-1}} \tag{25}$$

3.2 Complexity Analysis

Parameter-free SDD needs to create two matrixes with $N \times N$ to store distances in both high and low dimensional spaces and another matrix that stores the difference $P - Q$ with $N \times N$. In total, the computational and space complexity of parameter-free SDD is $O(N^2)$ and is significantly less than the computational and space complexity of SDD and MSDD.

3.3 Experimental Results

Parameter-free SDD is an innovative method that takes the highes performance of SDD but saves computational time significantly. As mentioned in Section 3, parameter-free SDD does not require tuning any parameter, and it uses only $deg = 1$, which results to be the $best_{deg}$ due to rescaling the pairwise distances in the range [0, 2].

The performance of parameter-free SDD has been evaluated using Kendall's Tau, Trustworthiness and Continuity and is compared with three different degrees of SDD including 1, 15 and $best_{deg}$. The experiments have all be conducted

on Python with the same number of iterations and optimisation parameters. The dataset considered are Iris, Breast Cancer, Swiss and MNIST. The first dataset considered is the Iris dataset, which contains 150 flowers and 4 attributes for each (length and width of petal and sepal). There are three different types of flowers and fifty samples per type. As shown in Fig. 8, the largest fraction of samplesof Iris data has relatively short and medium distances, in which, SDD approach has proved to perform better than other methods [25]. The Breast Cancer dataset[5]

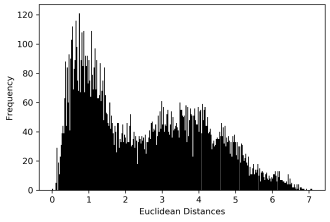


Fig. 8: Euclidean distance distribution of Iris dataset.

with 30 attributes is the second datasets considered. The distance distribution of breast cancer data is shown in Fig. 9, where the majority of samples have relatively short distances, in which SDD is expected to maintain better the data structure [25]. Swiss Roll data is a popular synthetics dataset with 1600 samples

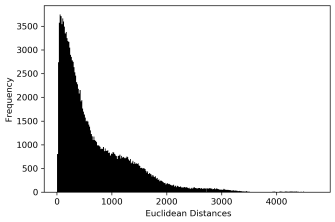
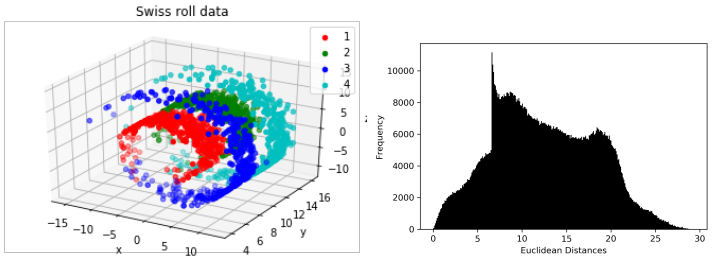


Fig. 9: Euclidean distance distribution of Breast Cancer dataset.

and three attributes and forms a swiss roll shape, as shown in Fig. 10(a). The largest fraction of samples has the pairwise distances in medium range from 5 to 20 as presented in Fig. 10(b).

MNIST with 2500 samples (hand written numbers) and 784 attributes (pixels) is the fourth dataset considered,with distance distribution shown in Fig. 11, dominated by entries with medium, large distances.

[5] Load breast cancer from sklearn, Python.



(a) (b)

Fig. 10: Swiss Roll data (a) and its Euclidean distance distribution (b).

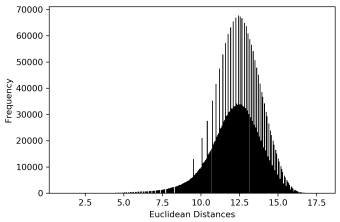
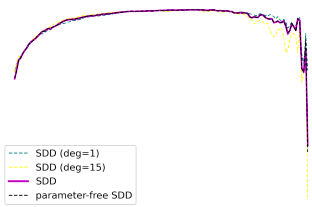


Fig. 11: Euclidean distance distribution of MNIST data.

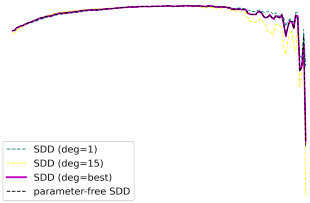
Based on experimental results, SDD itself has proved to be a very good method in capturing data structure on data having majority of samples with relatively short or medium distances, parametric-free SDD appears provenly appropriate to capture local and global data structures due to the high sensitivity of degree-distribution with $deg = 1$ has in the short and large distances into the intervals 0 and 2. As shown in Fig. 5, parameter-free SDD (SDD with $deg = 1$ in [0, 2]) captures slightly the same the local data structure (short distances) compared with SDD ($deg = best$) in [0, 1]. However, parameter-free SDD can capture global data structure better than SDD ($deg = best$), as demonstrated in Figs. 12 and 13.

In addition, the performance of parameter-free SDD has been evaluated using Kendall's Tau, which, as is demonstrated in Table 2, is very similar to SDD ($deg = best$). However, in terms of computational time, parameter-free SDD is significantly less expensive than SDD, as shown in Table 2. Parameter-free SDD takes 0.41, 7.79, 36.93, 183.94 seconds to generates low dimensional data of Iris, Breast Cancer, Swiss Rolls and MNIST data instead of 9.49, 111.33, 552.96 and 2452.12 seconds that SDD takes.

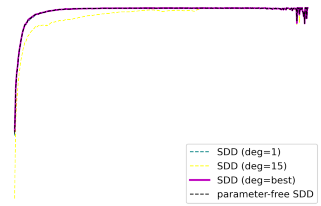
In summary, parameter-free SDD can capture the structure of the data very well due to 1) the long tail of Student-t distribution in capturing the global data structure and 2) to the advantages of re-scaling the pairwise distances DIS in the interval [0, 2], which improves capturing the local data structure. It might happen that because the sensitivity for large distances is small when distance is



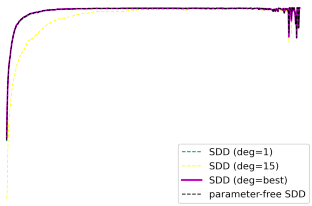
(a) Trustworthiness (Iris)



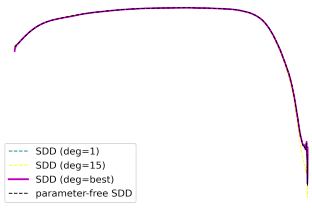
(b) Continuity (Iris)



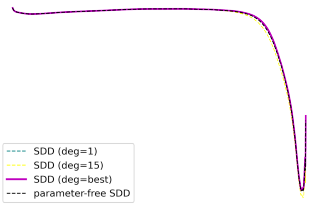
(c) Trustworthiness (Breast Cancer)



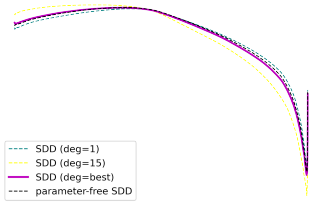
(d) Continuity (Breast Cancer)



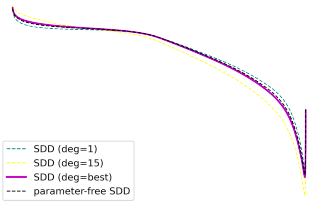
(e) Trustworthiness (Swiss Roll)



(f) Continuity (Swiss Roll)

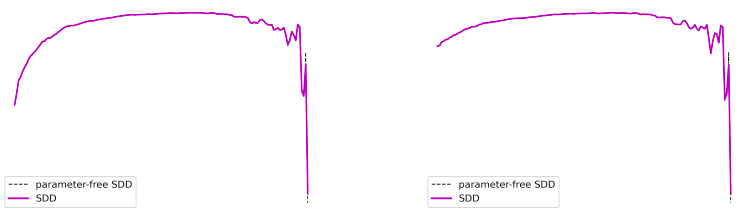


(g) Trustworthiness (MNIST)



(h) Continuity (MNIST)

Fig. 12: Trustworthiness and Continuity for SDD with $deg = 1$, deg_{best} and 15 for re-scaled distances in range [0, 1], and parameter-free SDD.



(a) Trustworthiness (Iris)

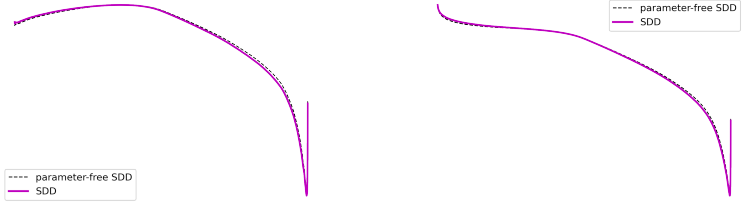
(b) Continuity (Iris)

(c) Trustworthiness (Breast Cancer)

(d) Continuity (Breast Cancer)

(e) Trustworthiness (Swiss Roll)

(f) Continuity (Swiss Roll)



(g) Trustworthiness (MNIST)

(h) Continuity (MNIST)

Fig. 13: Trustworthiness and Continuity for SDD with deg_{best}, and parameter-free SDD.

Table 2: THE PERFORMANCE OF SDD AND PARAMETER-FREE SDD IN TERMS OF KENDALL'S TAU COEFFICIENT AND COMPUTATIONAL TIME

		Datasets			
		Iris	**Breast Cancer**	**Swiss Rolls**	**MNIST**
SDD	deg	6	10	1	1
	τ	0.967328	0.998118	0.914711	0.606525
	$time(sec)$	9.49	111.33	552.96	2454.12
parameter-free SDD	τ	0.967339	0.998086	0.914578	0.607947
	$time(sec)$	0.41	7.79	36.93	183.94

increased into [0, 2], they may be negligible from the cost function, making the global structure not as good as degree-dsitribution with $deg = 1$. However, the global structure captured by parameter-free SDD is better than the global data structure captured by SDD with $deg = best$ in pairwise distance ranges in [0, 1].

Furthermore, as shown in Table 2, parameter-free SDD is capable to maintain the same data structure as SDD but in significantly less computational time. Note that SDD has been demonstrated to be best methods in maintaining data structure [25], and parameter-free SDD is capable to maintain the same performance in terms of maintaining the same data structure with spending significantly less computational time. This makes parameter-free SDD a very useful approach for visualization high dimensional data having different data type including nonlinear data and with complex manifold representations.

4 Parametric SDD

PCA is one of the most famous parametric dimensionality reduction methods; however, it is a linear method that favours preserving global data structure at the expense of neglecting local data structure. Also, RBMs, a parametric method, favours capturing the global data structure, and it is a more complicated method due to the number of parameters required to tune. RBMs were proposed [27] to make t-SNE a parametric method. Parametric t-SNE is intended to maintain the local data structure, whereas RBMs maximises the data covariance, and that it means it captures the global data structure. And as a result, parametric t-SNE is ineffective in preserving well-separated clusters it contrasts with RBM's objective function that maximises the variance. To deal with this problem, using supervised learning with neural networks was proposed by [28] to make t-SNE a parametric method. A neural network was trained to learn the two-dimensional data generated by a given dimensionality reduction method (t-SNE). This approach has been very effective in using high dimensional data in data structure capturing, scalability, and simplicity of genericity [27]. It minimises the distance

between the two-dimensional data generated by the standard method and the two-dimensional data generated by the trained neural network.

Although SDD is an excellent structure capturing method, it is still not feasible for the out-of-the sample data. It was inspired by the effectiveness of using neural networks (ANN) to learn how to transform the original high-dimensional data to its corresponding low-dimensional data, i.e., embedding generated by SDD. The logical flowchart of the project for training an NN to learn an embedding is shown in Fig. 14. A successfully trained NN can be used to embed any new data, and therefore, this makes SDD a parametric method. In other words, the trained NN provides an explicit model to estimate the implicant data embedding formed by SDD.

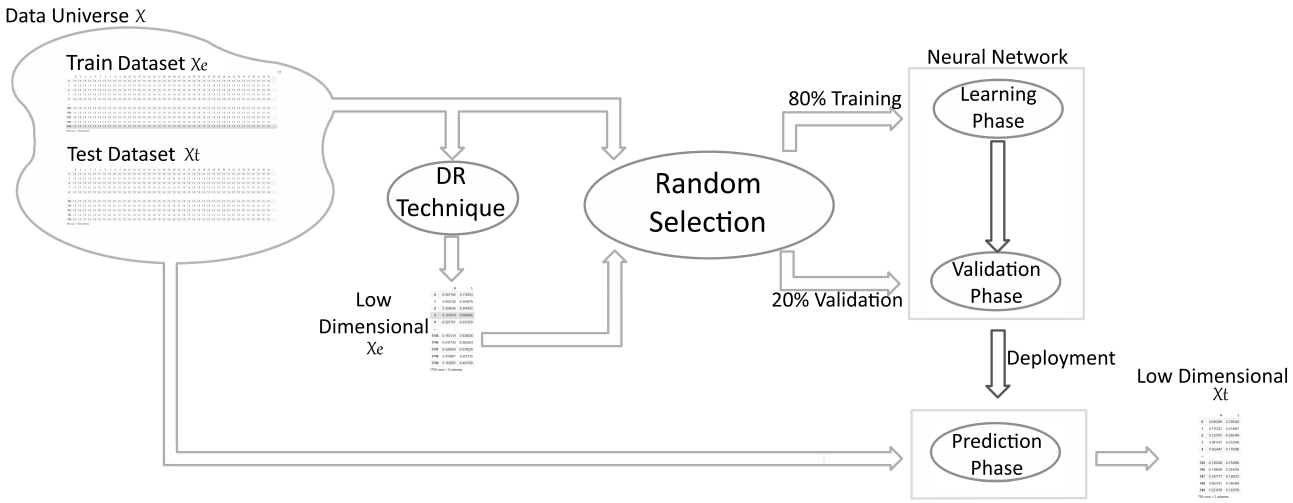


Fig. 14: Framework of Learning Projection.

To testify the neural network, the dataset X will be split into training set X_e and testing set X_t, where X_e is used to train the neural network, and X_t to test the neural network. The neural network uses X_e and the two-dimensional data generated by dimensionality reduction techniques $DR(X_e)$. Each datasets used is randomly split into 80% training the network and 20% to validate the network. After the ANN has achieved satisfactory results in terms of classification accuracy, the network is used to embedd will predict X_t.

The architecture of the ANN employed is shown in Fig. 15, and it has three fully connected hidden layers with 256, 512 and 256 units, respectively, using the ReLU activation function as shown in Fig. 11. The last layer has two elements and uses a sigmoid function to encode the two-dimensional projection, scaled to the interval [0,1]. The optimisation method used for the training is the ADAM optimiser, a derivate of the stochastic gradient descent. The training lasts up

to 80 epochs and stops when there is no significant change to the loss in three successive epochs. The loss function used is the Mean Squared Error (MSE), and can be expressed as following:

$$MSE = \frac{1}{N}\sum_{i=1}^{N} \|DR(X_e)_i - NN(X_e)_i\|^2 \tag{26}$$

, where $DR(X_e)_i$, $NN(X_e)_i$ are the ground truth of two dimensional data generated by the dimensionality reduction method (DR) and the network (NN), respectively.

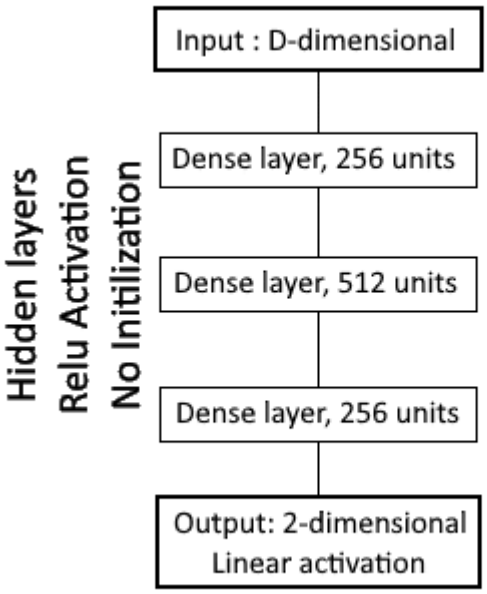


Fig. 15: Network Architecture employed.

4.1 Experiments and Results Discussion

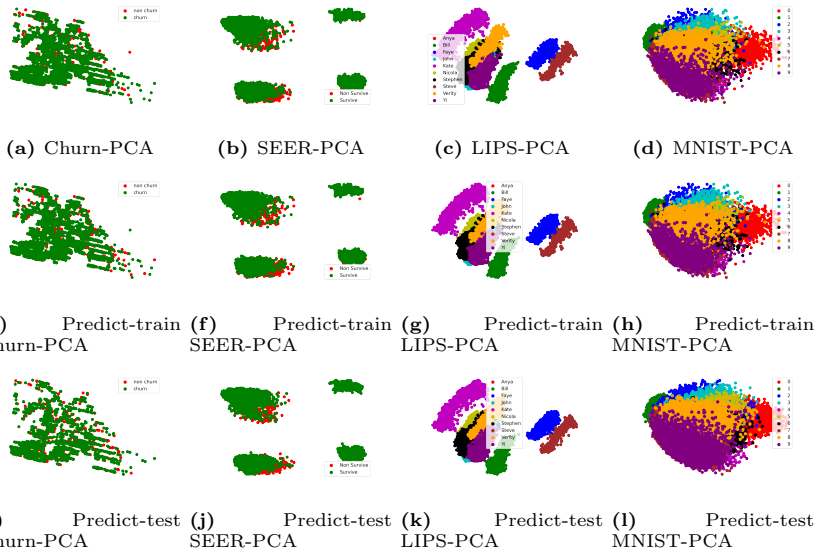
The dimensionality reduction techniques considered are PCA, MDS, Isomap, LE, t-SNE, UMAP, and MSDD, run with the same iteration using Python. The performance of these methods depends on some tunning parameters, so we have tuned parameters to check their performance estimated by Kendall's Tau correlation coefficient (τ). Dataset considered in this section are non-temporal data, such as Synthetic data (MNIST), Medical data (SEER Breast Cancer),

Customer data (Churn data), Image processing data (AVletters (LIPS)). The same architecture of CNN has been applied to two-dimensional data generated by PCA, Isomap, t-SNE, Umap and parameter-free SDD. PCA is a parametric method, and there exists a parametric t-SNE; however, for comparison reasons, the same Neural Network architecture has been applied to all methods. The MNIST data with 60,000 gray images from 0 to 9 with 28×28 pixels will be flattened into 784 dimensional record.[6] The SEER Breast Cancer data contains a totally of 291,760 incidences registered in the US from 1974 to 2017. The original data sets need to be pre-processed and transformed to a target data set for analysis. The most crucial task in the data pre-processing process is to identify any data quality issues and further adopt appropriate strategies to address them accordingly. The data pre-processing process was very time-consuming, and it has eventually led to a resultant target data set with 260,000 incidences and 961 variables. The variable survival that represents if a patient survived has been considered the target variable since this analysis aims to identify crucial factors that potentially affect the survival of a breast cancer patient. The Churn data contains 9786 customers that are described by 2 variables, a customer is churn or no. The AVletters database (LIPS Reading data) consists of three repetitions by each of 10 talkers, five male (two with moustaches) and five female, of the isolated letters A-Z, a total of 780 utterances. Each talker was requested to begin and end each letter utterance with their mouth in the closed position. No head restraint was used, but talkers were provided with a close-up view of their mouths and asked not to move out of the frame. The full face images were further cropped to a region of 80×60 pixels after manually locating the centre of the mouth in the middle frame of each utterance.

[6] MNIST data from Keras.

Table 3: THE PERFORMANCE OF METHODS (ROWS) IN DATASETS (COLUMNS) IN TERMS OF KENDALL'S TAU COEFFICIENT

DR algorithm	Churn	SEER	LIPS	MNIST
PCA	**0.9691**	**0.5154**	**0.7391**	**0.3533**
PCA (predict train)	0.9688	0.5152	0.7401	0.3489
PCA (predict test)	0.9693	0.5194	0.7417	0.3401
t-SNE	**0.7466**	**0.2729**	**0.3824**	**0.2395**
t-SNE (predict train)	0.8509	0.2764	0.3825	0.2413
t-SNE (predict test)	0.8562	0.2796	0.3799	0.2432
Isomap	**0.9043**	**0.4778**	**0.7399**	**0.4021**
Isomap (predict train)	0.9062	0.4783	0.7399	0.4018
Isomap (predict test)	0.91081	0.4845	0.7416	0.3984
SDD	**0.9723**	**0.7279**	**0.7948**	**0.6199**
SDD (predict train)	0.9711	0.7258	0.7838	0.6130
SDD (predict test)	0.9715	0.7247	0.7849	0.6059

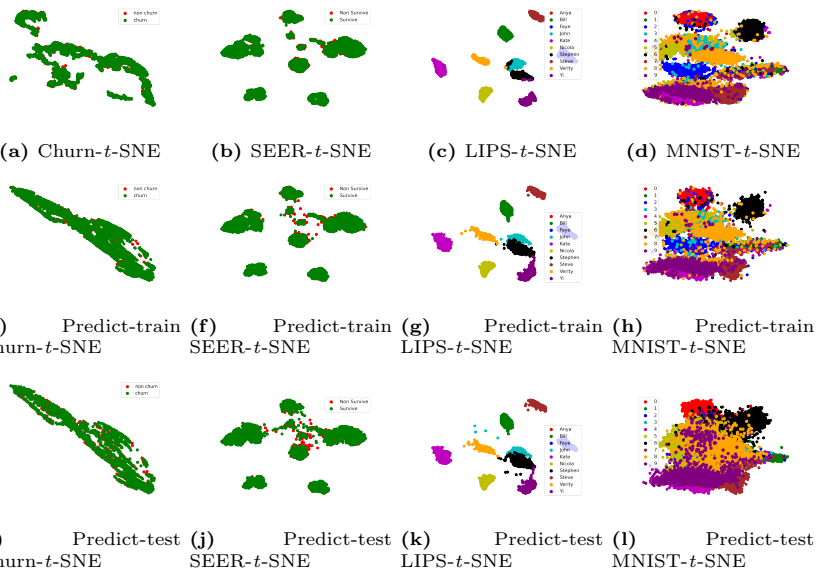


(a) Churn-PCA **(b)** SEER-PCA **(c)** LIPS-PCA **(d)** MNIST-PCA

(e) Predict-train Churn-PCA **(f)** Predict-train SEER-PCA **(g)** Predict-train LIPS-PCA **(h)** Predict-train MNIST-PCA

(i) Predict-test Churn-PCA **(j)** Predict-test SEER-PCA **(k)** Predict-test LIPS-PCA **(l)** Predict-test MNIST-PCA

Fig. 16: The visualisation of the two-dimensional representation of the Churn (23 attributes), SEER Breast Cancer (960 attributes), Lips (4800 attributes), MNIST (784 attributes) generated by PCA.

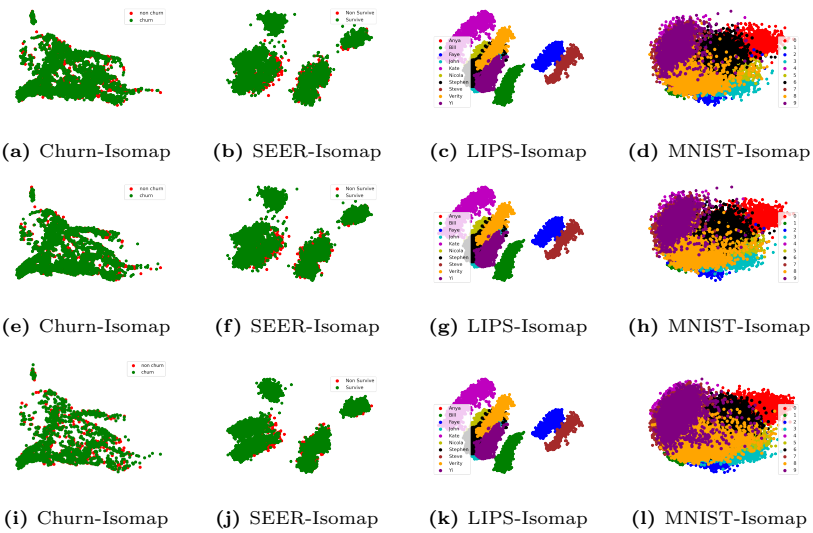


(a) Churn-t-SNE **(b)** SEER-t-SNE **(c)** LIPS-t-SNE **(d)** MNIST-t-SNE

(e) Predict-train Churn-t-SNE **(f)** Predict-train SEER-t-SNE **(g)** Predict-train LIPS-t-SNE **(h)** Predict-train MNIST-t-SNE

(i) Predict-test Churn-t-SNE **(j)** Predict-test SEER-t-SNE **(k)** Predict-test LIPS-t-SNE **(l)** Predict-test MNIST-t-SNE

Fig. 17: The visualisation of the two-dimensional representation of the Churn (23 attributes), SEER Breast Cancer (960 attributes), Lips (4800 attributes), MNIST (784 attributes) generated by t-SNE.

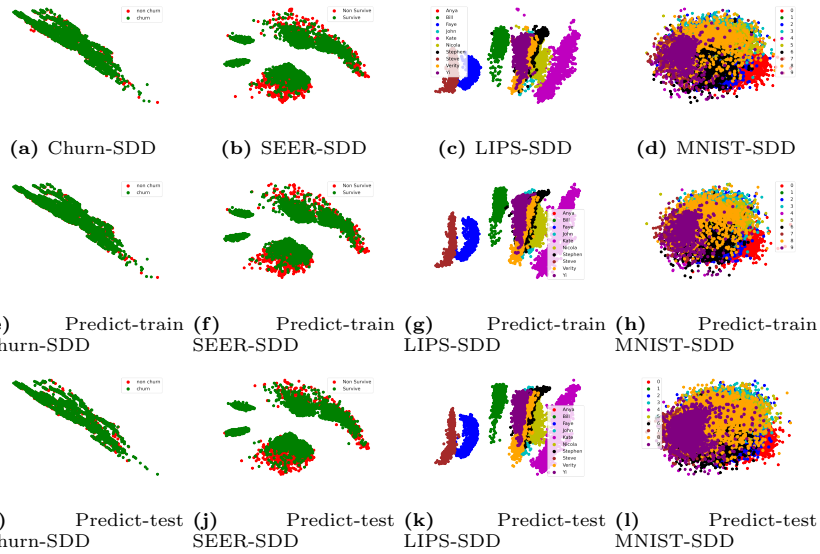


(a) Churn-Isomap **(b)** SEER-Isomap **(c)** LIPS-Isomap **(d)** MNIST-Isomap

(e) Churn-Isomap **(f)** SEER-Isomap **(g)** LIPS-Isomap **(h)** MNIST-Isomap

(i) Churn-Isomap **(j)** SEER-Isomap **(k)** LIPS-Isomap **(l)** MNIST-Isomap

Fig. 18: The visualisation of the two-dimensional representation of the Churn (23 attributes), SEER Breast Cancer (960 attributes), Lips (4800 attributes), MNIST (784 attributes) generated by Isomap.



(a) Churn-SDD **(b)** SEER-SDD **(c)** LIPS-SDD **(d)** MNIST-SDD

(e) Predict-train Churn-SDD **(f)** Predict-train SEER-SDD **(g)** Predict-train LIPS-SDD **(h)** Predict-train MNIST-SDD

(i) Predict-test Churn-SDD **(j)** Predict-test SEER-SDD **(k)** Predict-test LIPS-SDD **(l)** Predict-test MNIST-SDD

Fig. 19: The visualisation of the two-dimensional representation of the Churn (23 attributes), SEER Breast Cancer (960 attributes), Lips (4800 attributes), MNIST (784 attributes) generated by SDD.

Note that the ratio of training/testing samples in % is different in different datasets as shown in Table 4. Training/testing samples (%) is 70%/30%, 50%/50%, 50%/50%, 25%/75% in Churn, SEER Breast Cancer, LIPS and MNIST datasets, respectively.

Table 4: DATASETS (ROWS) AND TRAINING, TESTING SAMPLES AND DIMENSIONALITY

Datasets	**Training Samples (%)**	**Testing Samples (%)**	**Dimensionality**
Churn	6850 (70 %)	2936 (30%)	23
SEER Breast Cancer	15000 (50 %)	15000 (50 %)	960
LIPS	9280 (50 %)	9280 (50 %)	4800
MNIST	15000 (25 %)	45000 (75 %)	784

Although the ratio of training/testing samples varies with different datasets, it can be said that the employed ANN has been trained very good to embedd the training samples and testing samples. Low dimensional visualisations generated by PCA, t-SNE, Isomap and SDD, and their prediction have been presented in Figs. 16, 17, 18, and 19, respectively. Based on the experimental results shown in Table 3, it can be seen that the method that has been captured the best data structure is SDD to all datasets. As a result, the best structure of testing data has been captured by parametric SDD.

In summary, parametric methods employ ANN to capture the same data structure that their corresponding versions capture. The better the training data structure has been captured, the better the data structure of testing data will be captured.

5 Conclusion

SDD is a nonlinear dimensionality reduction method proposed recently, and it has demonstrated an outstanding performance in structure capturing and saving computational time compared to other states of the art methods. However, SDD still requires tuning the degree of degree-distribution to get the best performance, which may be more costly than other parameter-free methods, including PCA and MDS. This paper proposes a parameter-free SDD that can preserve a pretty similar data structure with SDD but in significantly less computational time. The benefits of parameter-free SDD are by using degree-distribution ($deg = 1$) in high and low dimensional space but re-scaling the pairwise distances of original data in the interval [0, 2] instead of [0, 1]. The performance of parameter-free SDD has been demonstrated experimentally that it achieves the same performance in terms of structure maintenance but in significantly less time than SDD. In terms of structure maintenance, parameter-free SDD outperforms all considered

methods. And in terms of computational time, it outperforms t-SNE, UMAP, Trimap, Isomap, LE, LLE, and MDS. A theoretical proof also supports the excellent performance of parameter-free SDD.

This paper also addresses the problem of out-of-sample data points for SDD by proposing the parametric SDD approach. Parametric SDD proposes using Neural Networks to mimic the two-dimensional data produced by SDD. It has been demonstrated experimentally that parametric SDD maintains the training data structure (where the networks have been trained) and testing data (for unseen data for network).

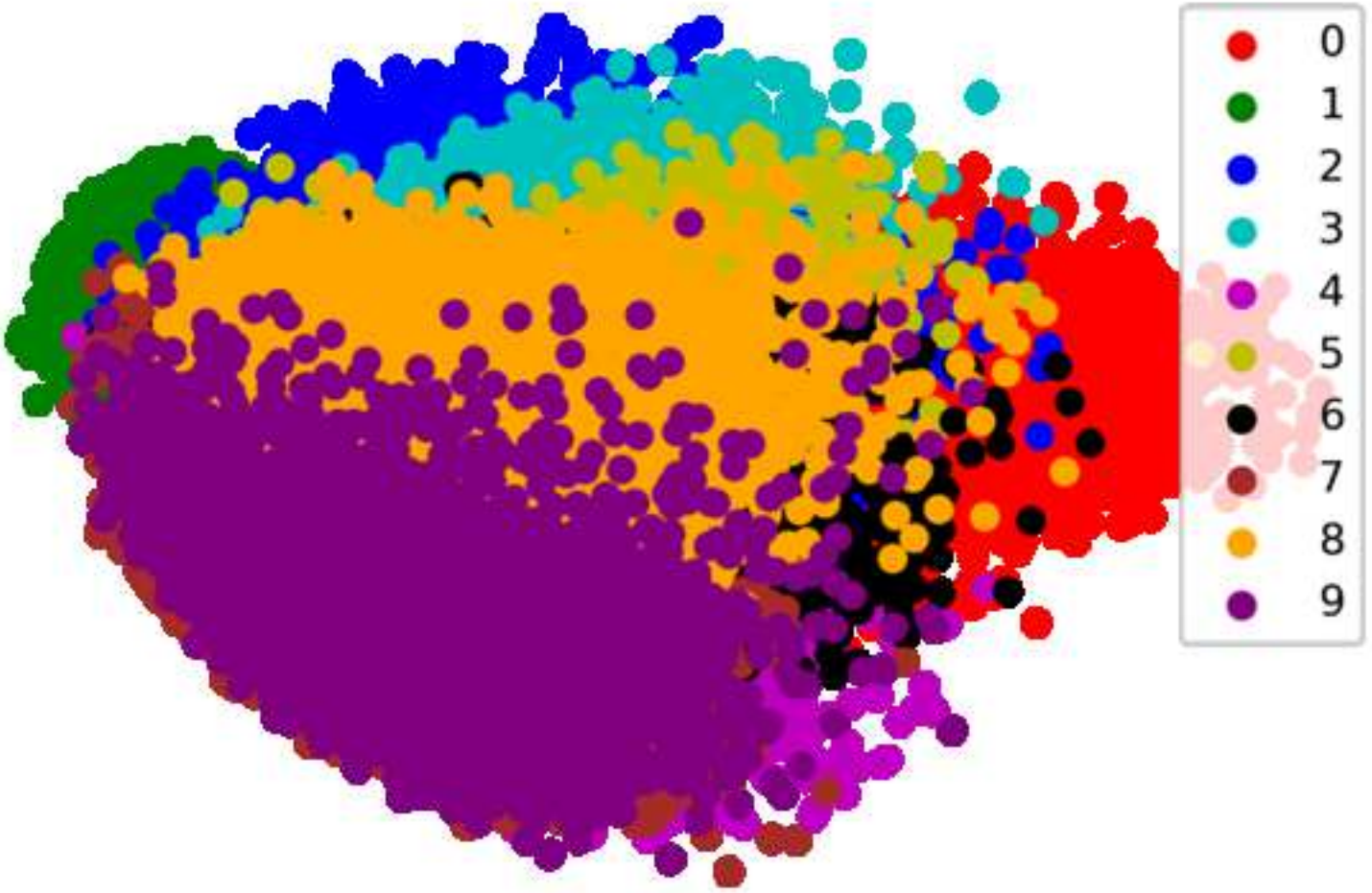
However, parameter-free SDD and parametric SDD over-perform all considered methods when the fraction of data has short and medium distances. For data with a large fraction of large pairwise distances, it remains for further studies.

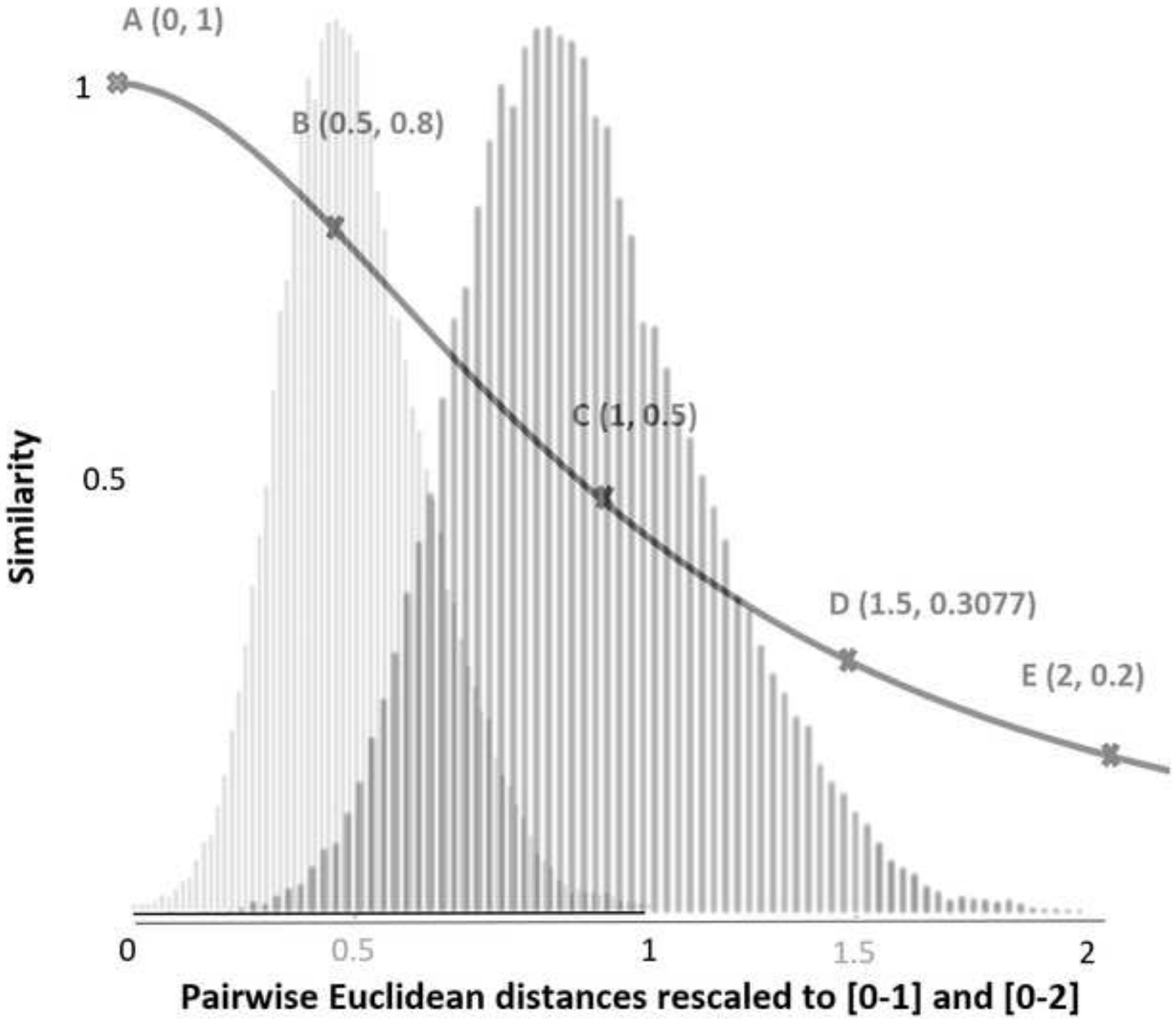
Acknowledgement - This study was supported under a joint scholarship by London South Bank University and Acctive Systems, Ltd. (http://www.acctive.co.uk/).

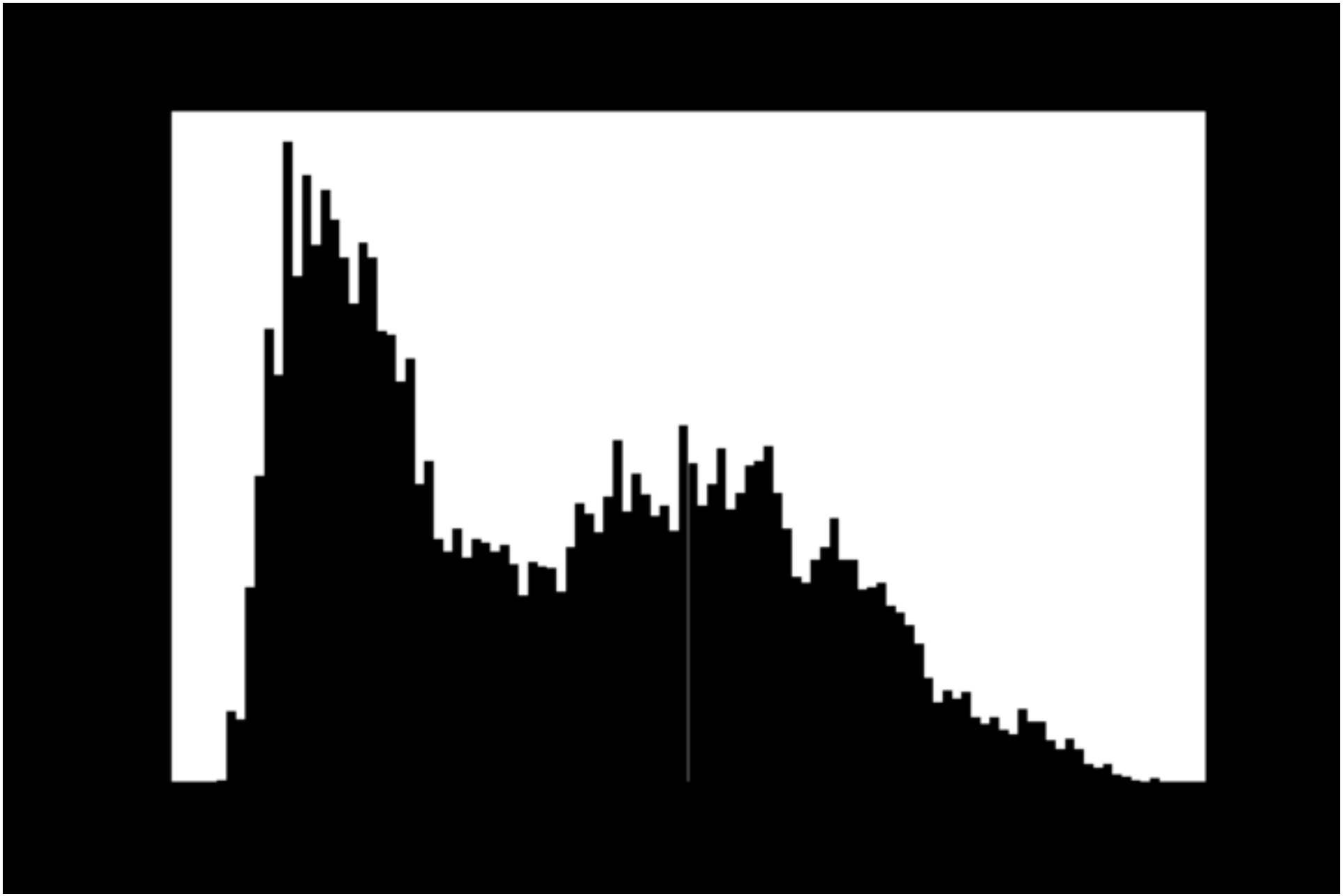
References

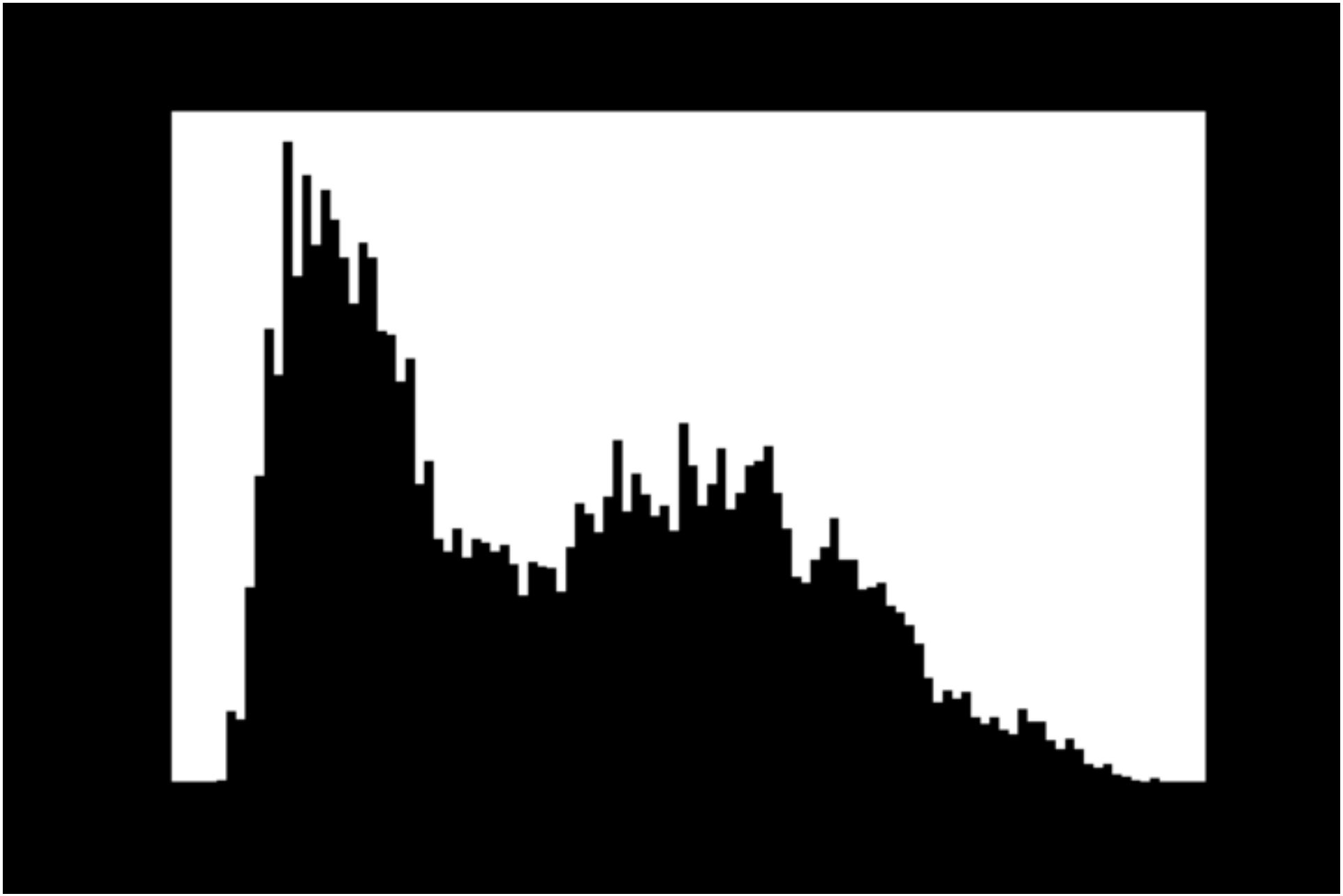
1. Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. Journal of educational psychology, 24(6), p.417.
2. Kruskal, J.B., 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika, 29(1), pp.1-27.
3. Sammon, J.W., 1969. A nonlinear mapping for data structure analysis. IEEE Transactions on computers, 100(5), pp.401-409.
4. Demartines, P. and Hérault, J., 1997. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. IEEE Transactions on neural networks, 8(1), pp.148-154.
5. Bishop, C., Svensén, M. and Williams, C., 1996. GTM: A principled alternative to the self-organizing map. Advances in neural information processing systems, 9.
6. Schölkopf, B., Smola, A. and Müller, K.R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural computation, 10(5), pp.1299-1319.
7. P.J. Werbos.Beyond regression: New tools for prediction and analysis in thebehavioral sciences. PhD thesis, Harvard University, 1974.
8. Tenenbaum, J.B., Silva, V.D. and Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. science, 290(5500), pp.2319-2323.
9. Roweis, S.T. and Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. science, 290(5500), pp.2323-2326.
10. Belkin, M. and Niyogi, P., 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in neural information processing systems, 14.
11. Donoho, D., 2003. Hessian eigenmaps: new tools for nonlinear dimensionality reduction. Proc. National Academy of Science, 100, pp.5591-5596.
12. Scholz, M., Kaplan, F., Guy, C.L., Kopka, J. and Selbig, J., 2005. Non-linear PCA: a missing data approach. Bioinformatics, 21(20), pp.3887-3895.
13. Zhang, Z. and Zha, H., 2004. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM journal on scientific computing, 26(1), pp.313-338.
14. Weinberger, K.Q. and Saul, L.K., 2006. Unsupervised learning of image manifolds by semidefinite programming. International journal of computer vision, 70(1), pp.77-90.

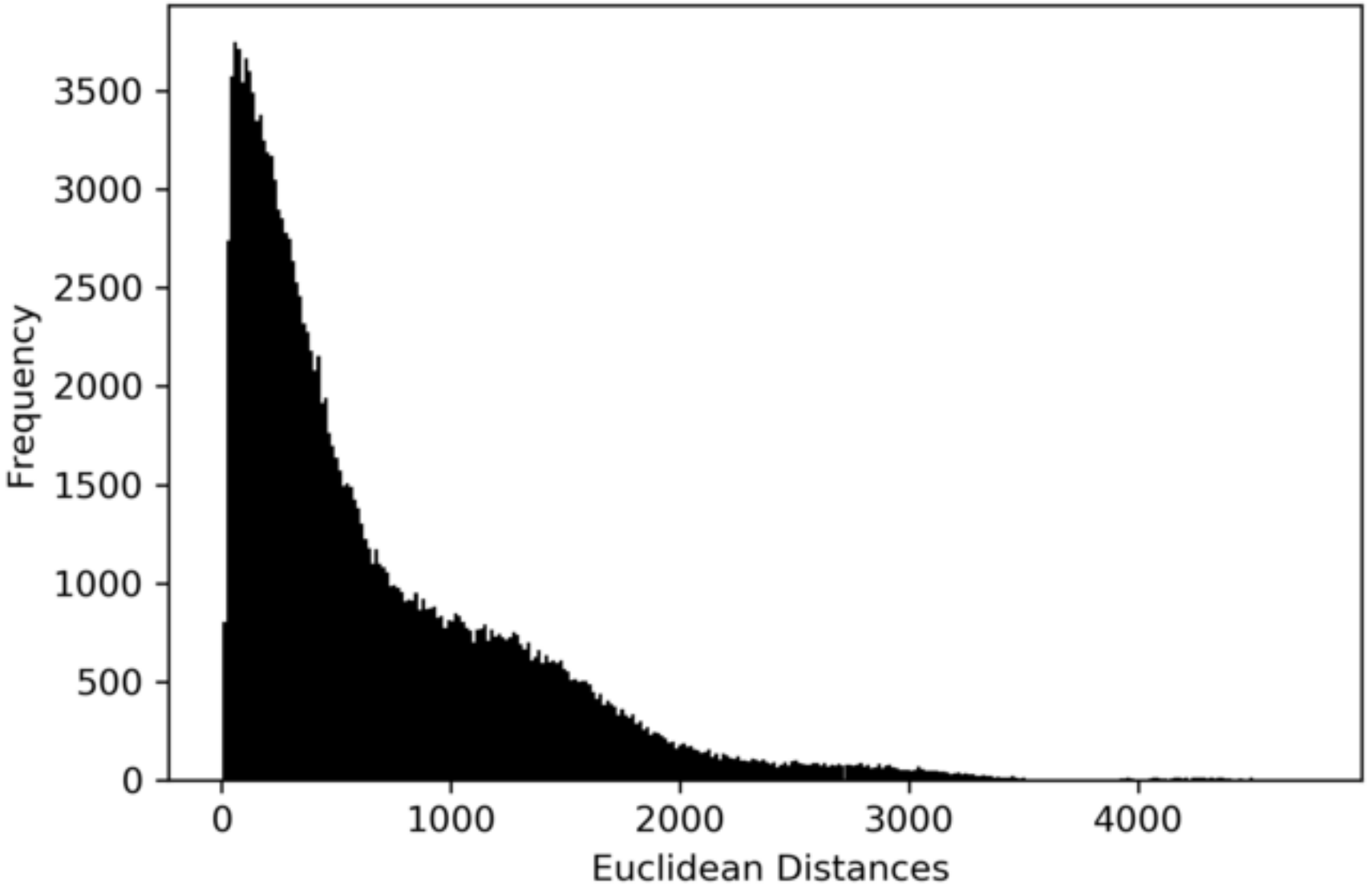
15. Coifman, R.R. and Lafon, S., 2006. Diffusion maps. Applied and computational harmonic analysis, 21(1), pp.5-30.
16. Hinton, G.E. and Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. science, 313(5786), pp.504-507.
17. Zhang, Z. and Wang, J., 2006. MLLE: Modified locally linear embedding using multiple weights. Advances in neural information processing systems, 19
18. Lespinats, S., Fertil, B., Villemain, P. and Hérault, J., 2009. RankVisu: Mapping from the neighborhood network. Neurocomputing, 72(13-15), pp.2964-2978.
19. Van der Maaten, L. and Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research, 9(11).
20. Gashler, M., Ventura, D. and Martinez, T., 2007. Iterative non-linear dimensionality reduction with manifold sculpting. Advances in Neural Information Processing Systems, 20.
21. Lespinats, S., Fertil, B., Villemain, P. and Hérault, J., 2009. RankVisu: Mapping from the neighborhood network. Neurocomputing, 72(13-15), pp.2964-2978.
22. Rosman, G., Bronstein, M.M., Bronstein, A.M. and Kimmel, R., 2010. Nonlinear dimensionality reduction by topologically constrained isometric embedding. International Journal of Computer Vision, 89(1), pp.56-68.
23. McInnes, L., Healy, J. and Melville, J., 2018. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.
24. Amid, E. and Warmuth, M.K., 2018. A more globally accurate dimensionality reduction method using triplets. arXiv preprint arXiv:1803.00854.
25. Hajderanj, L., Chen, D., Grisan, E. and Dudley, S., 2020. Single-and multi-distribution dimensionality reduction approaches for a better data structure capturing. IEEE Access, 8, pp.207141-207155.
26. Wang, J., 2012. Geometric structure of high-dimensional data and dimensionality reduction (Vol. 5). Berlin Heidelberg: Springer.
27. Van Der Maaten, L., 2009, April. Learning a parametric embedding by preserving local structure. In Artificial Intelligence and Statistics (pp. 384-391). PMLR.
28. Espadoto, M., Hirata, N.S.T. and Telea, A.C., 2020. Deep learning multidimensional projections. Information Visualization, 19(3), pp.247-269.

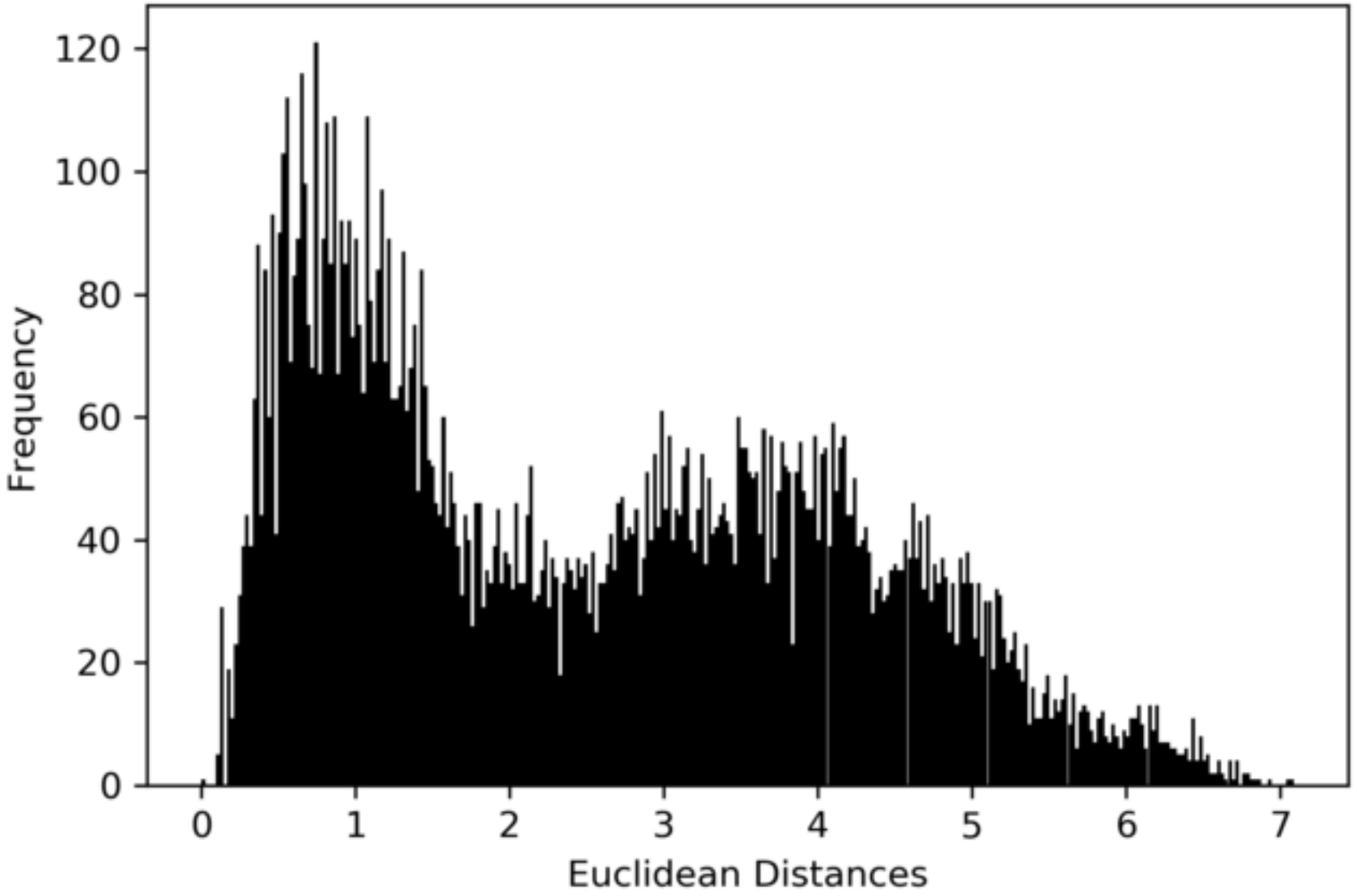


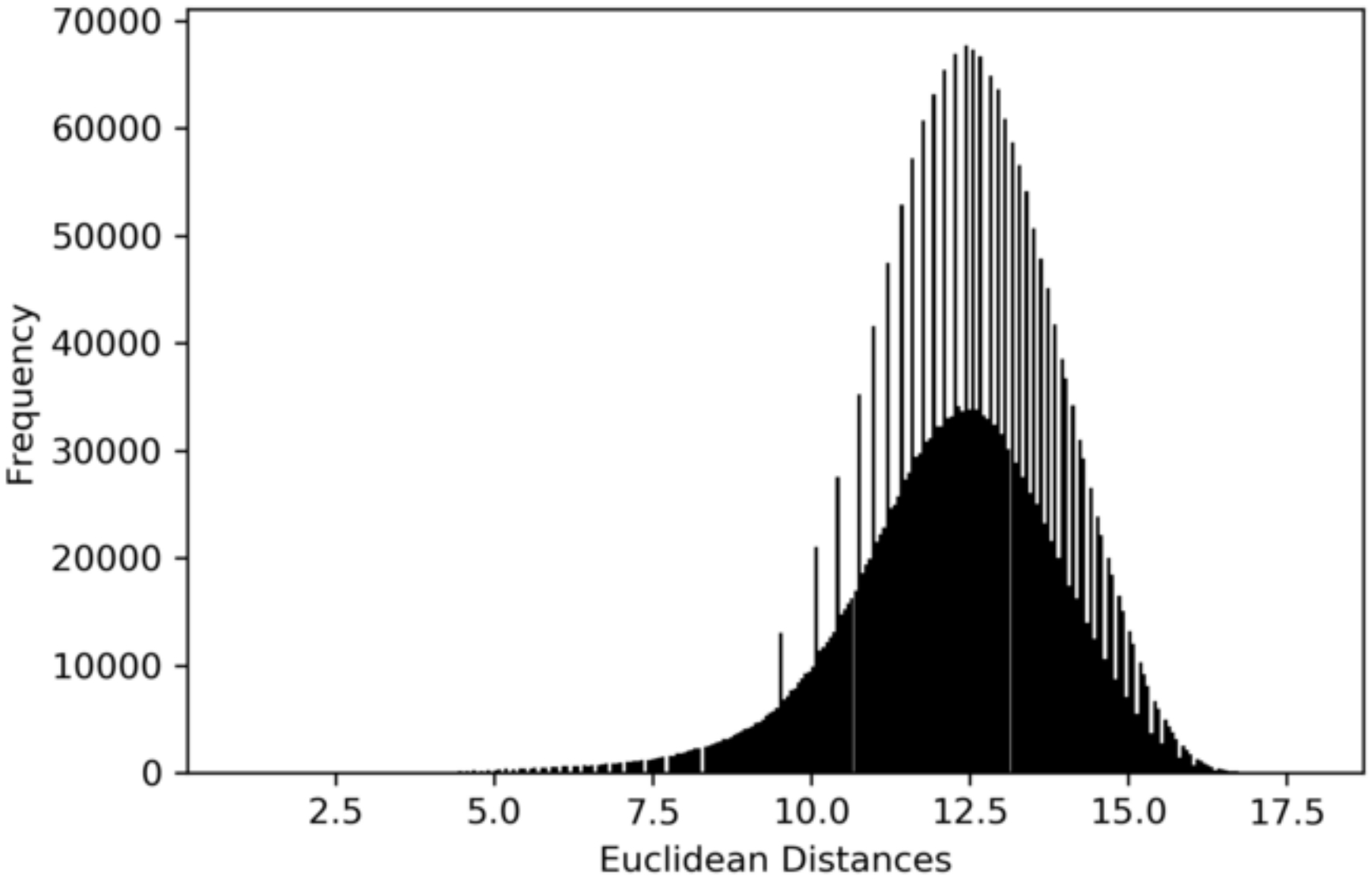


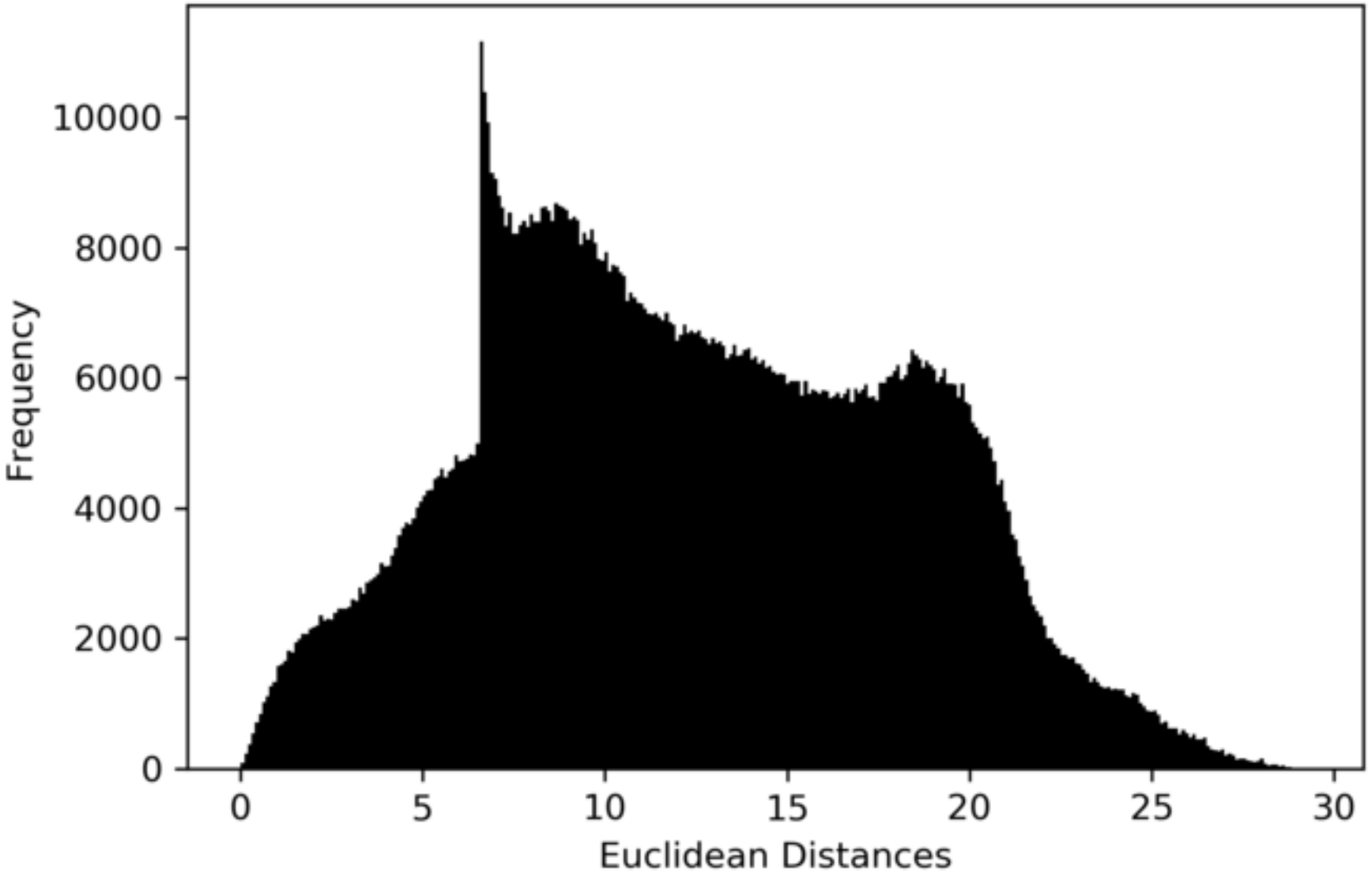


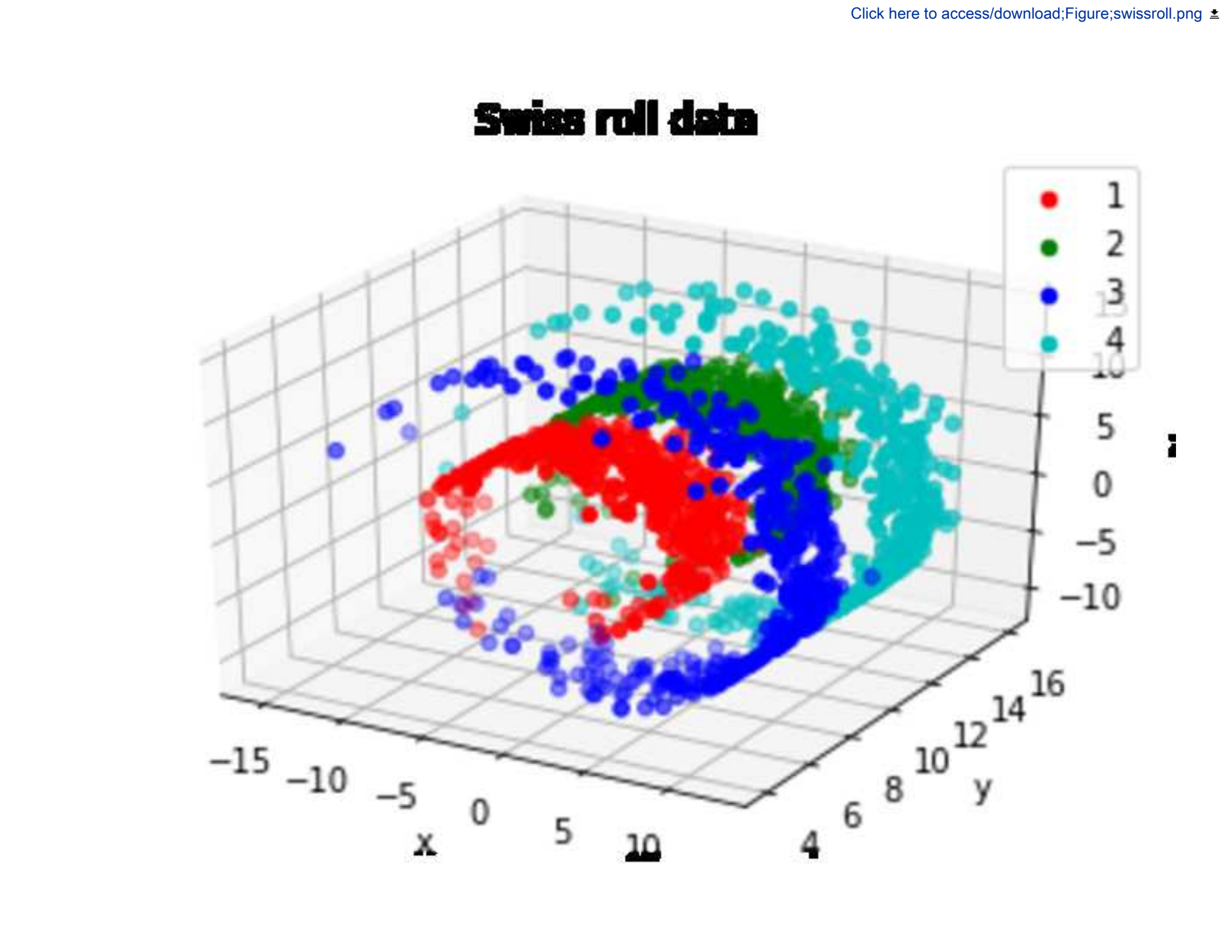


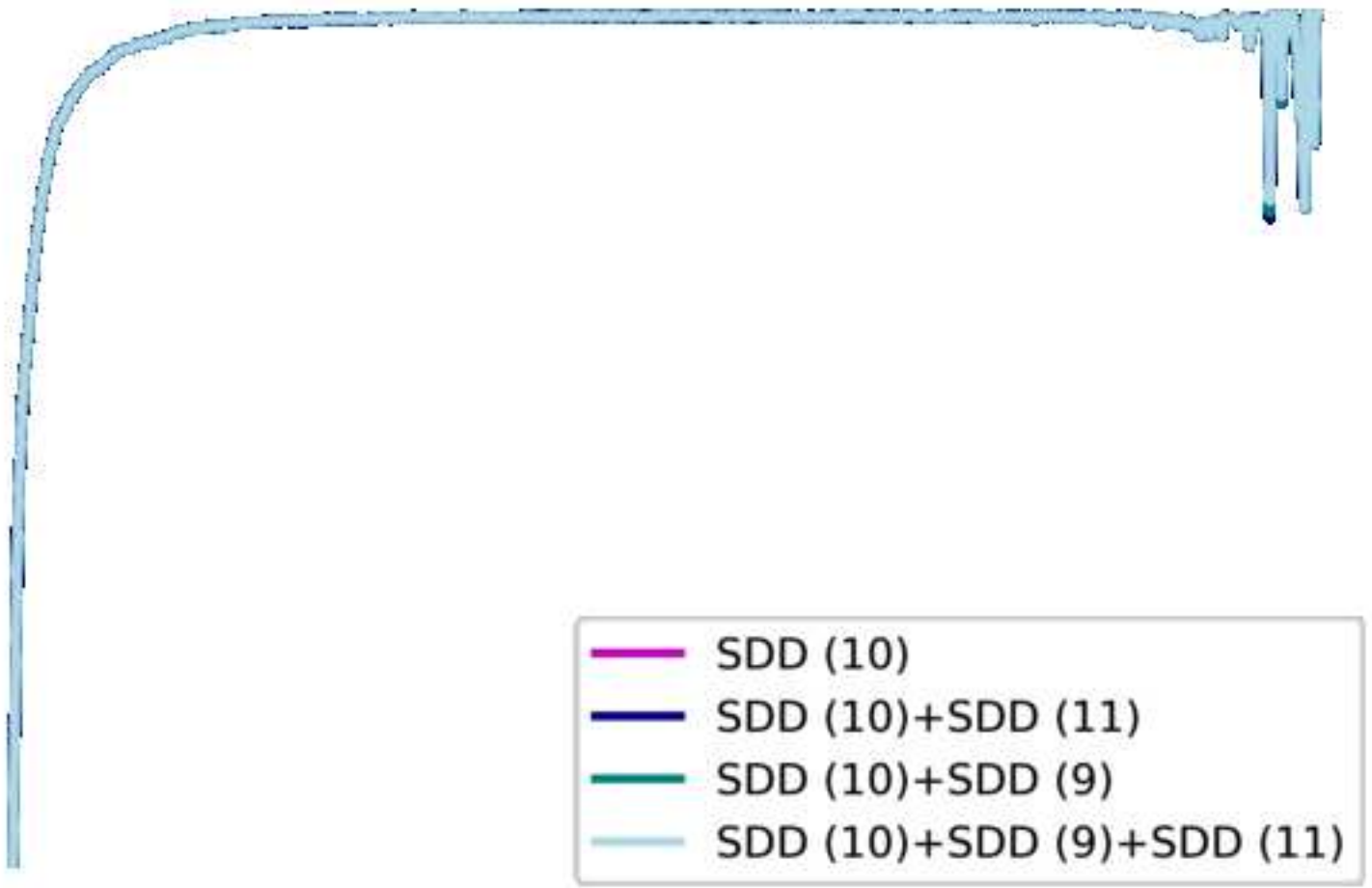


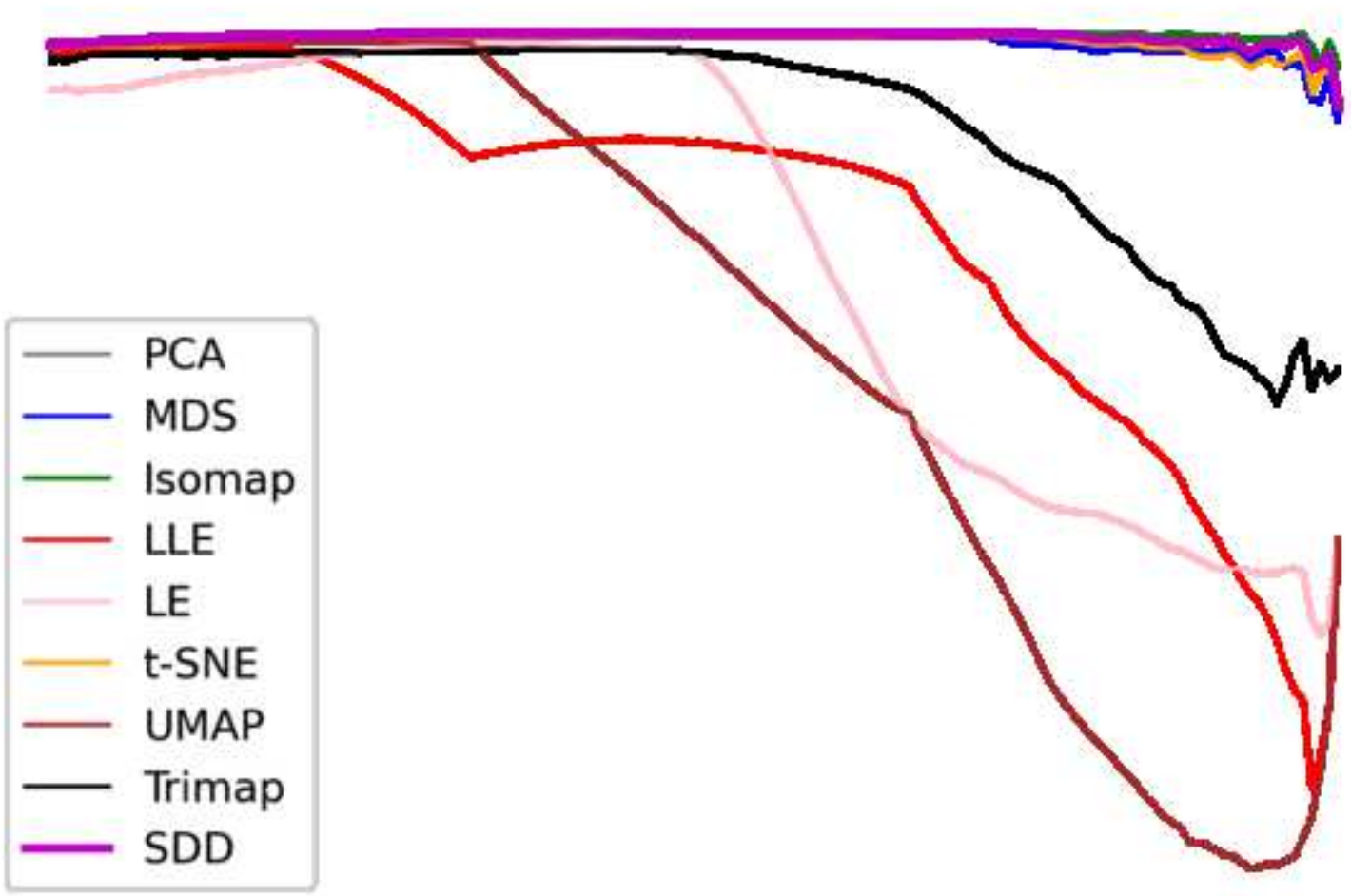


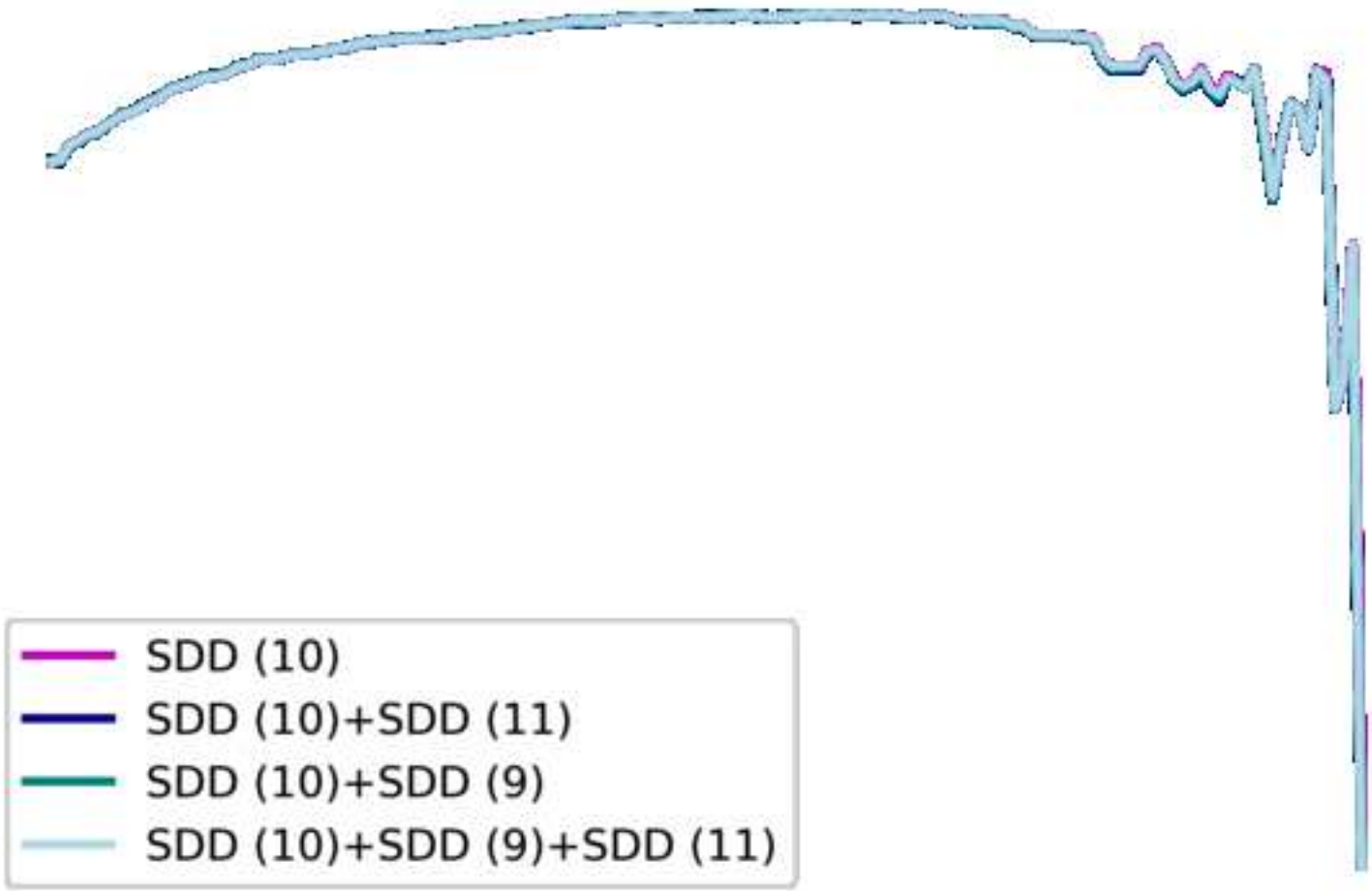


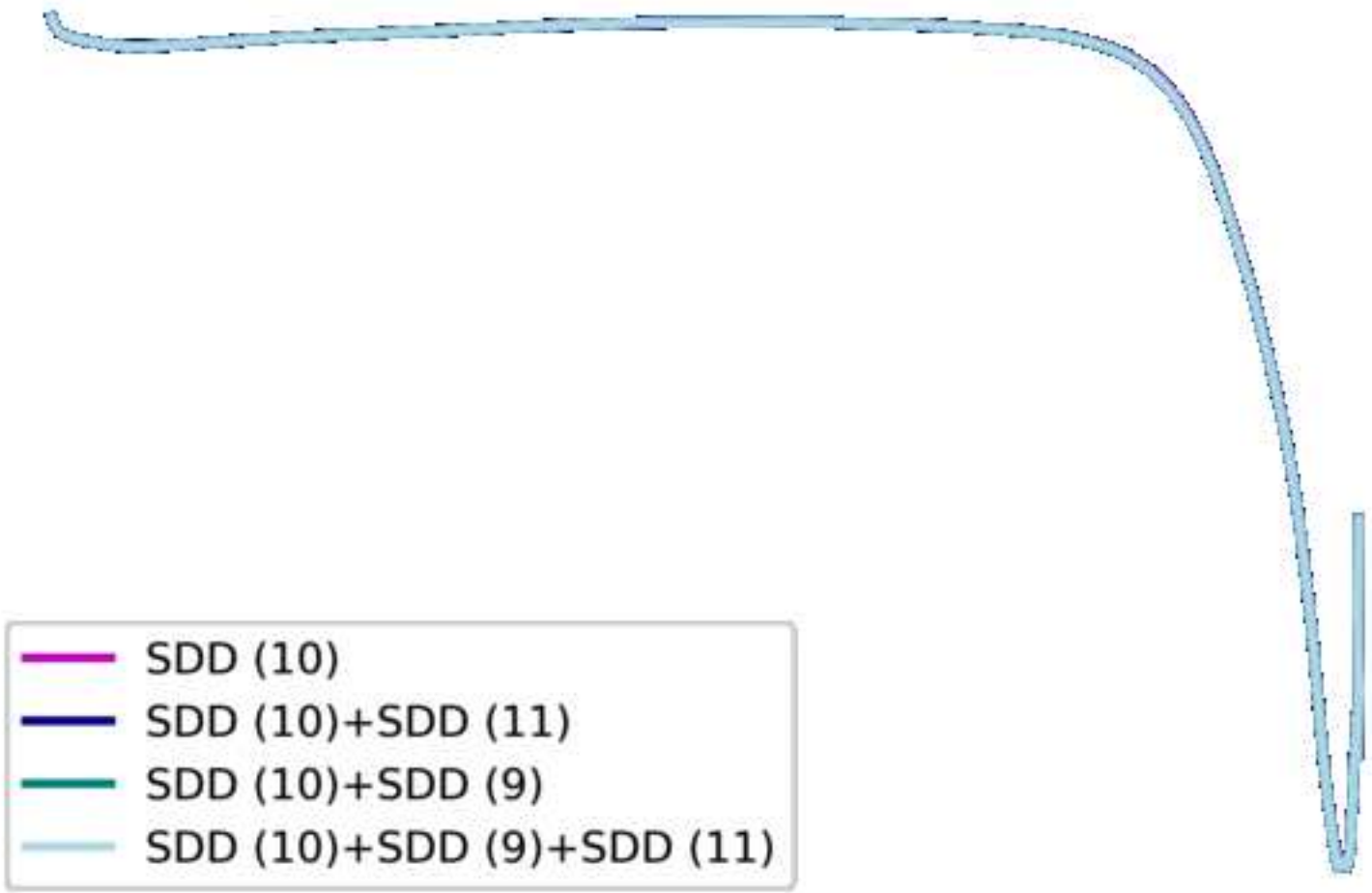


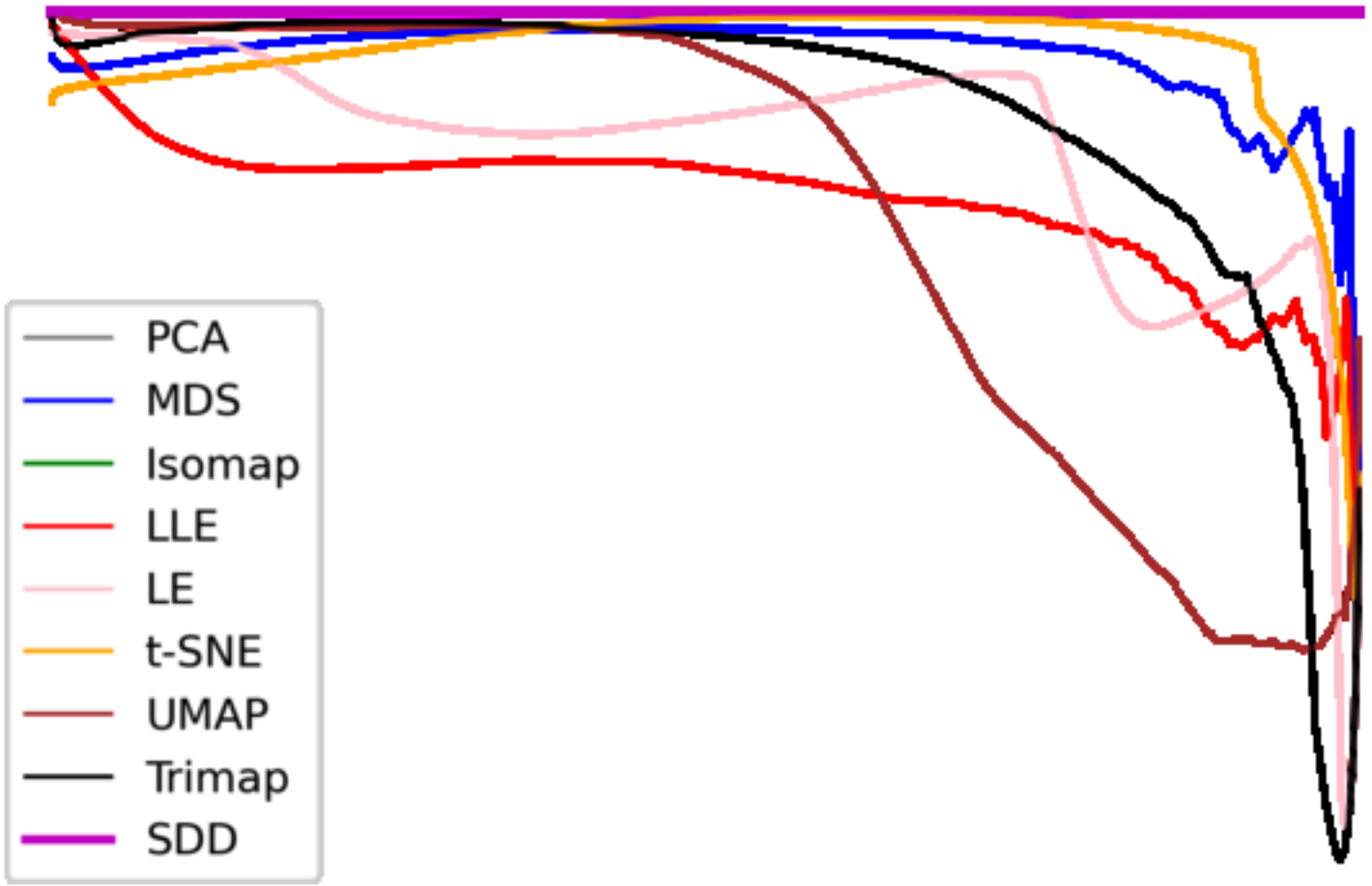


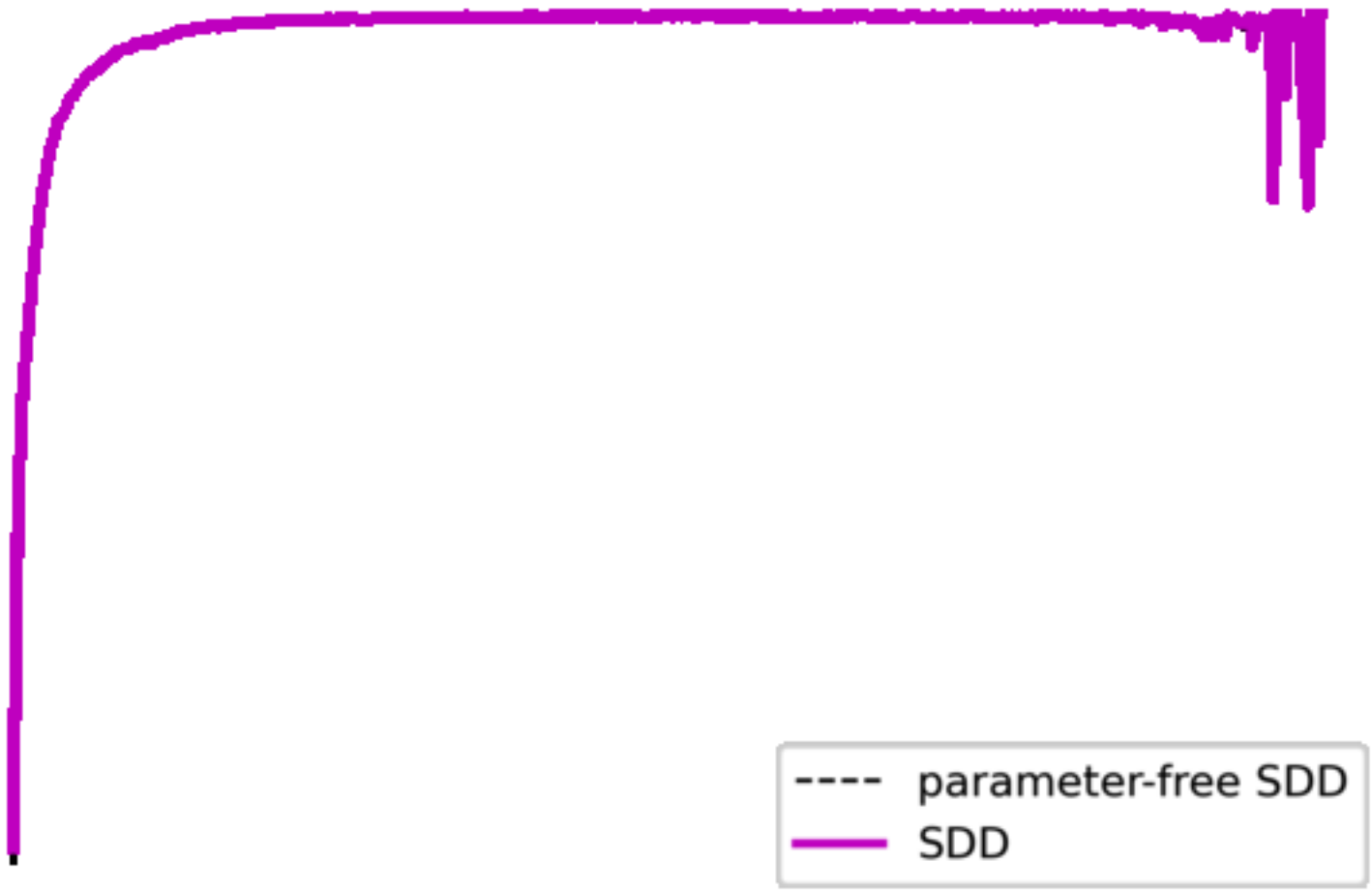


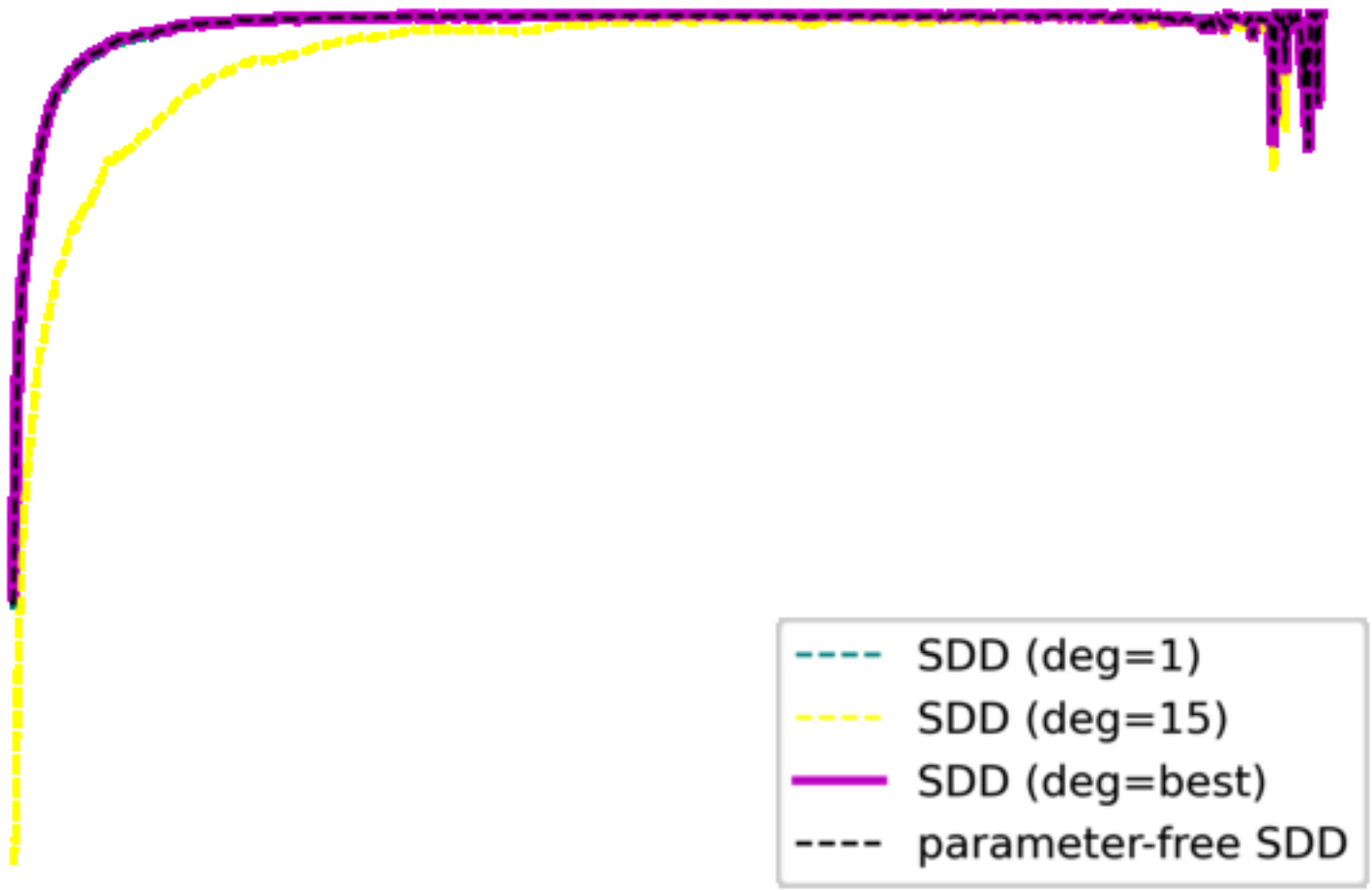


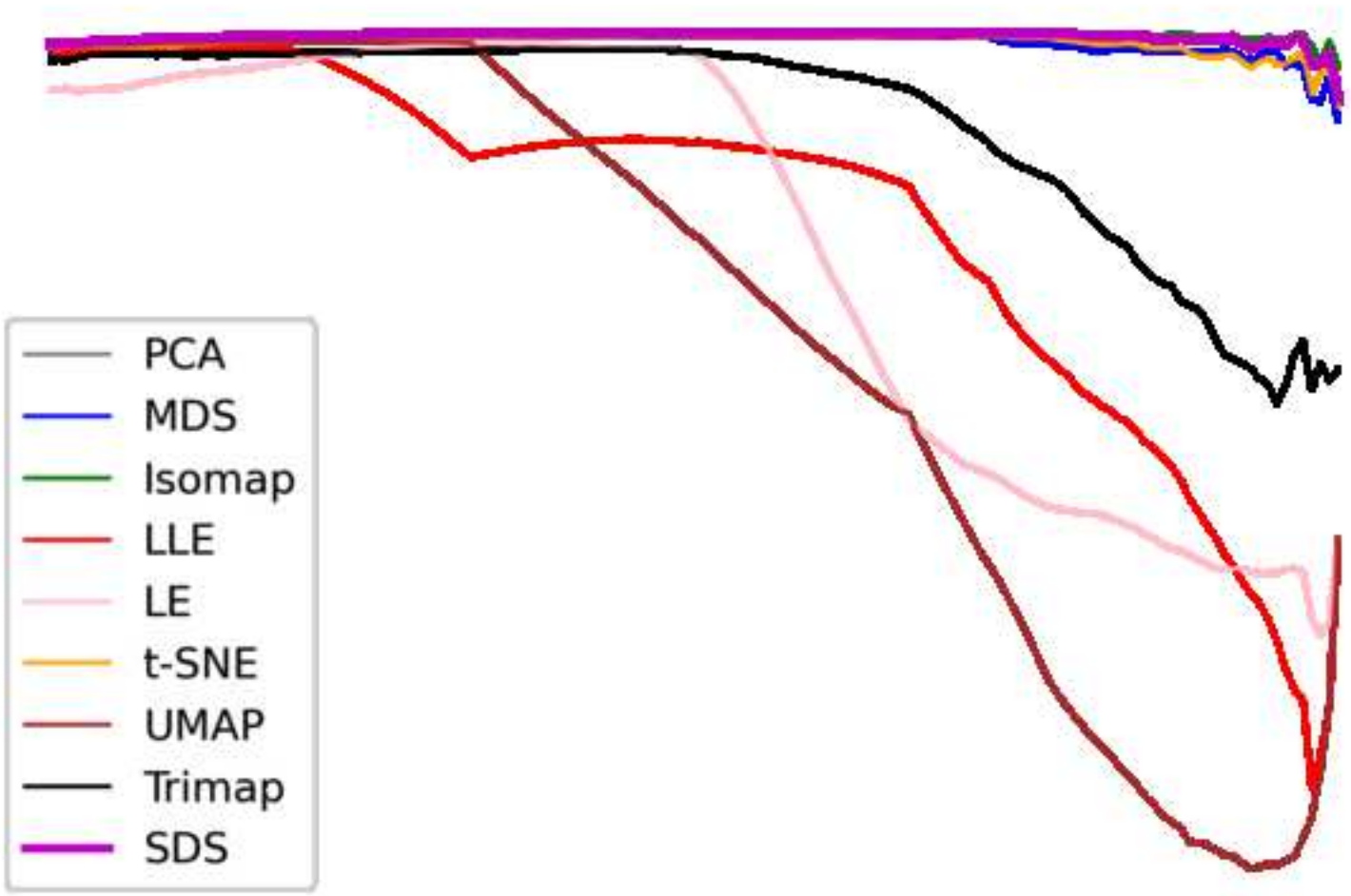


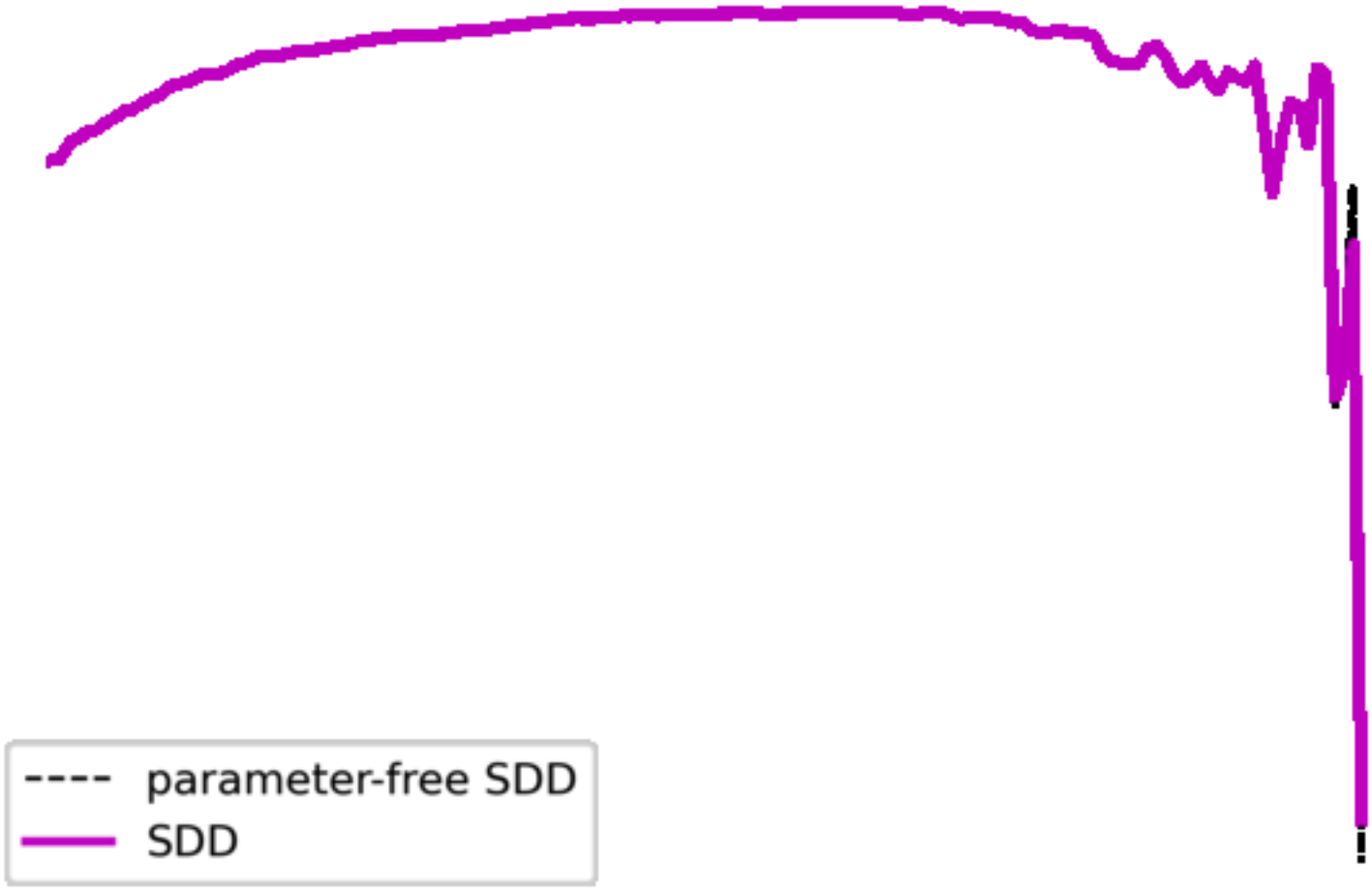


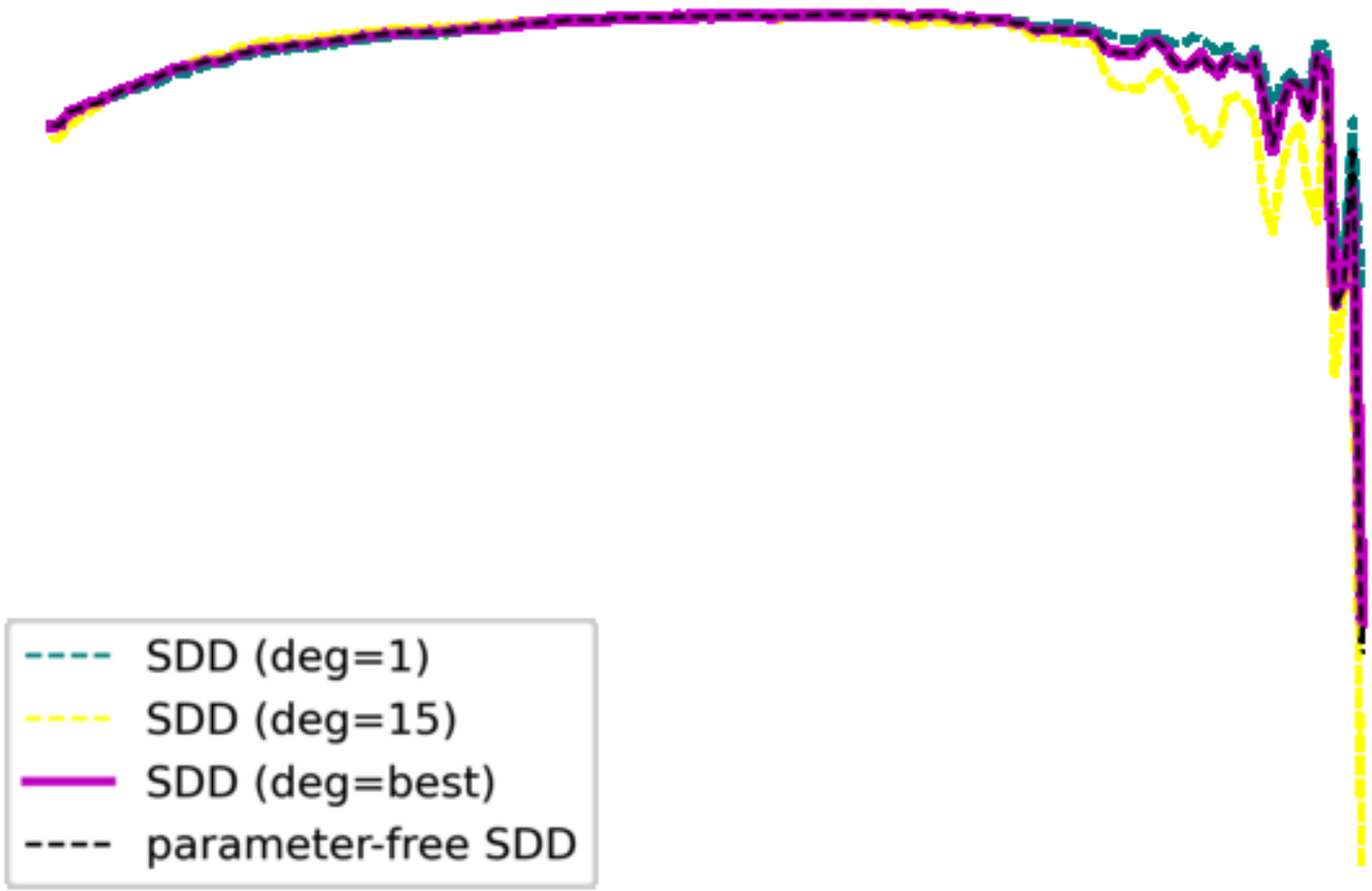


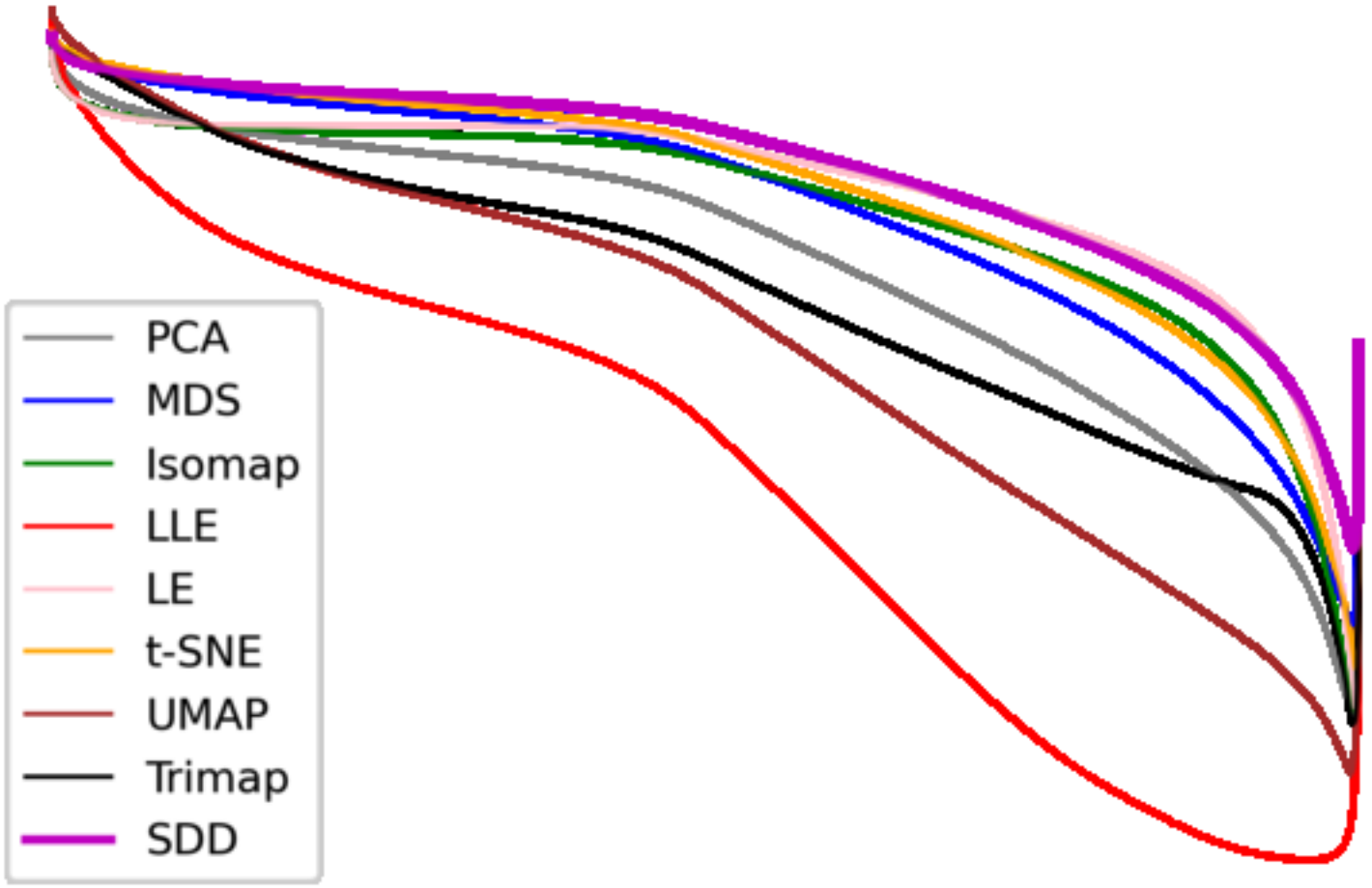


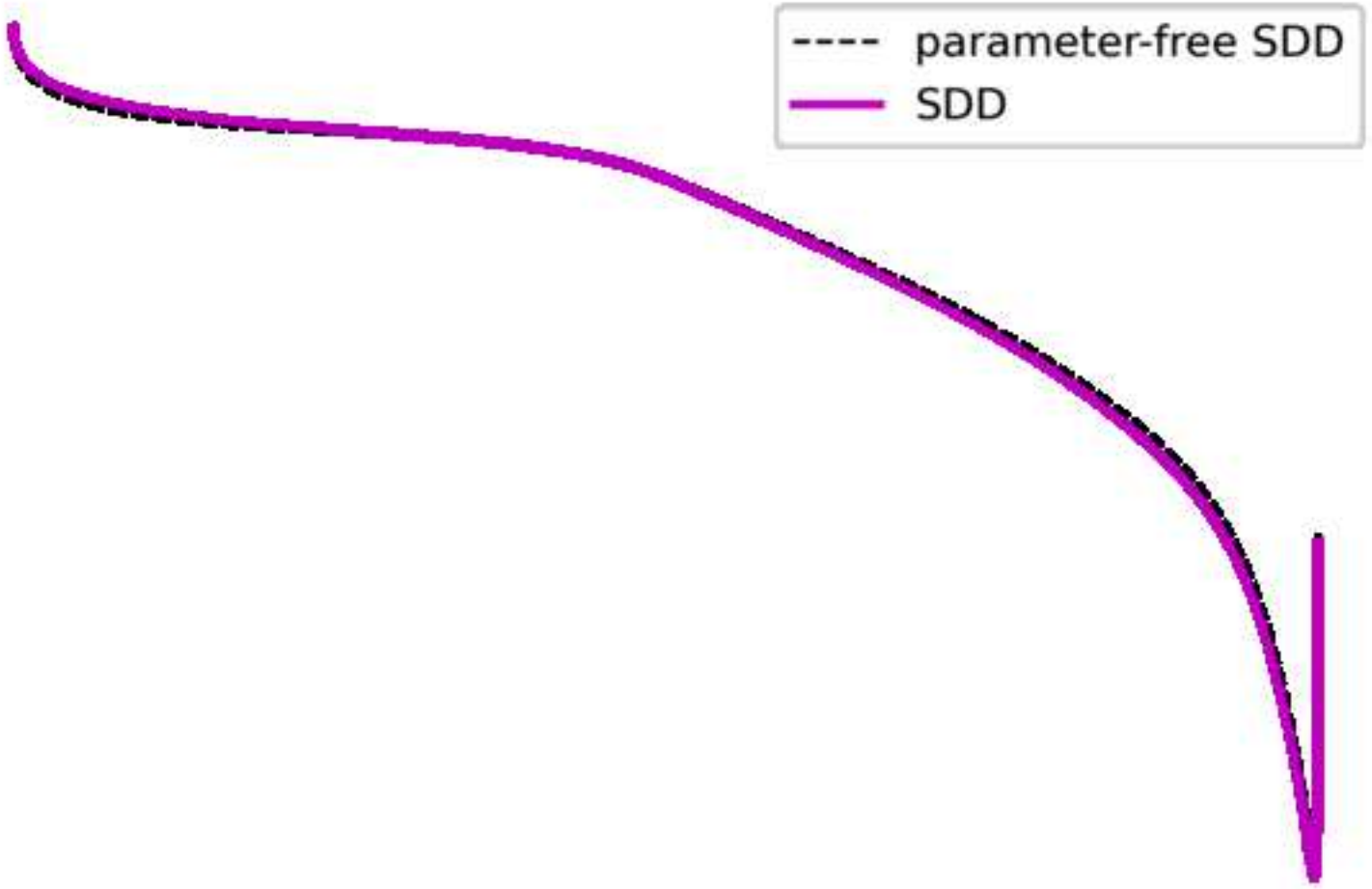


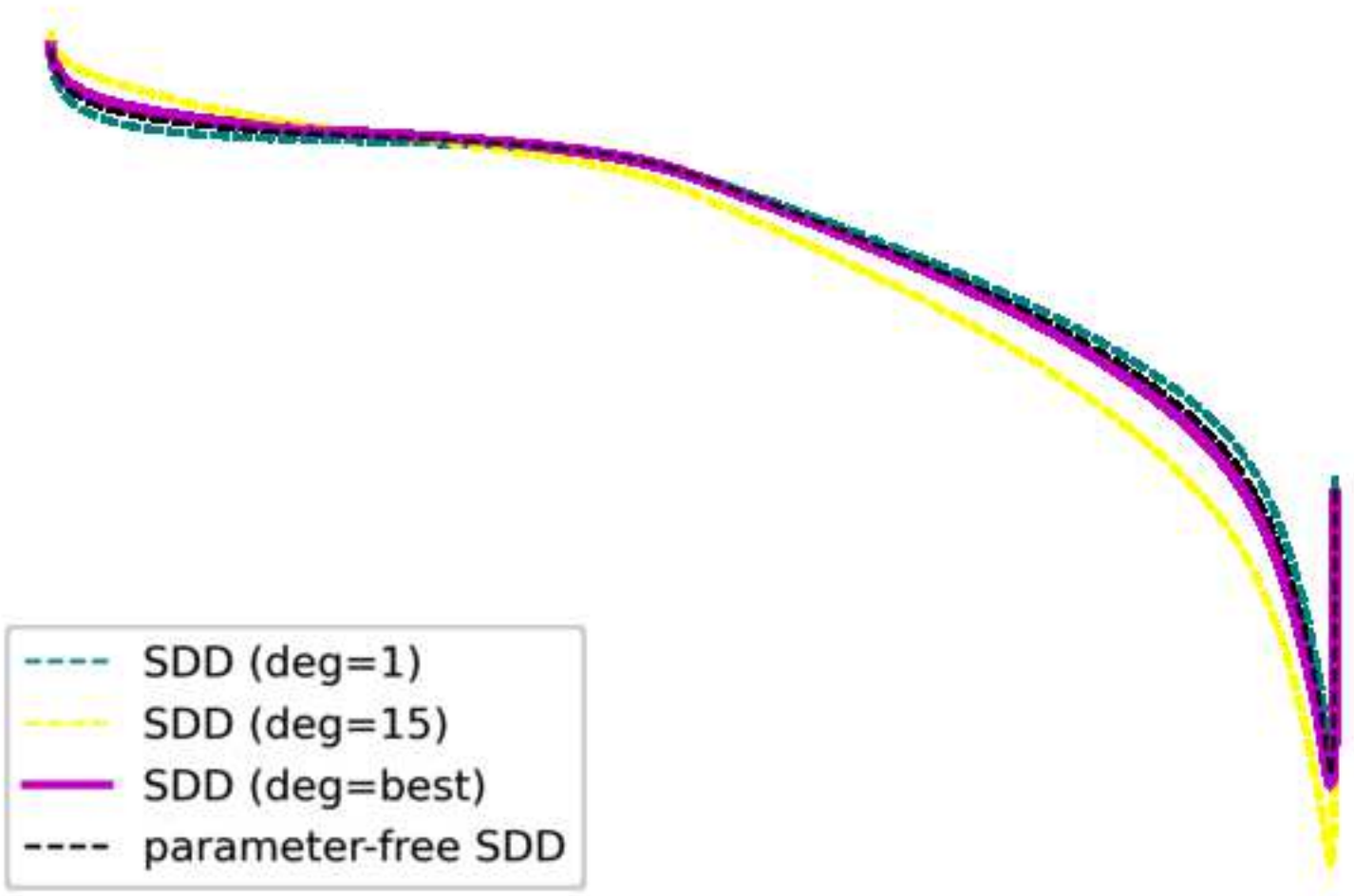


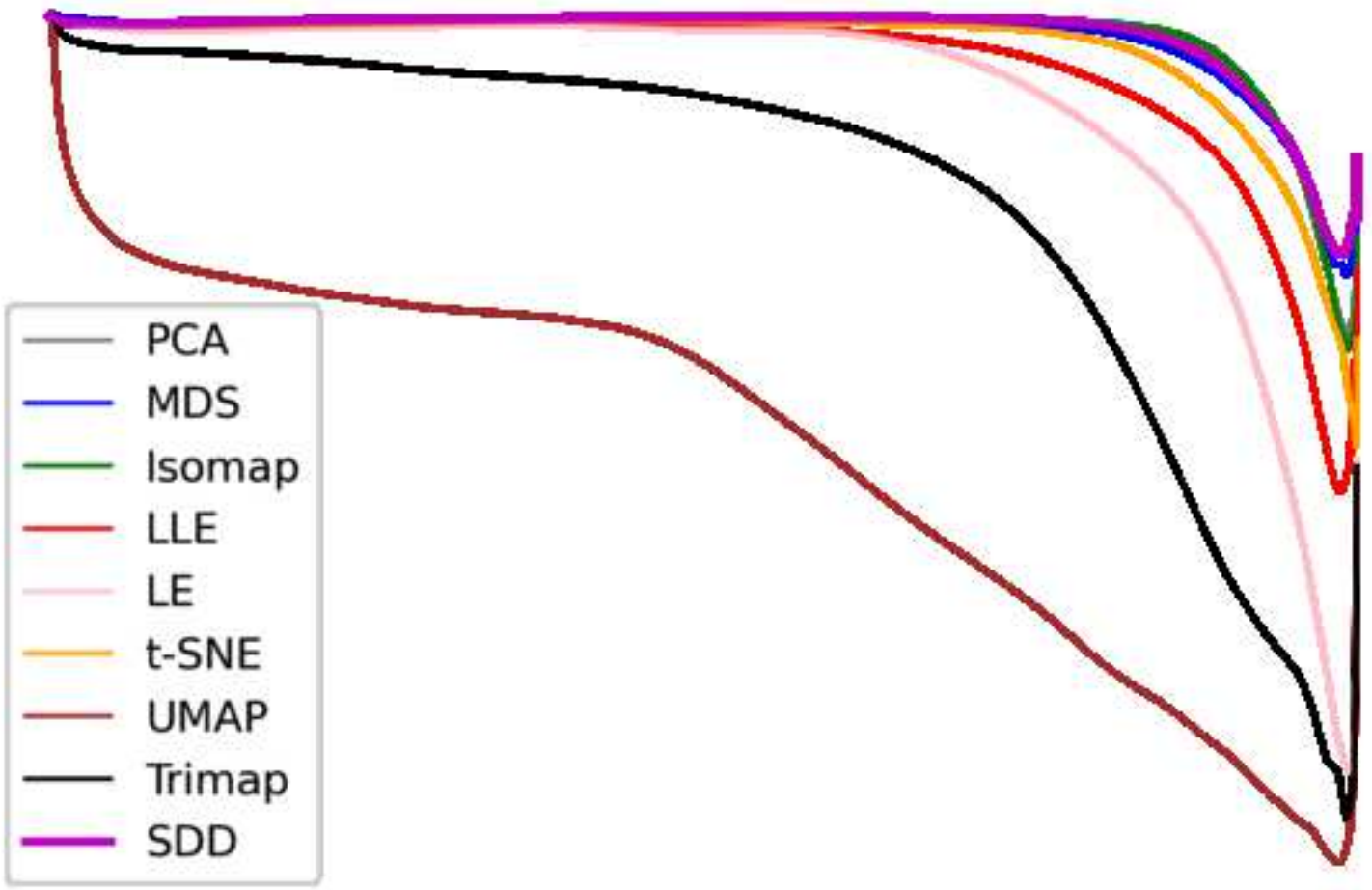


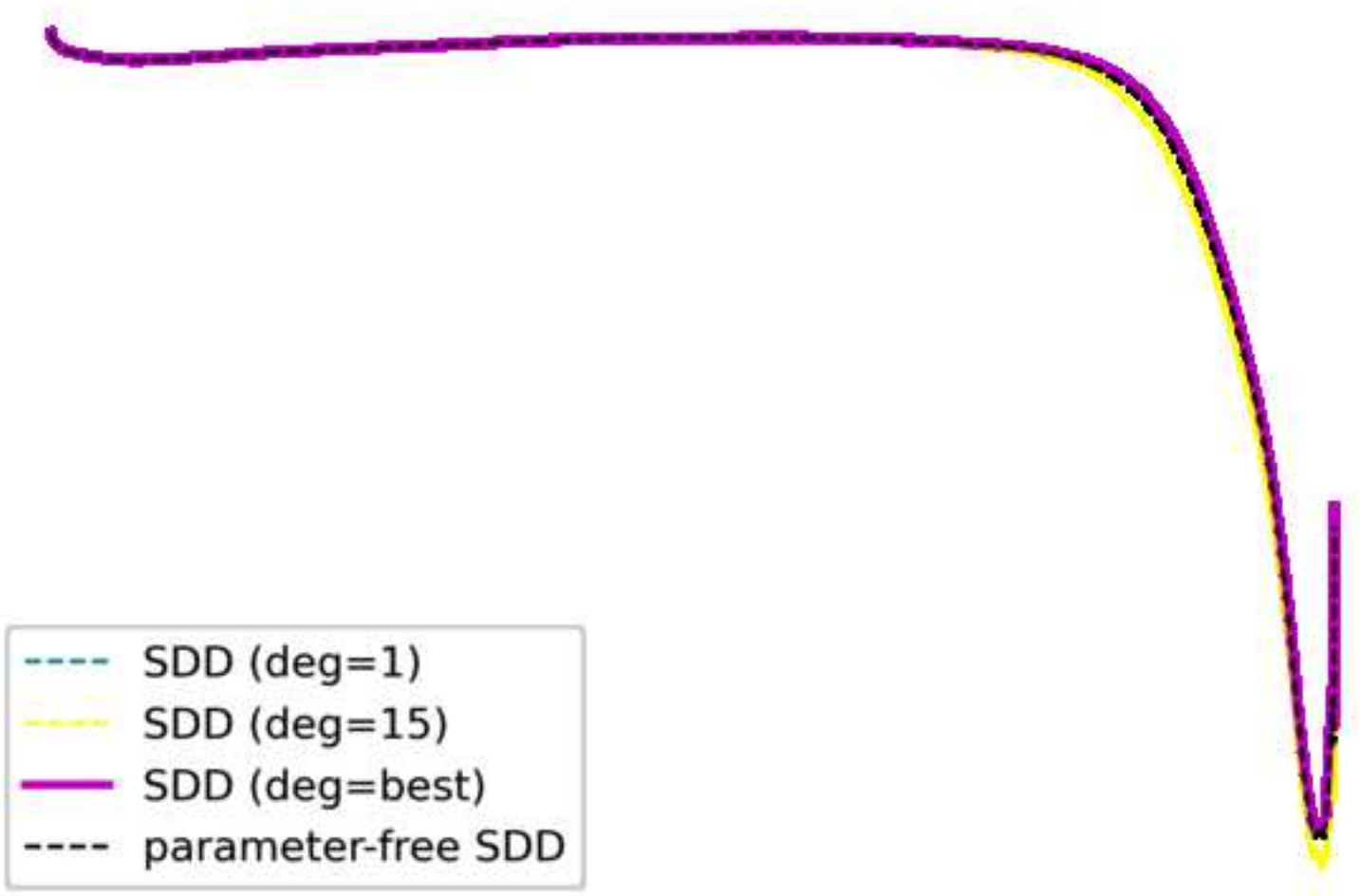


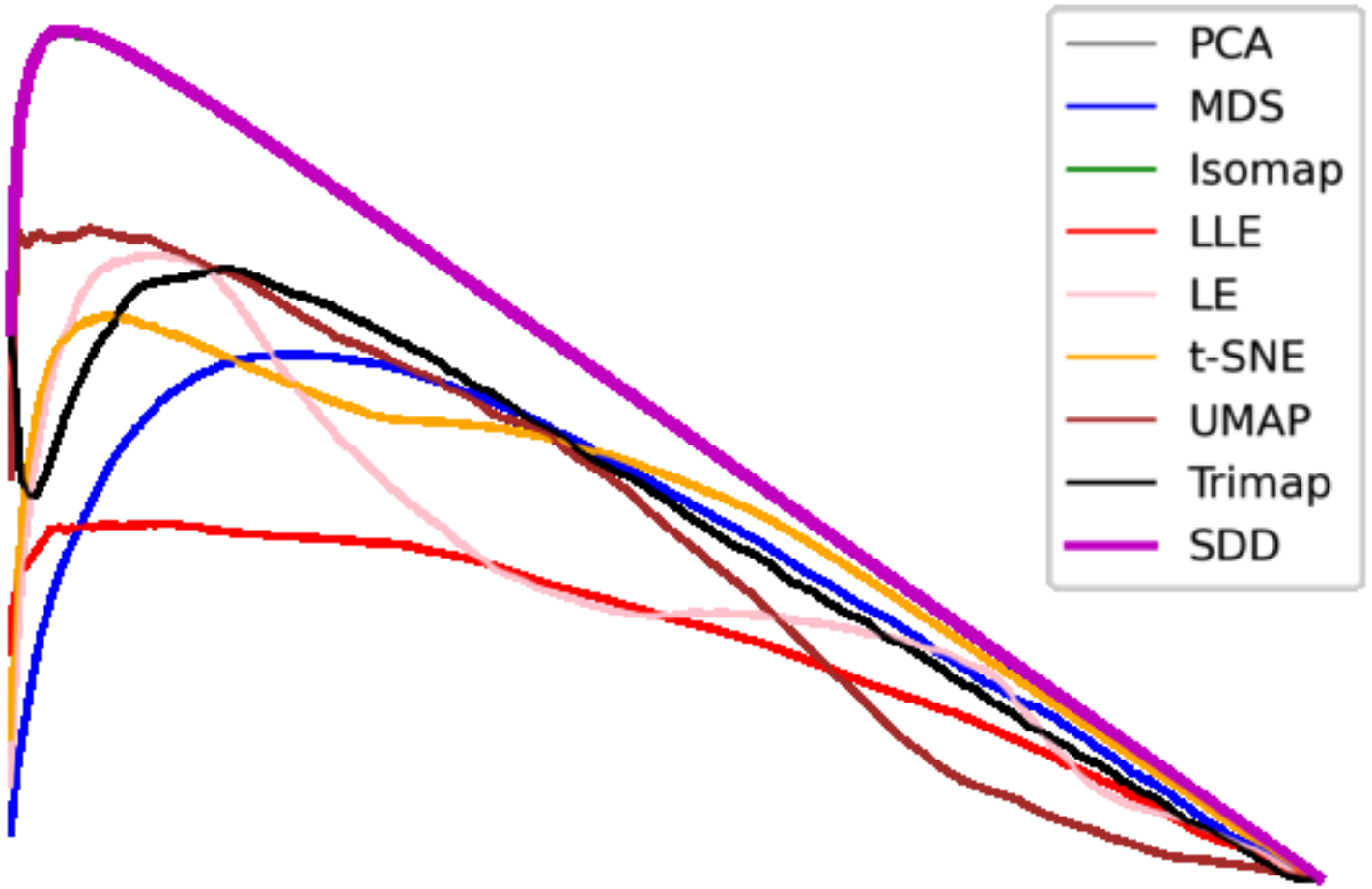


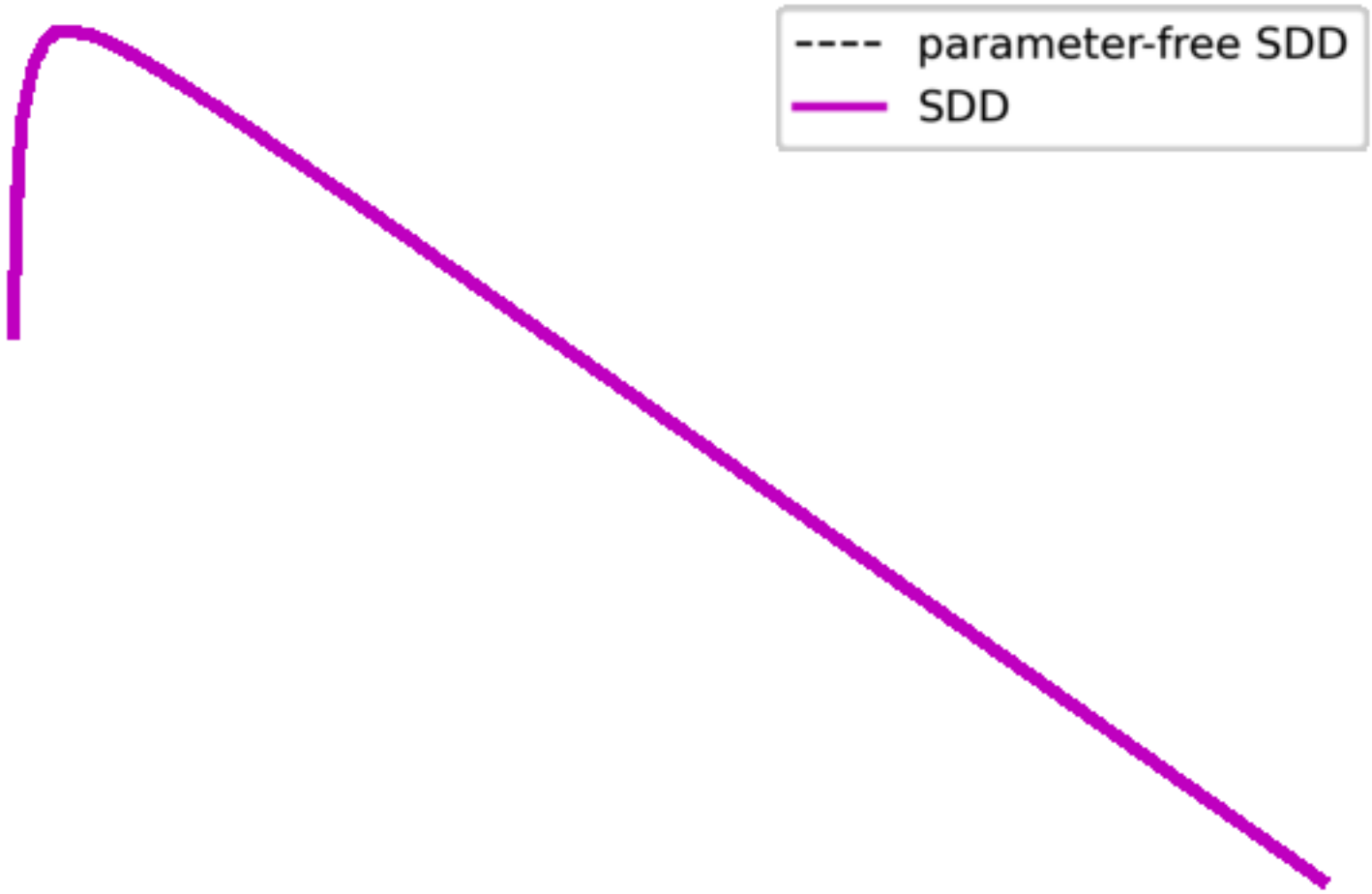


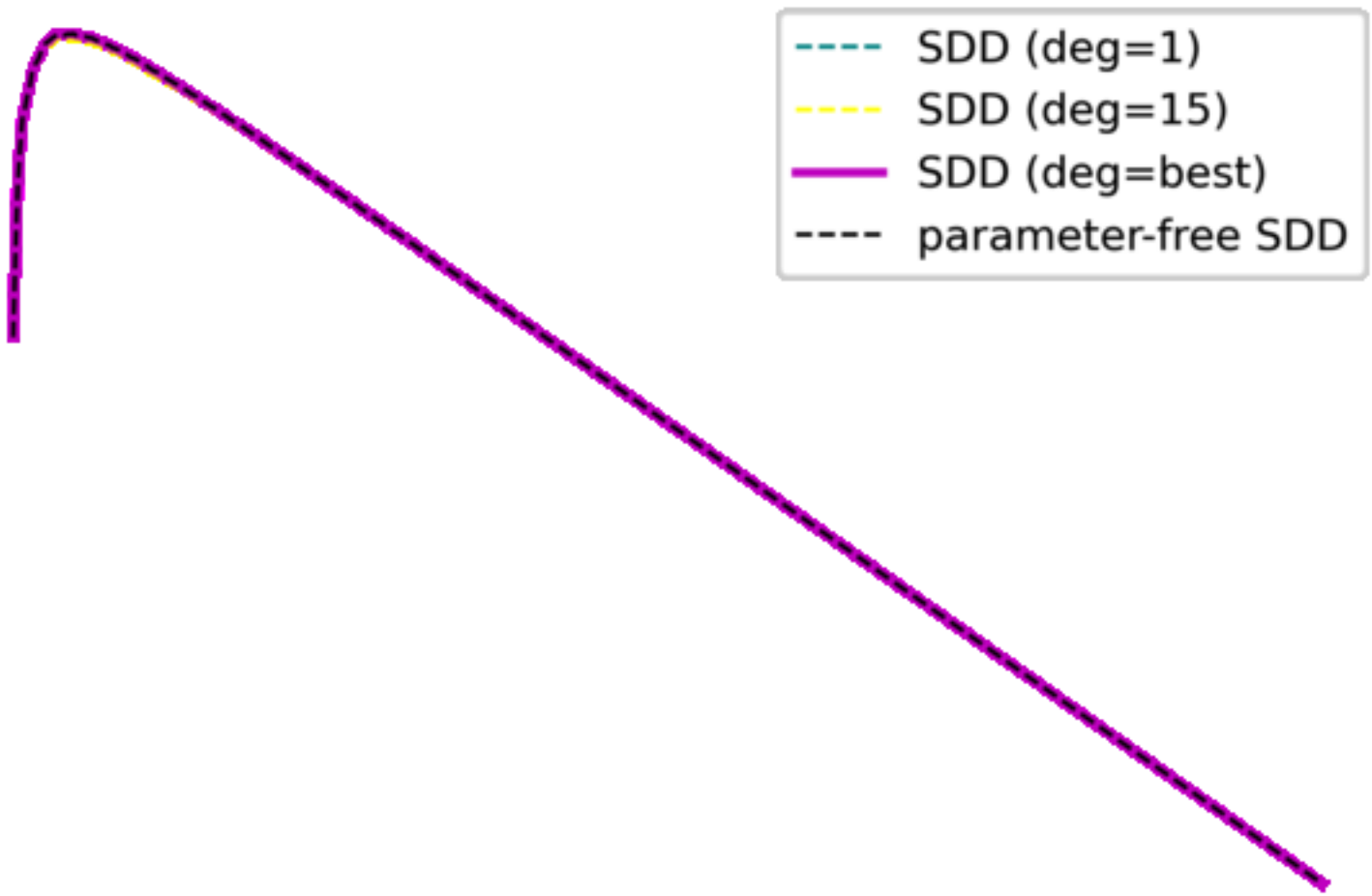


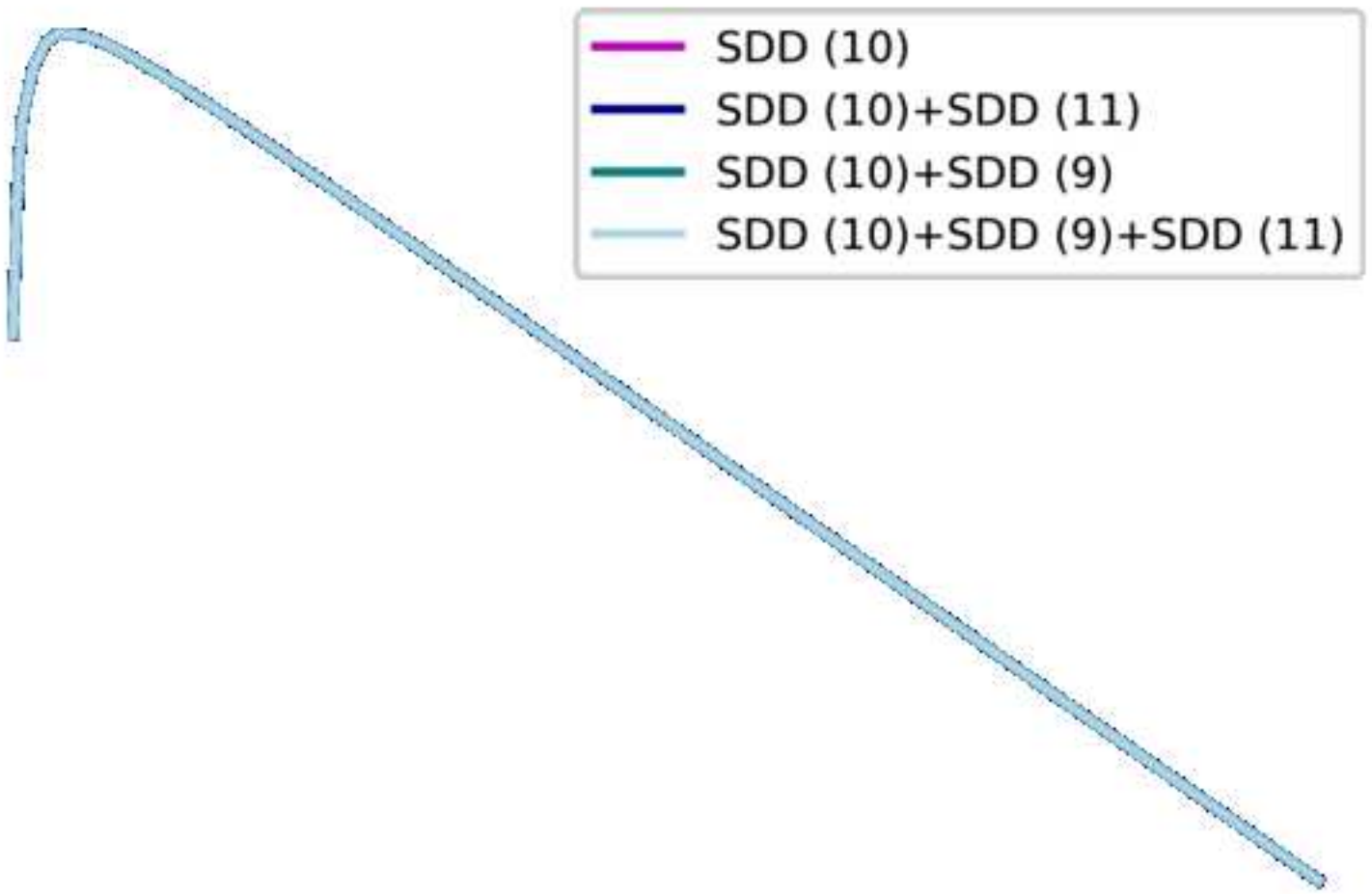


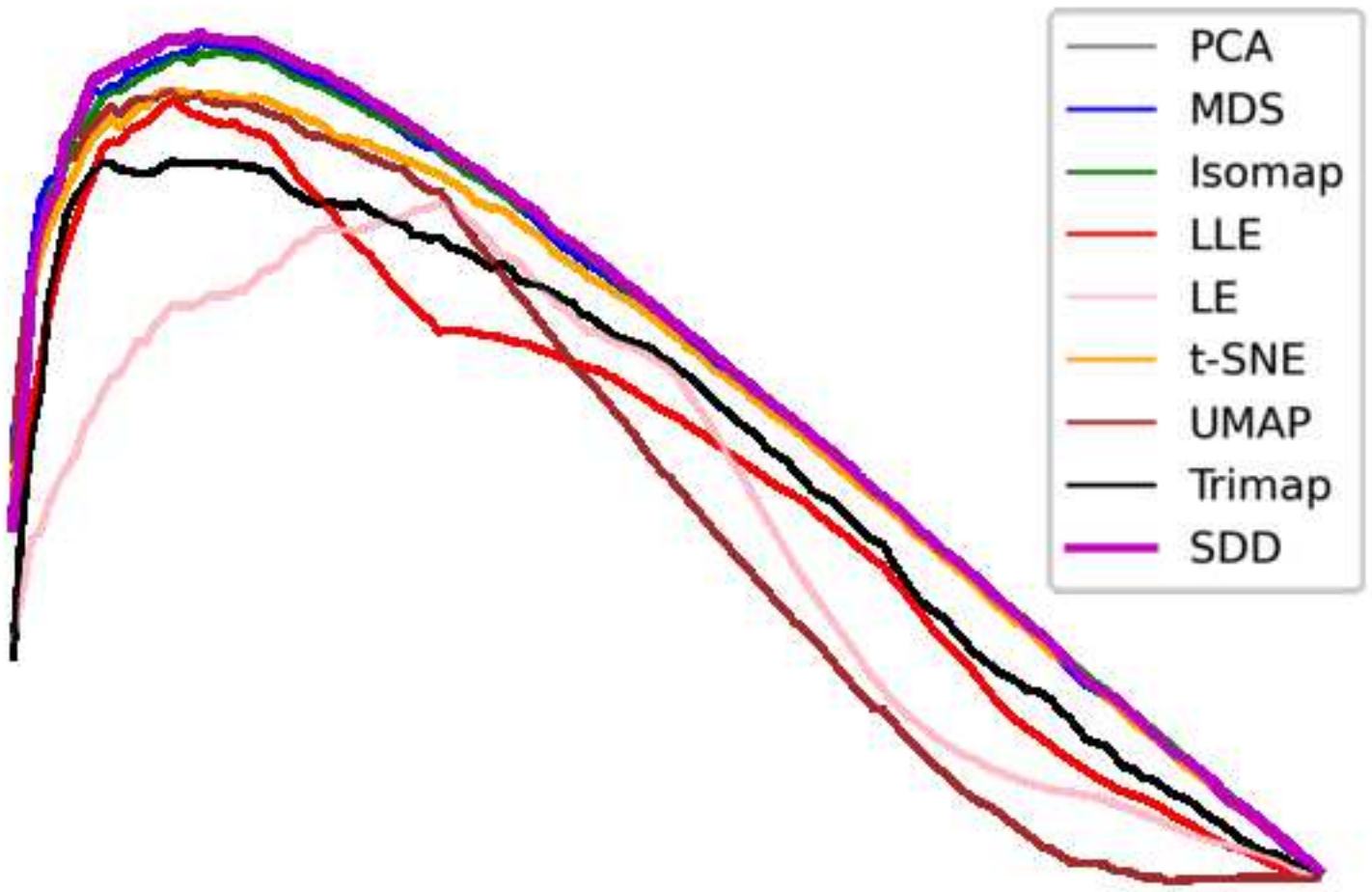


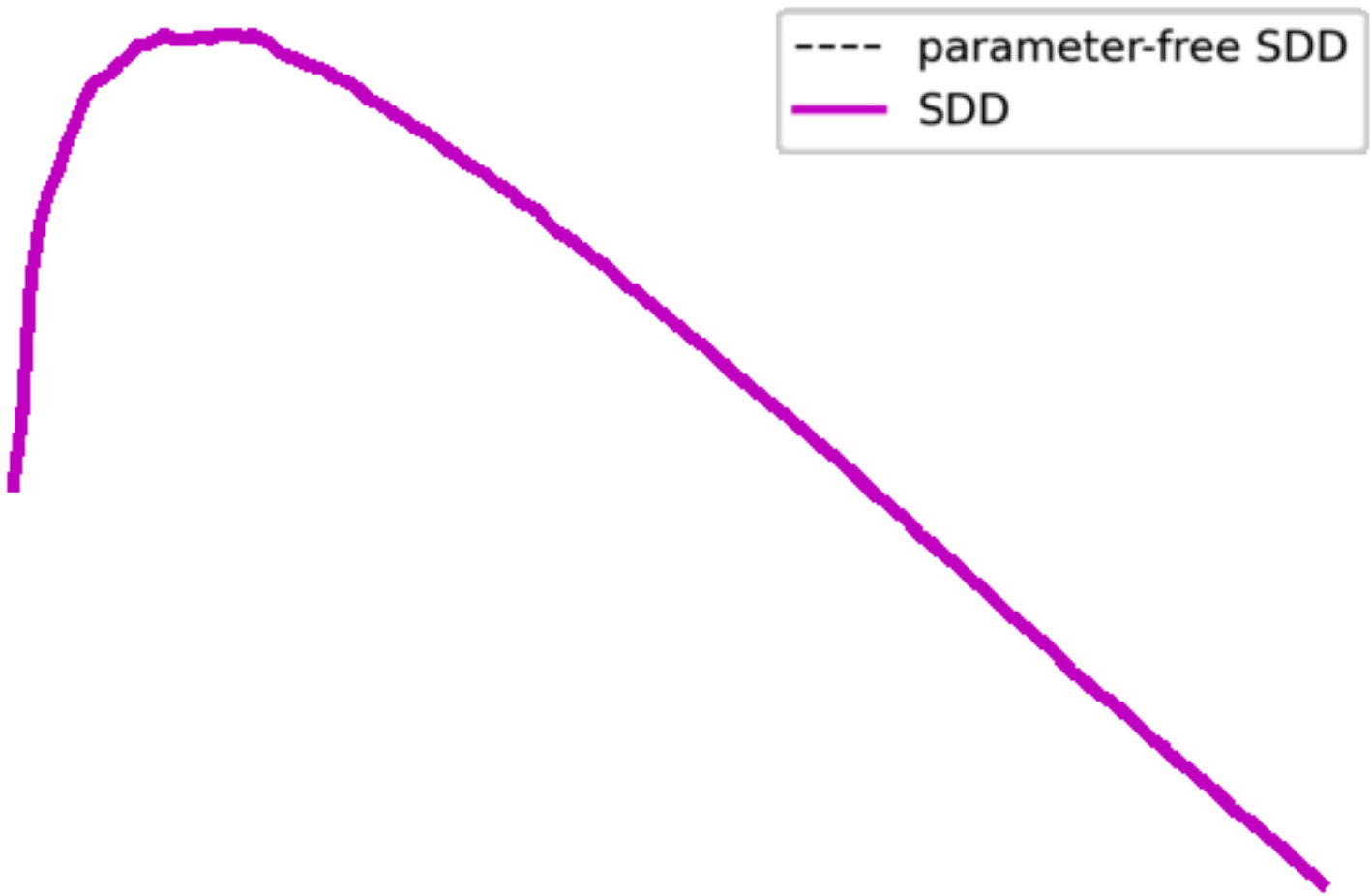


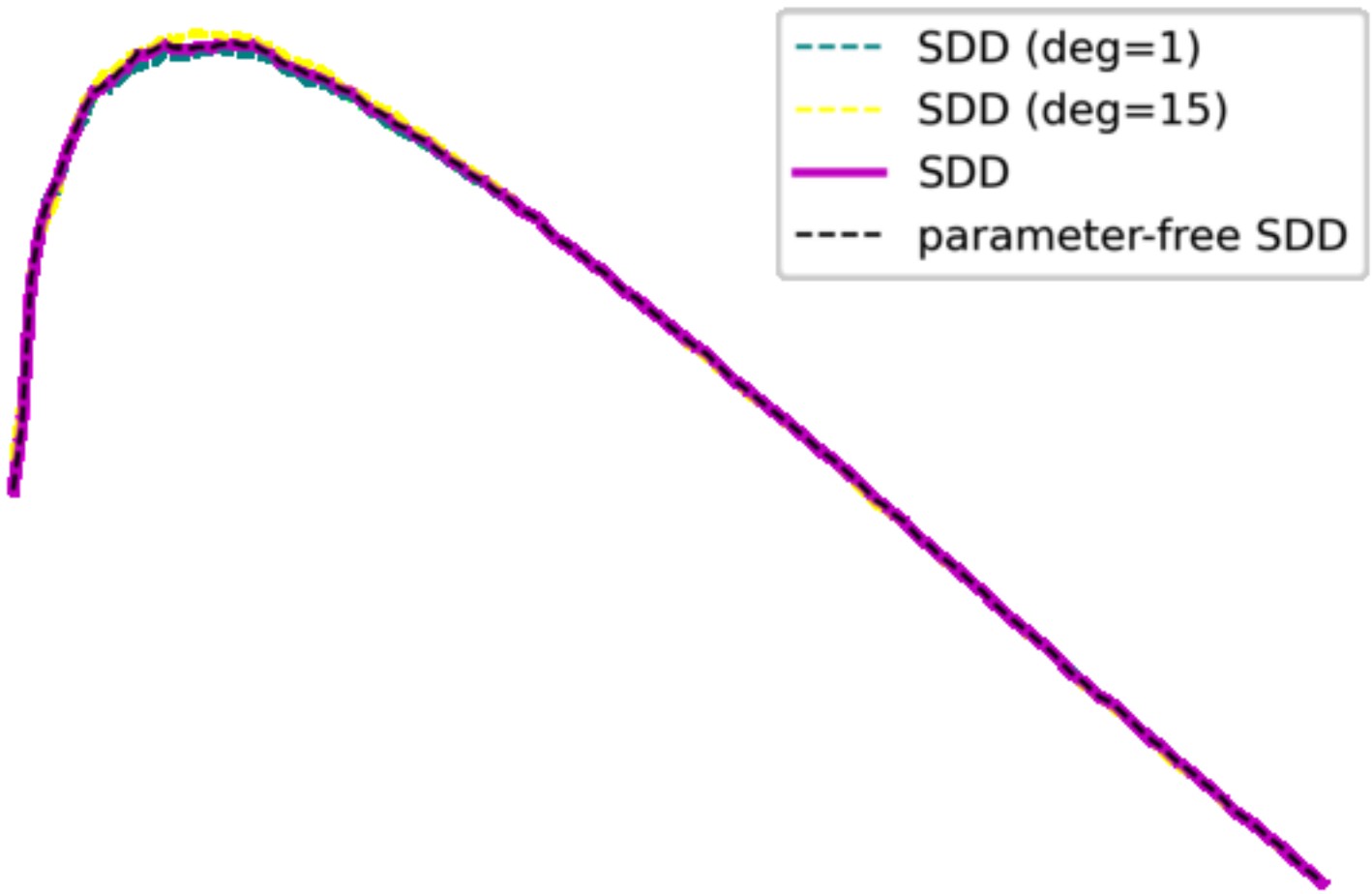


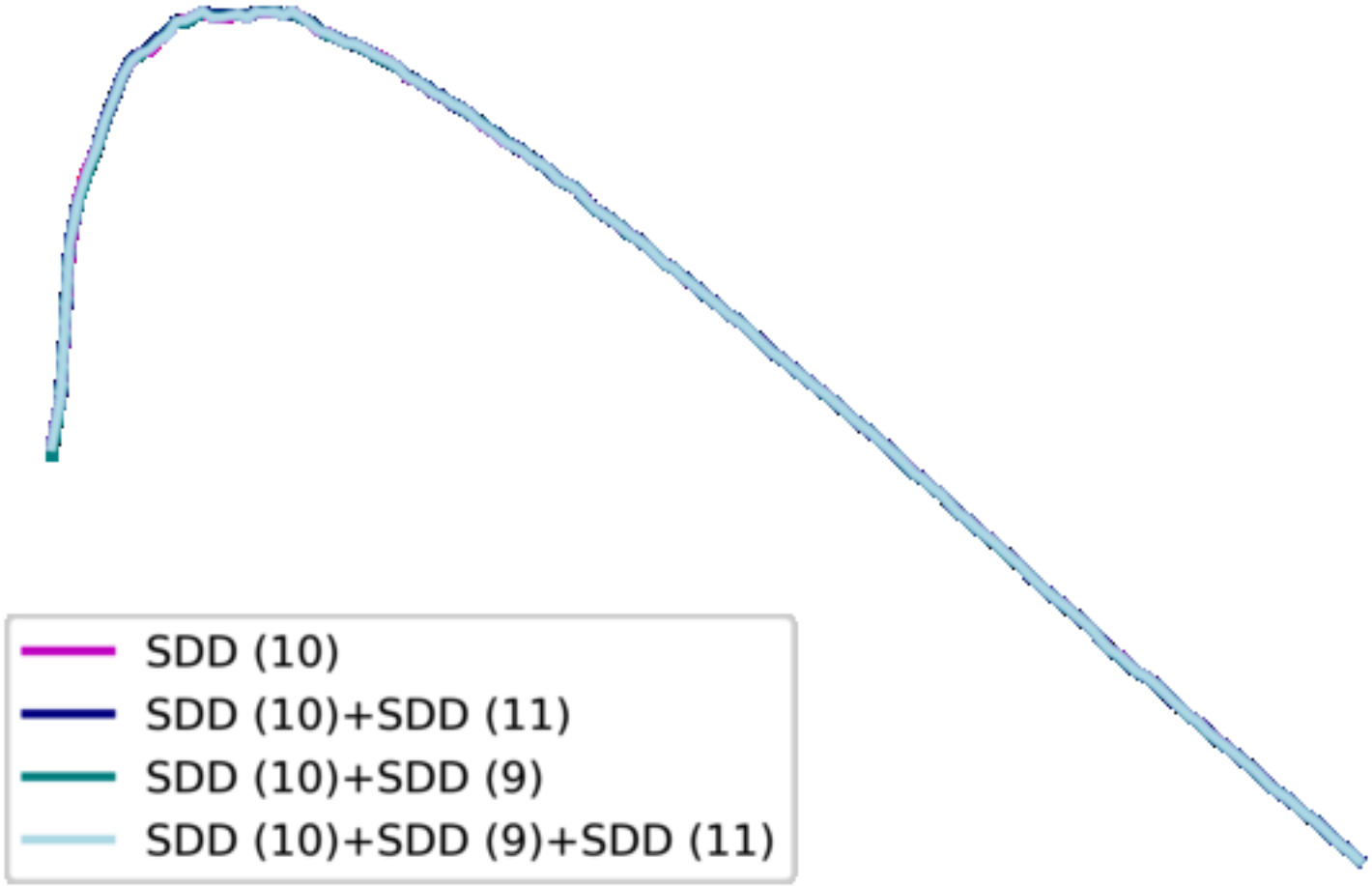


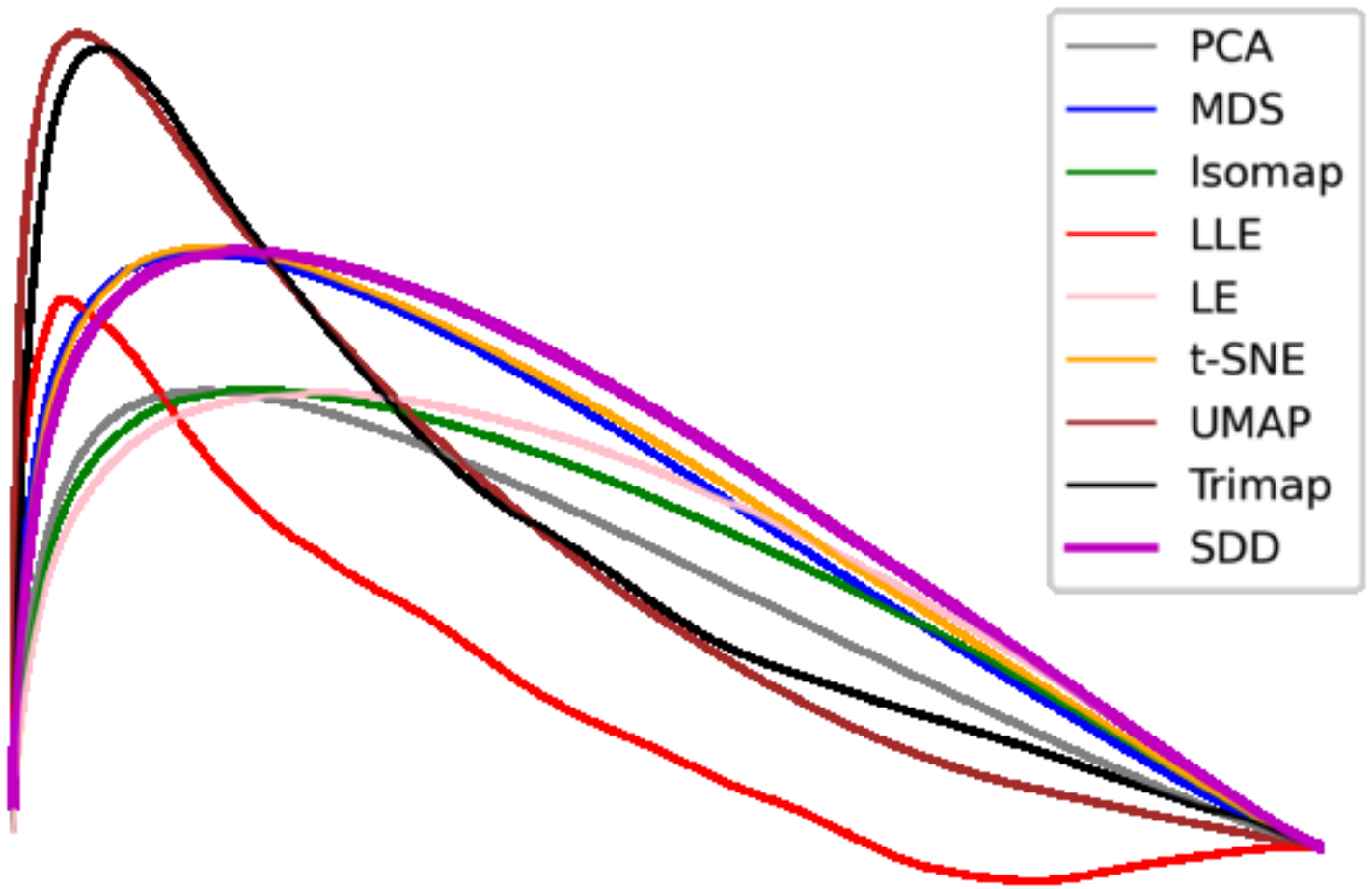


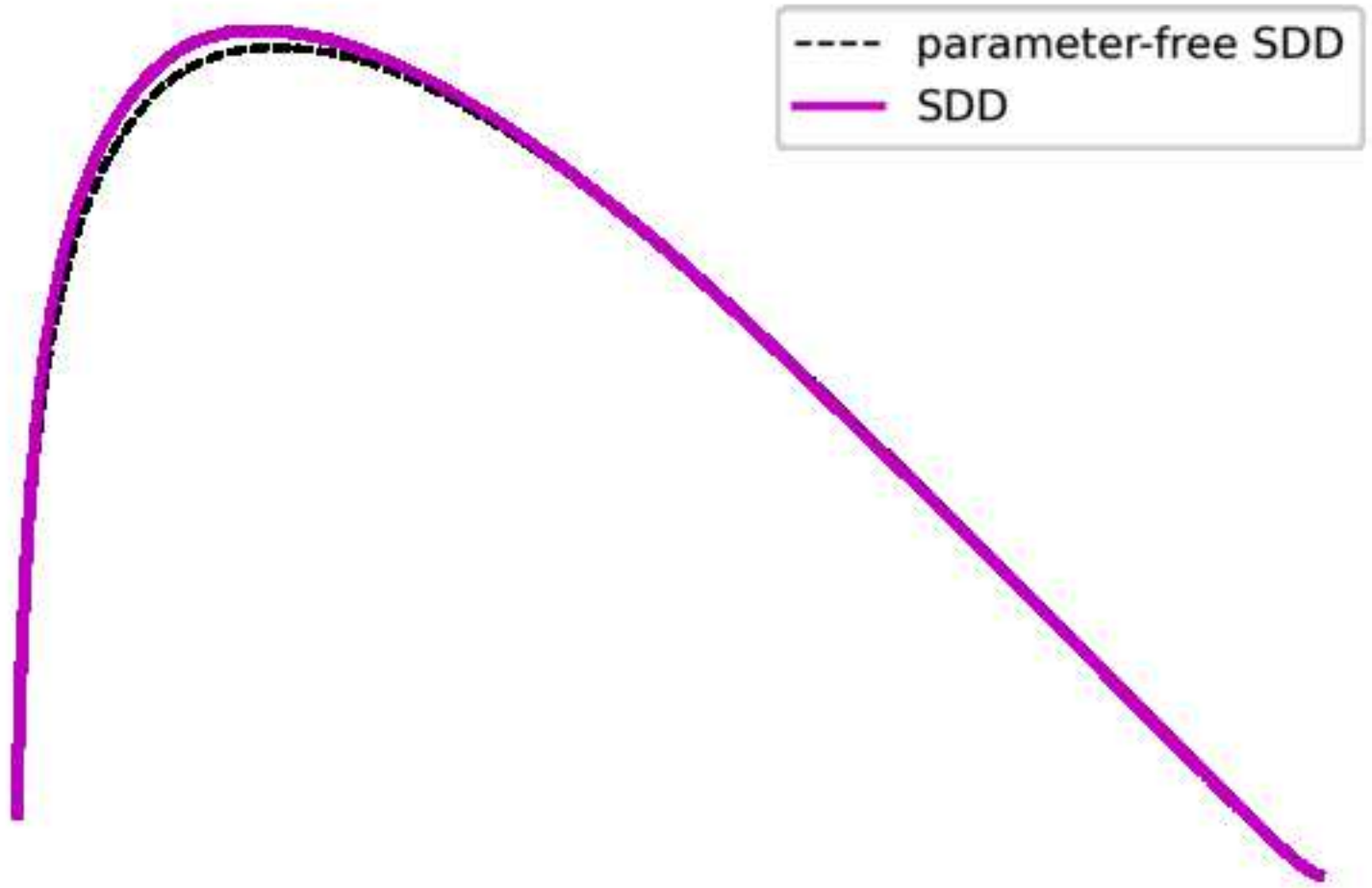


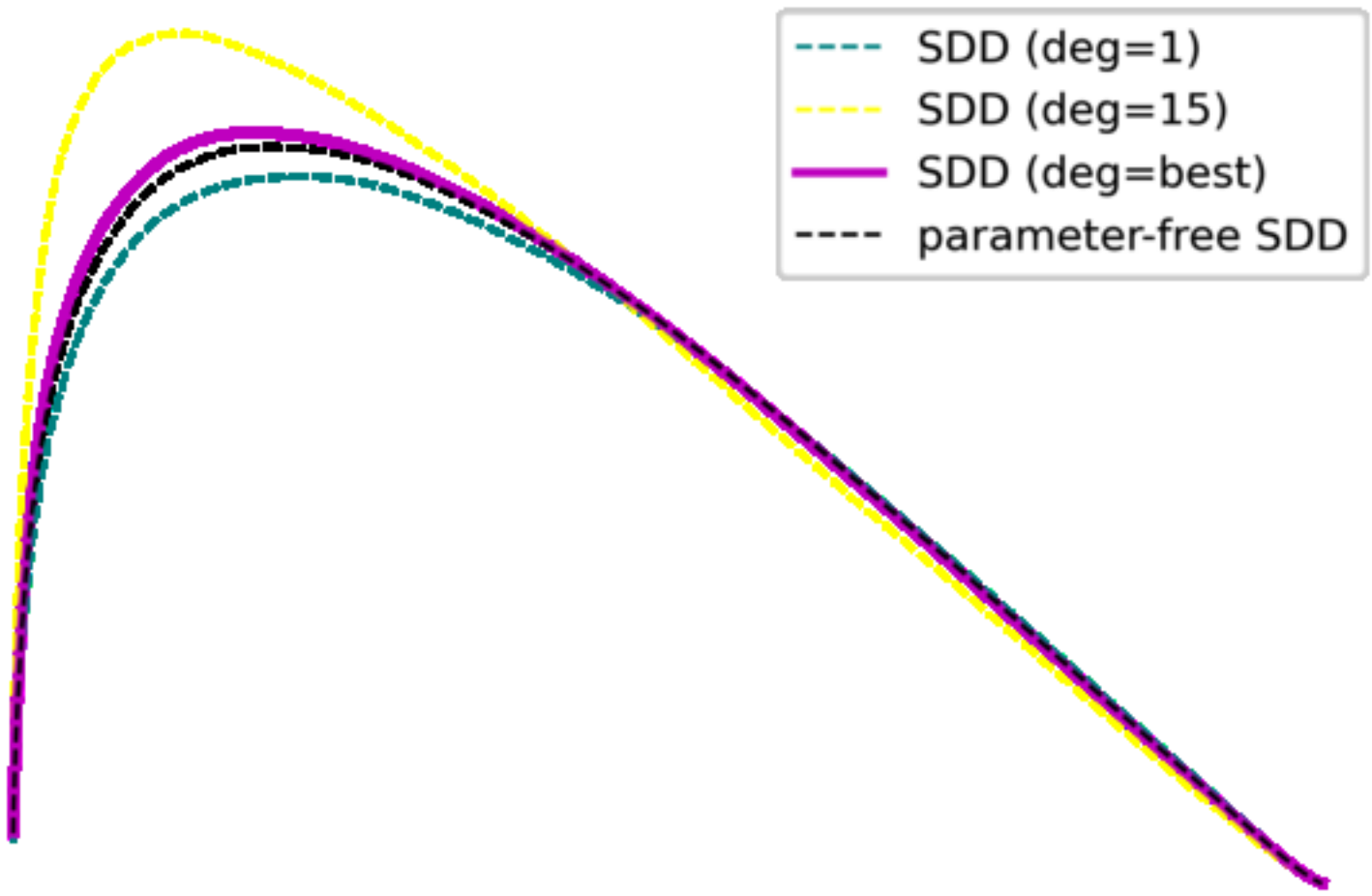


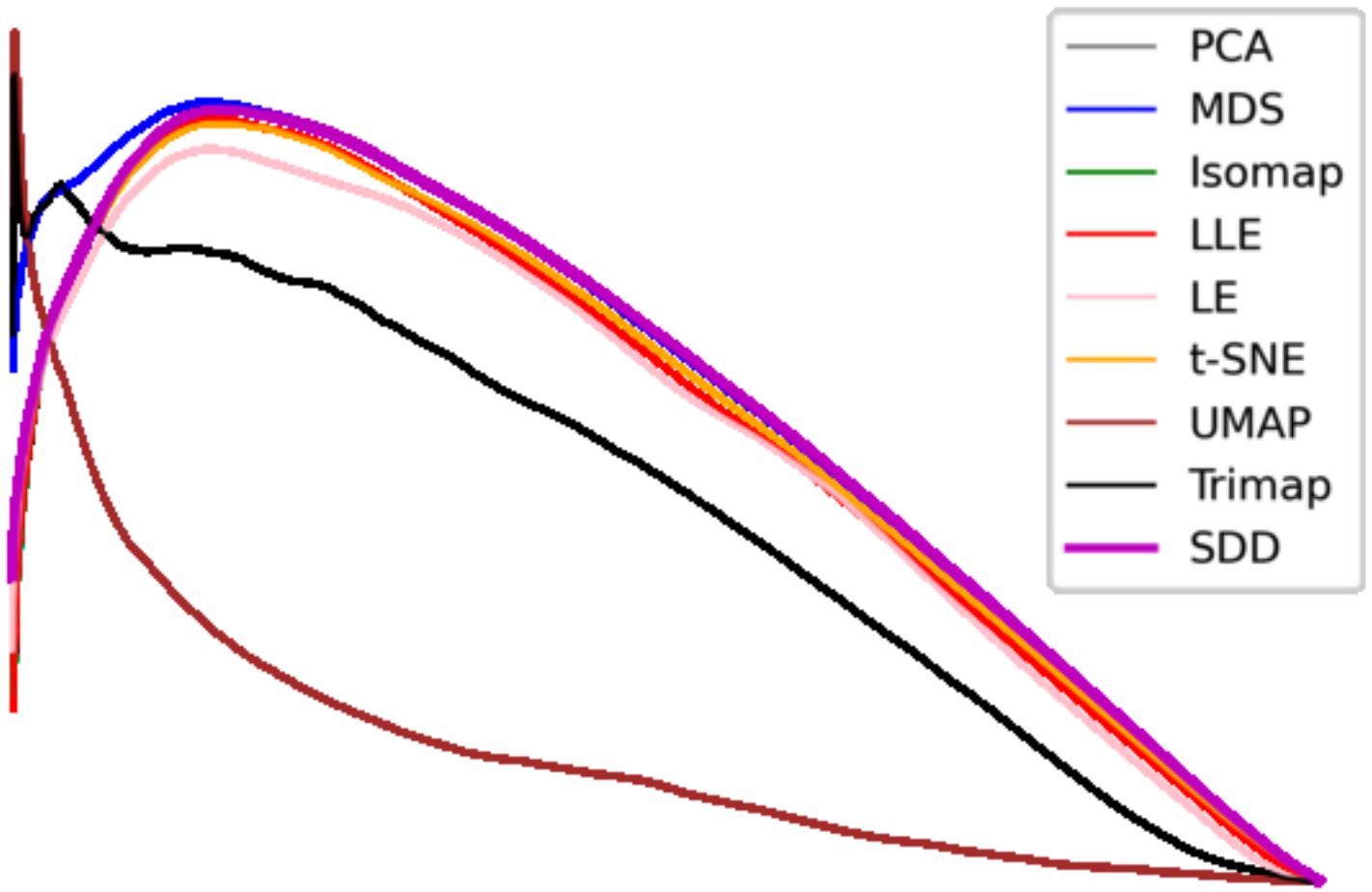


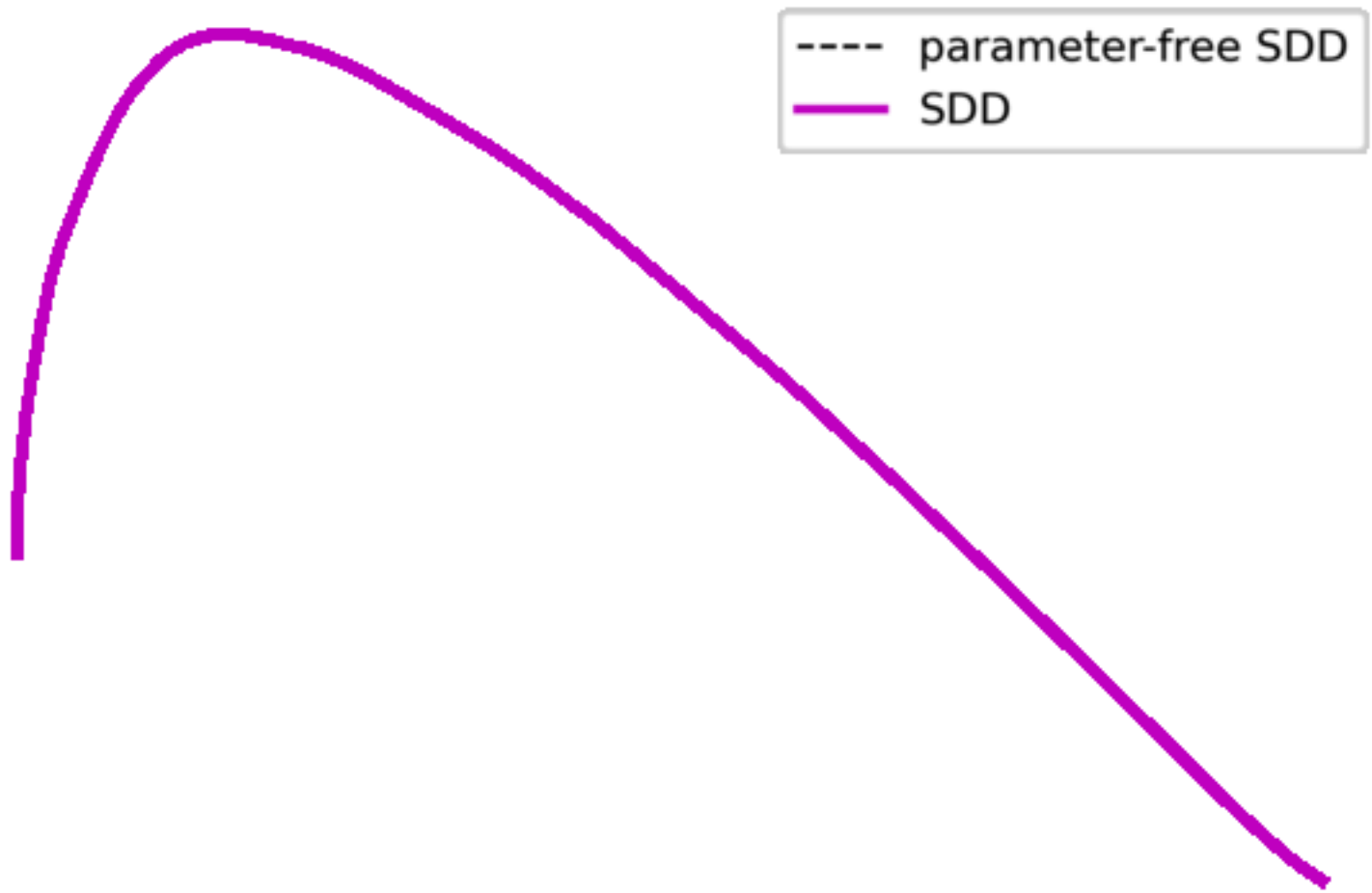


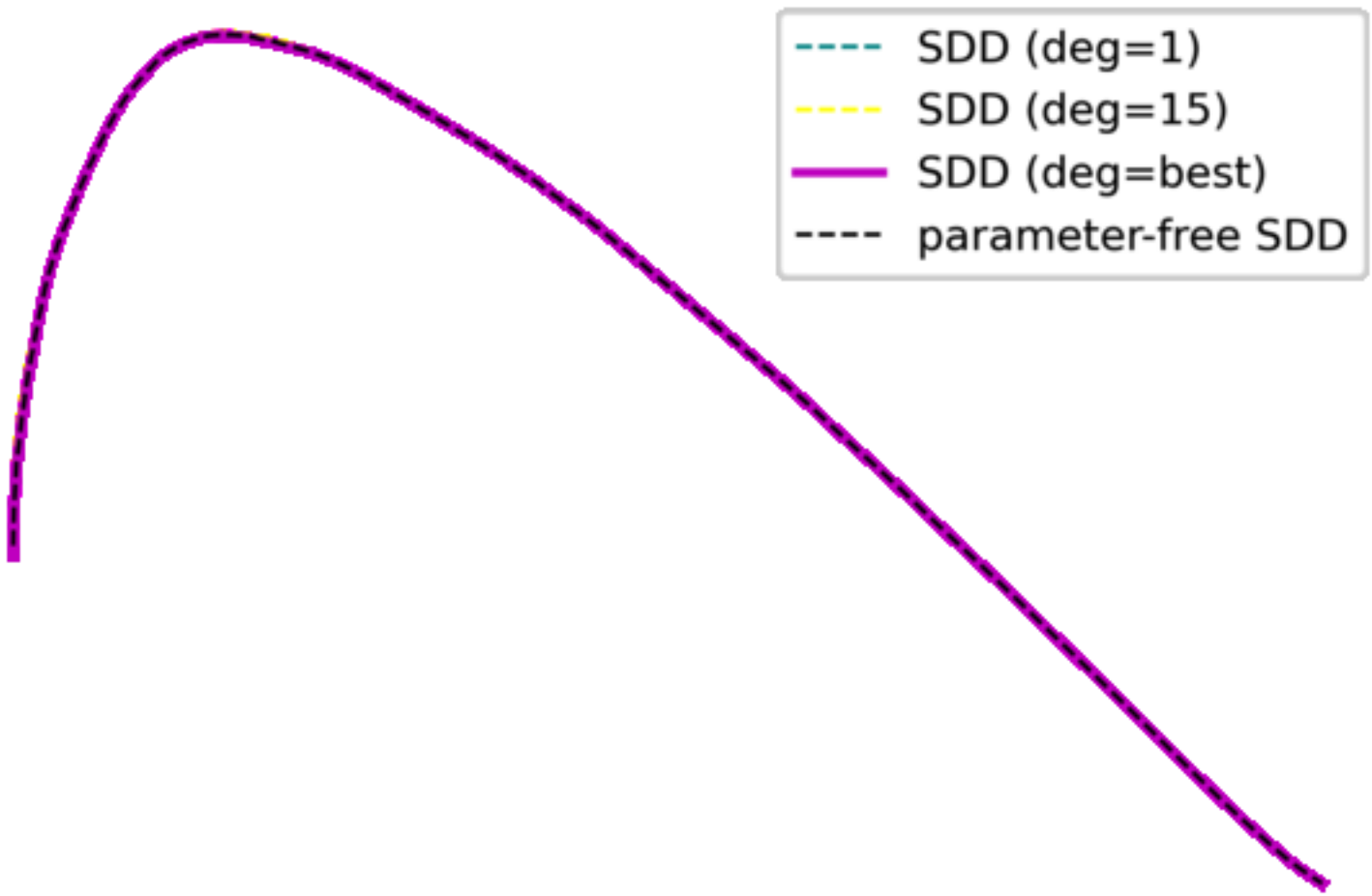


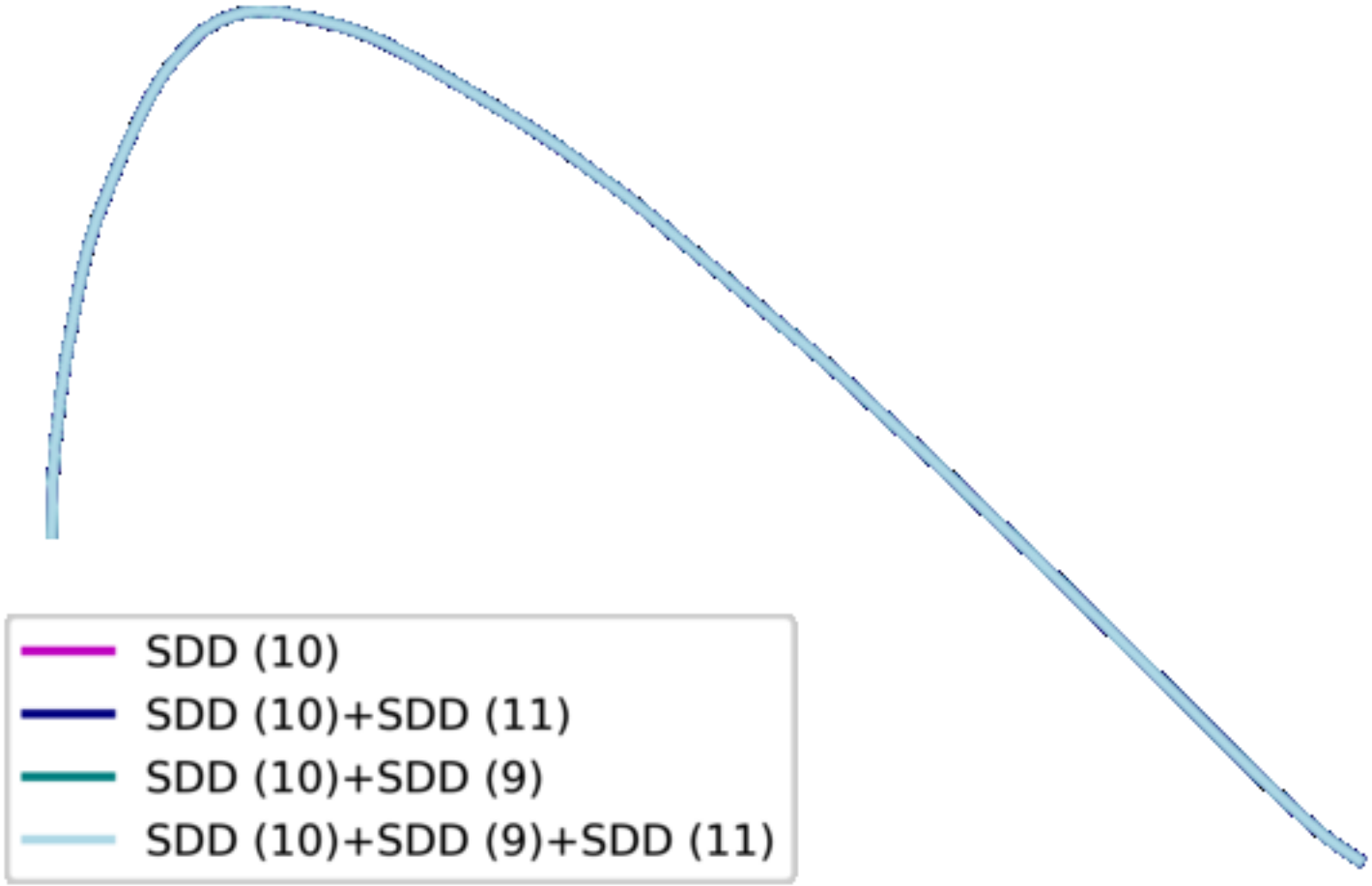


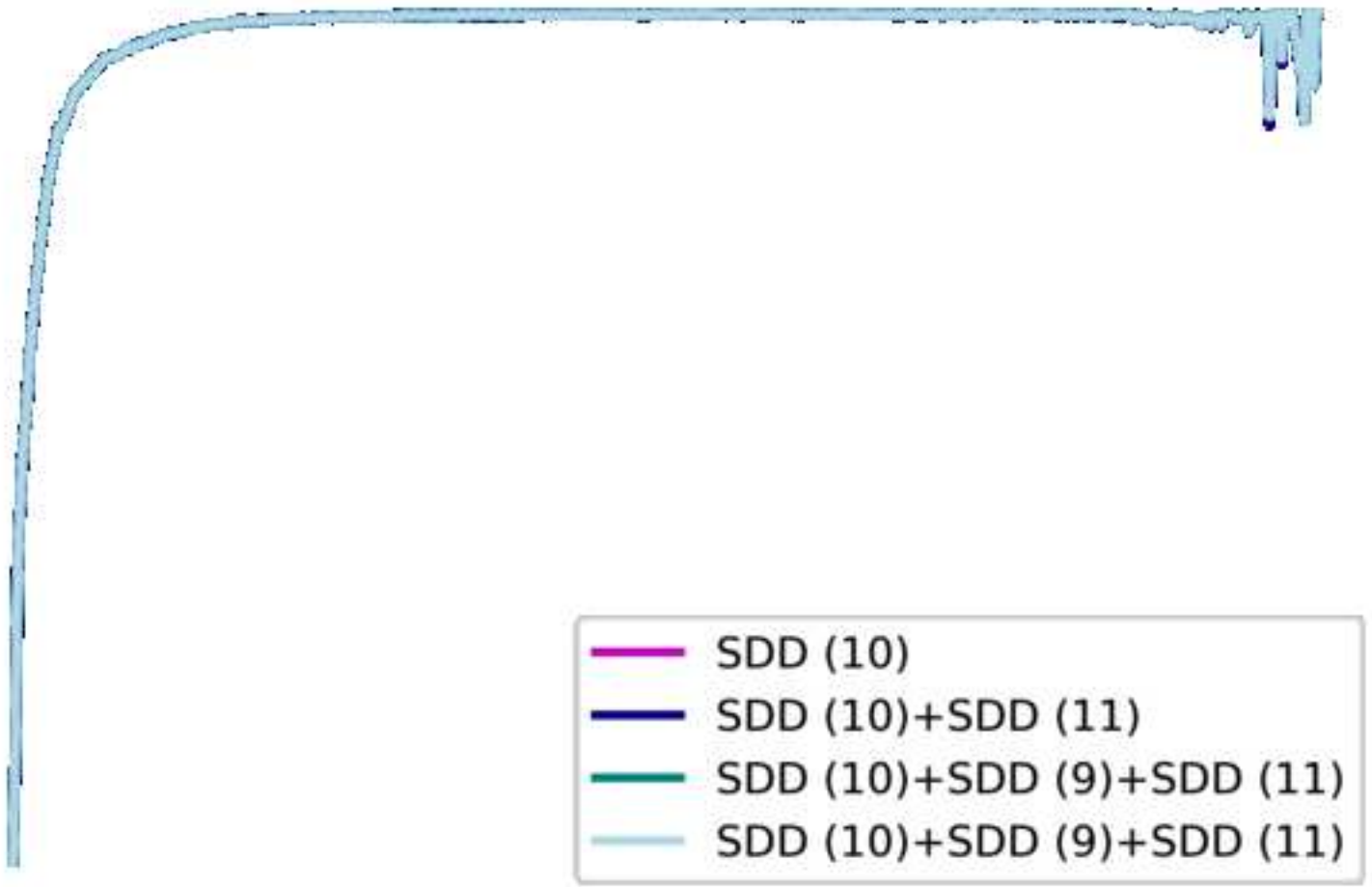


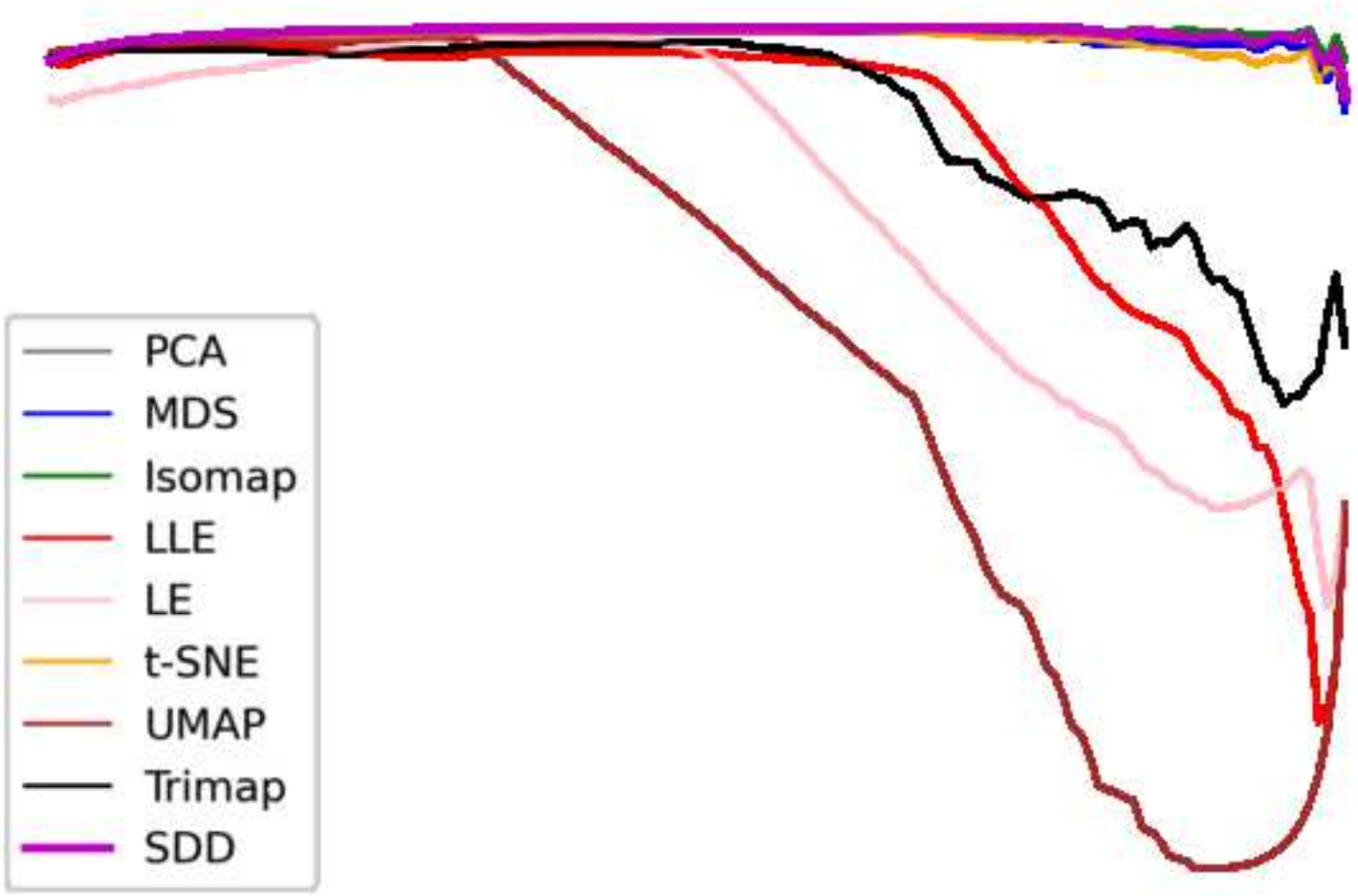


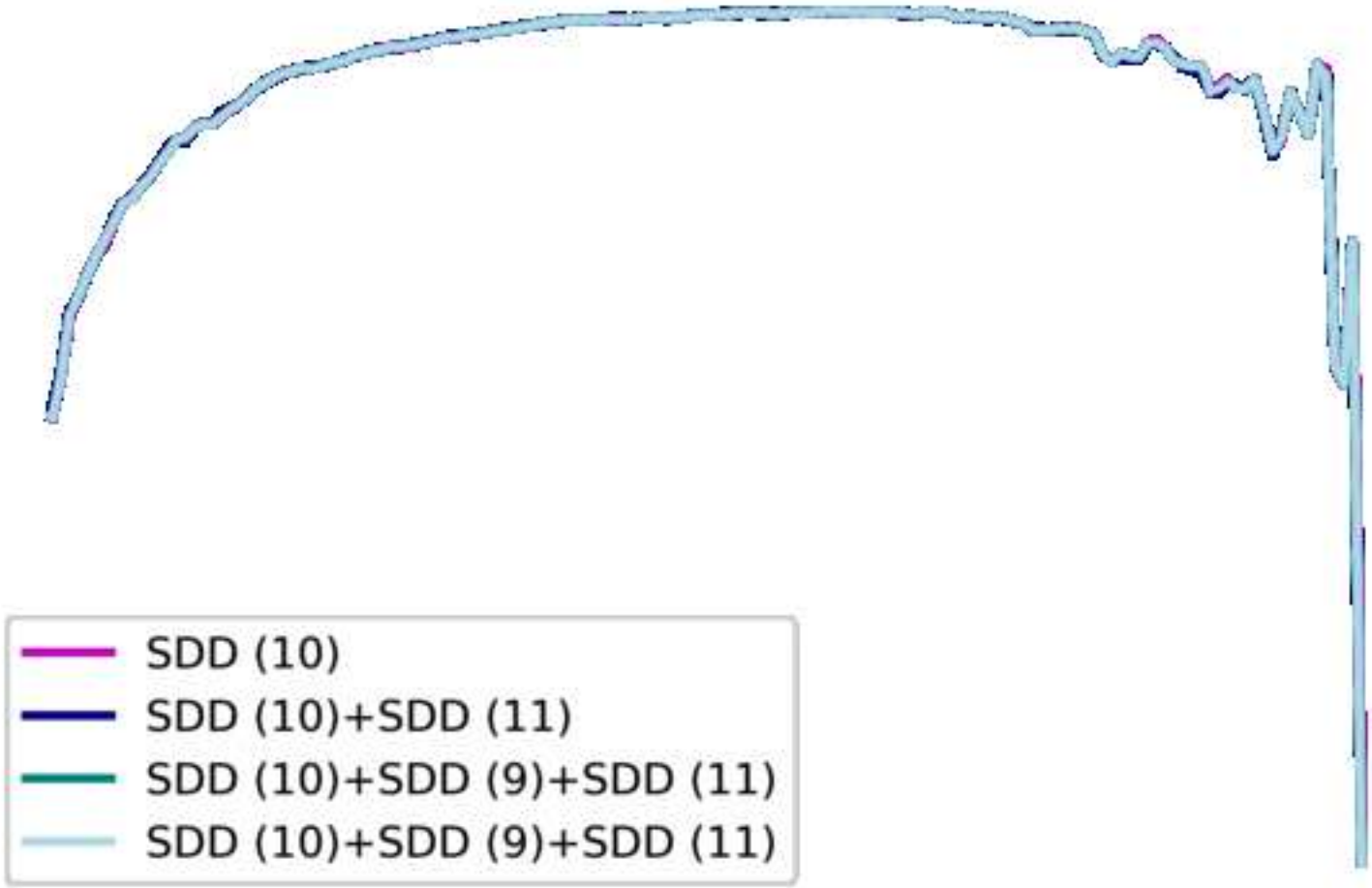


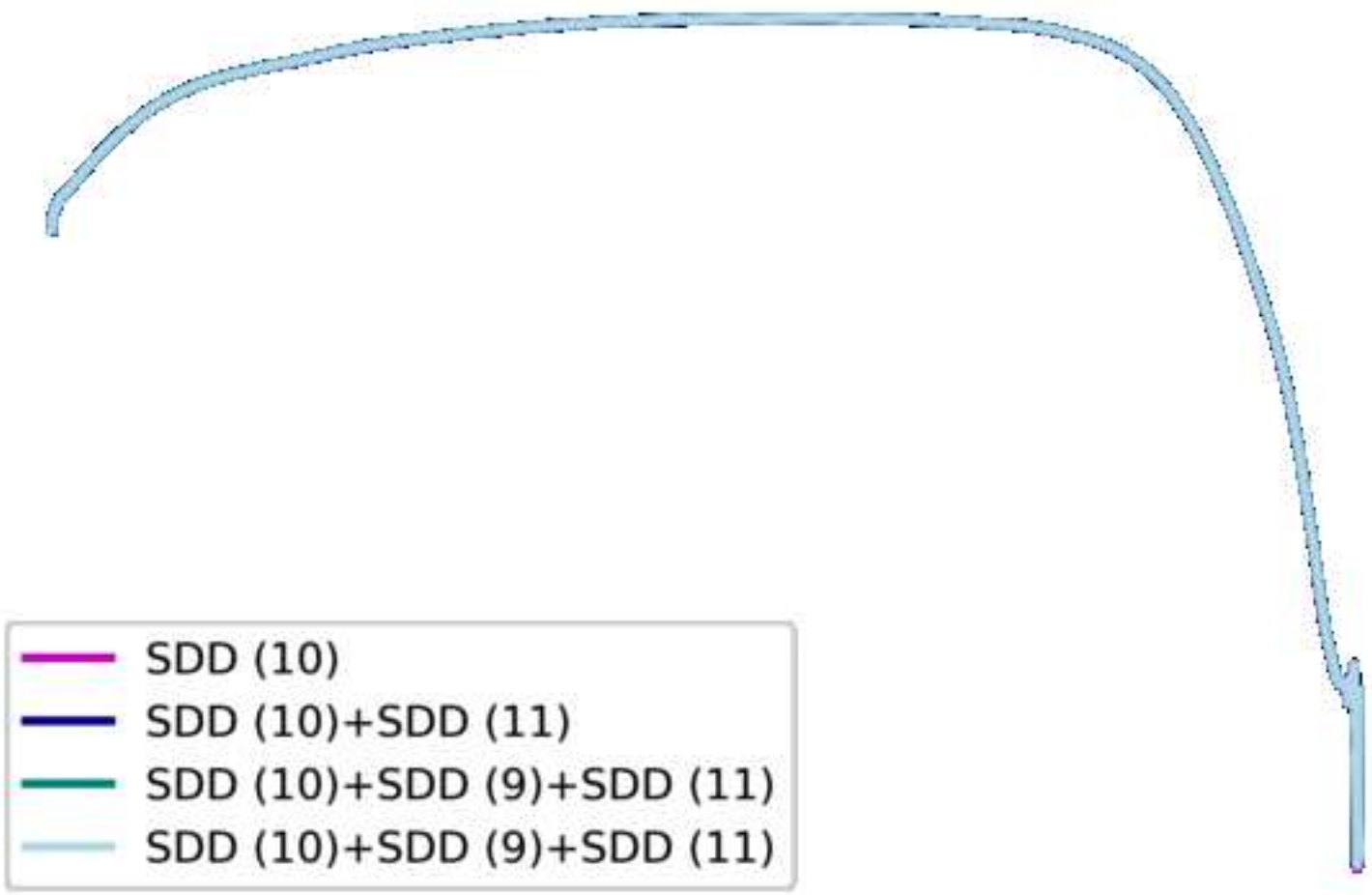


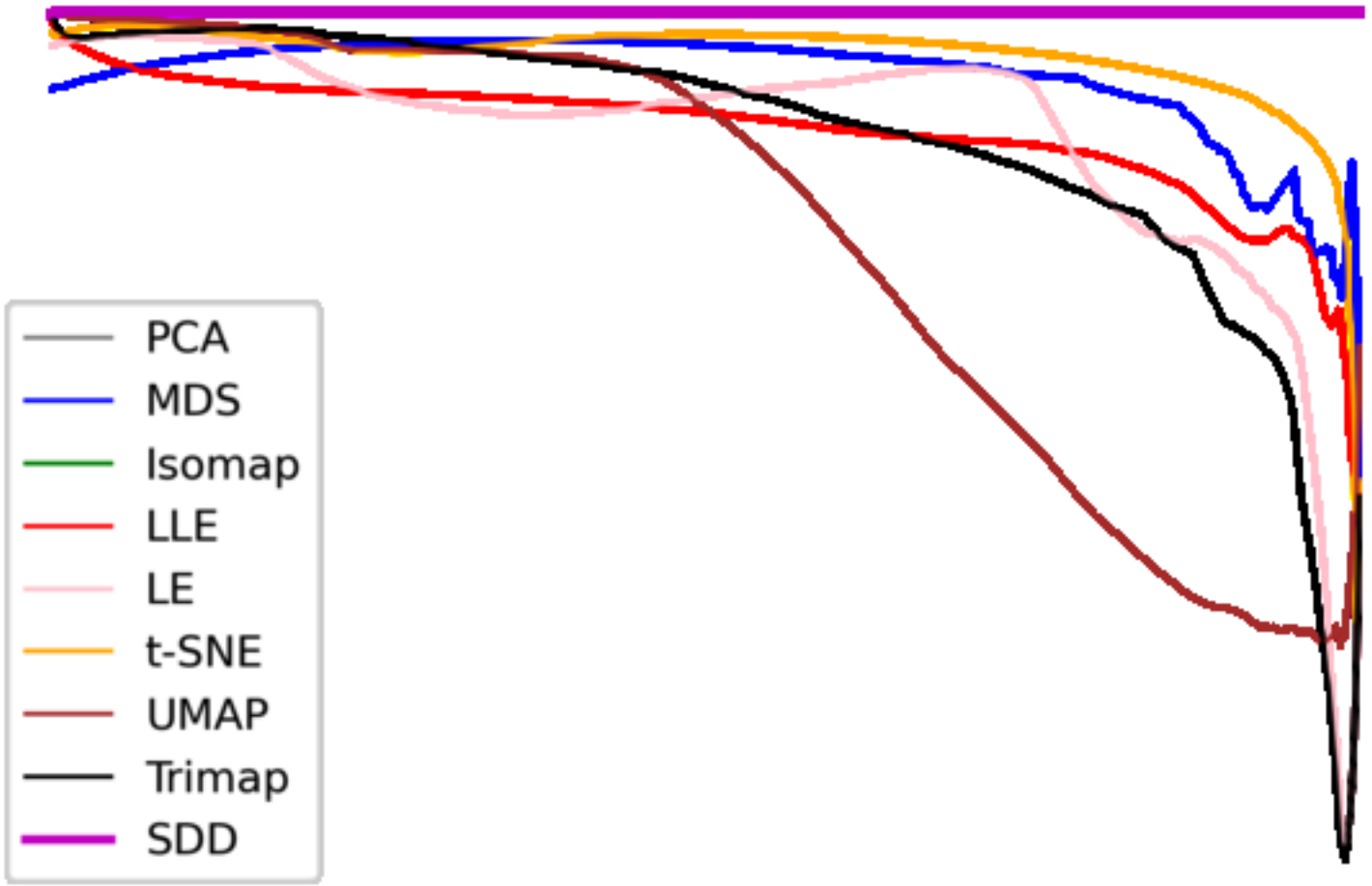


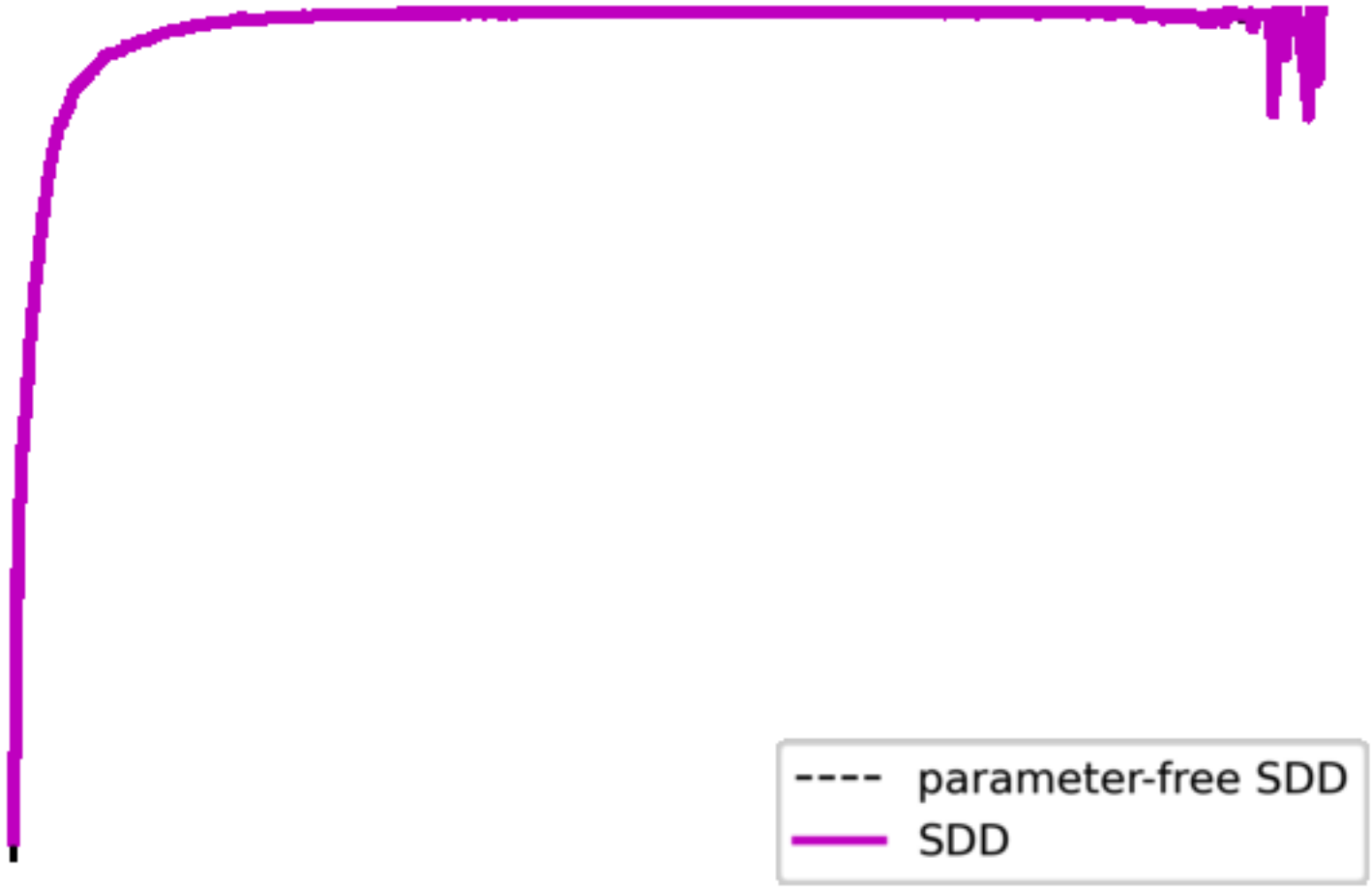


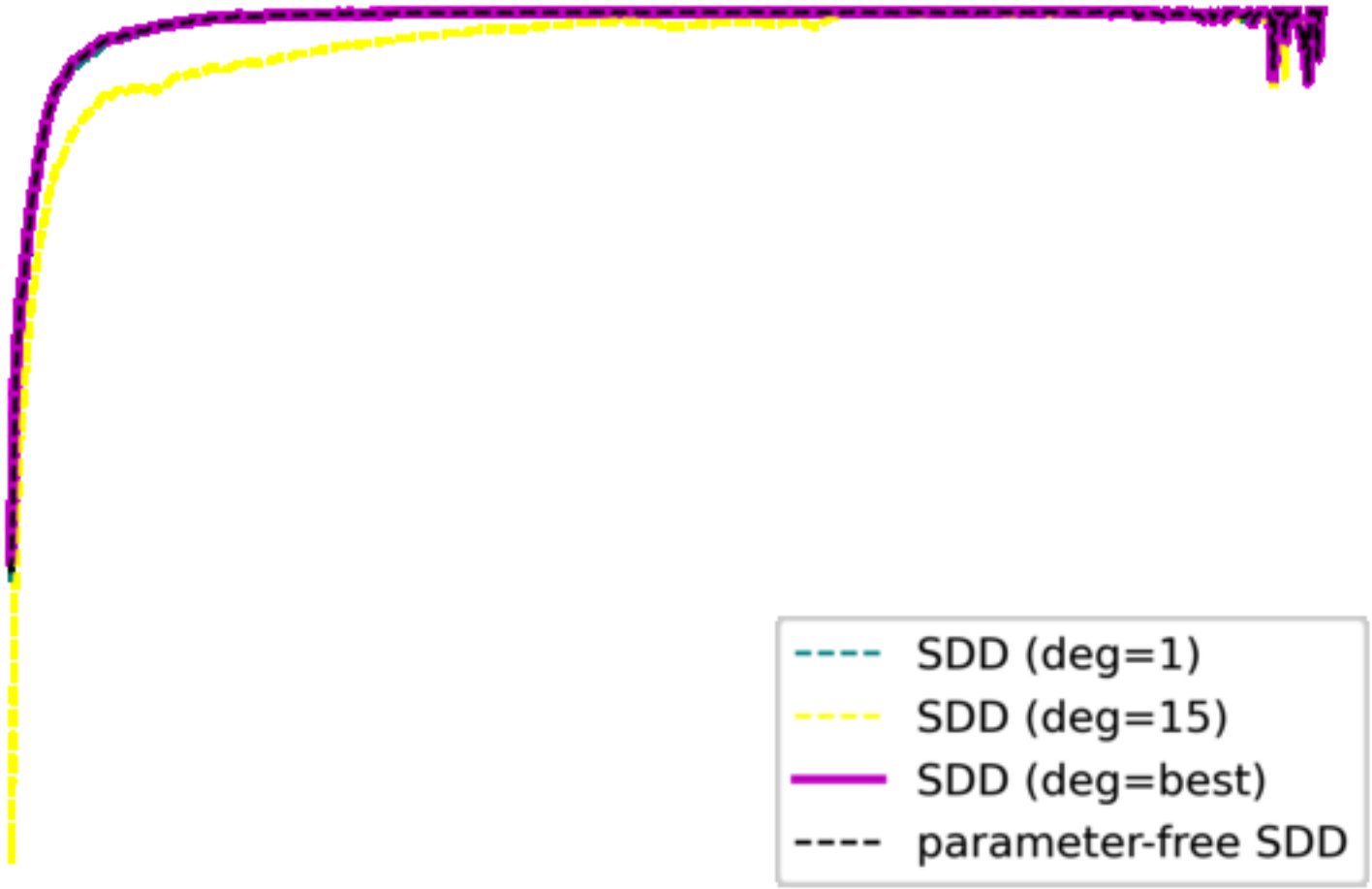


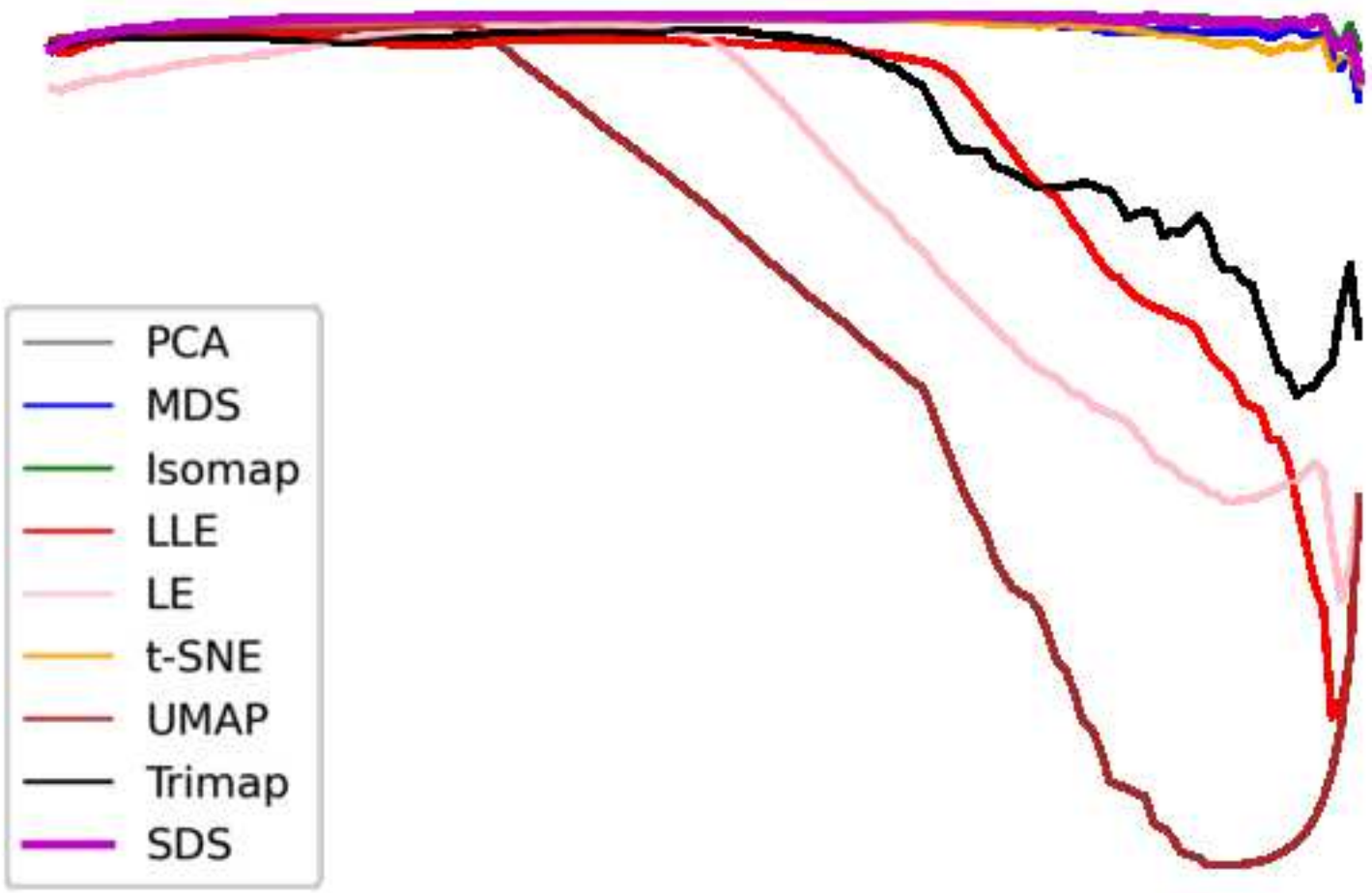


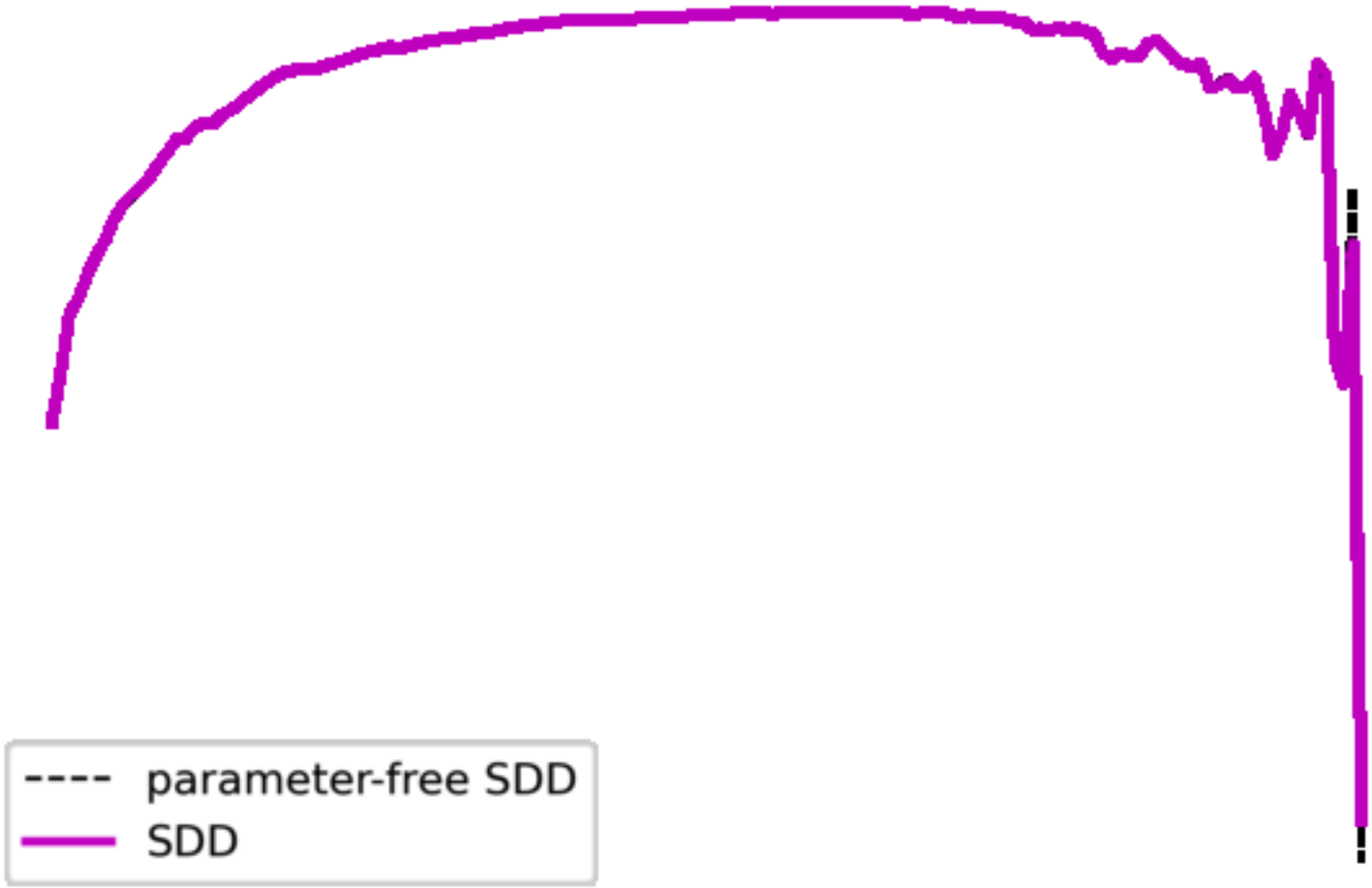


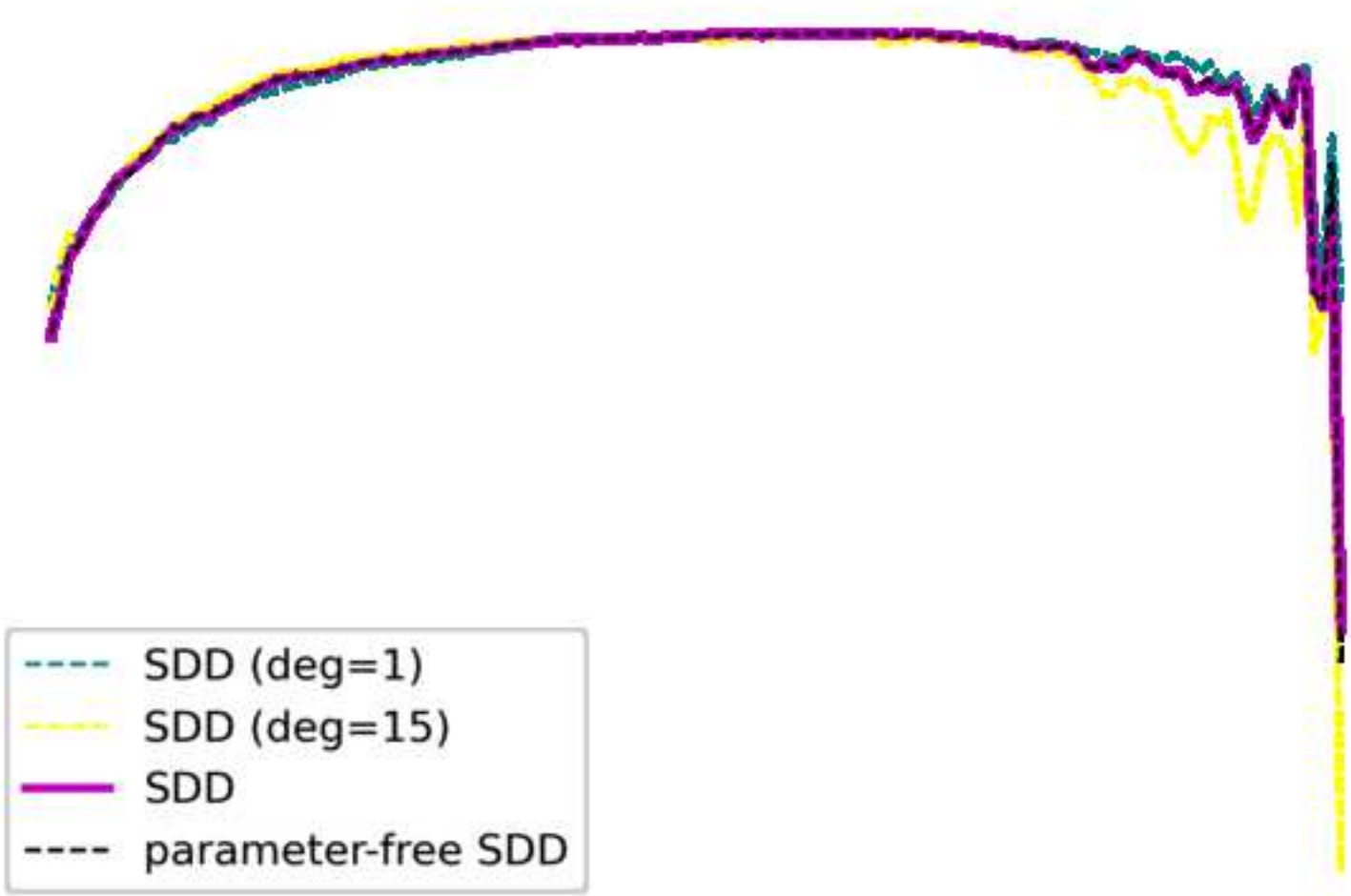


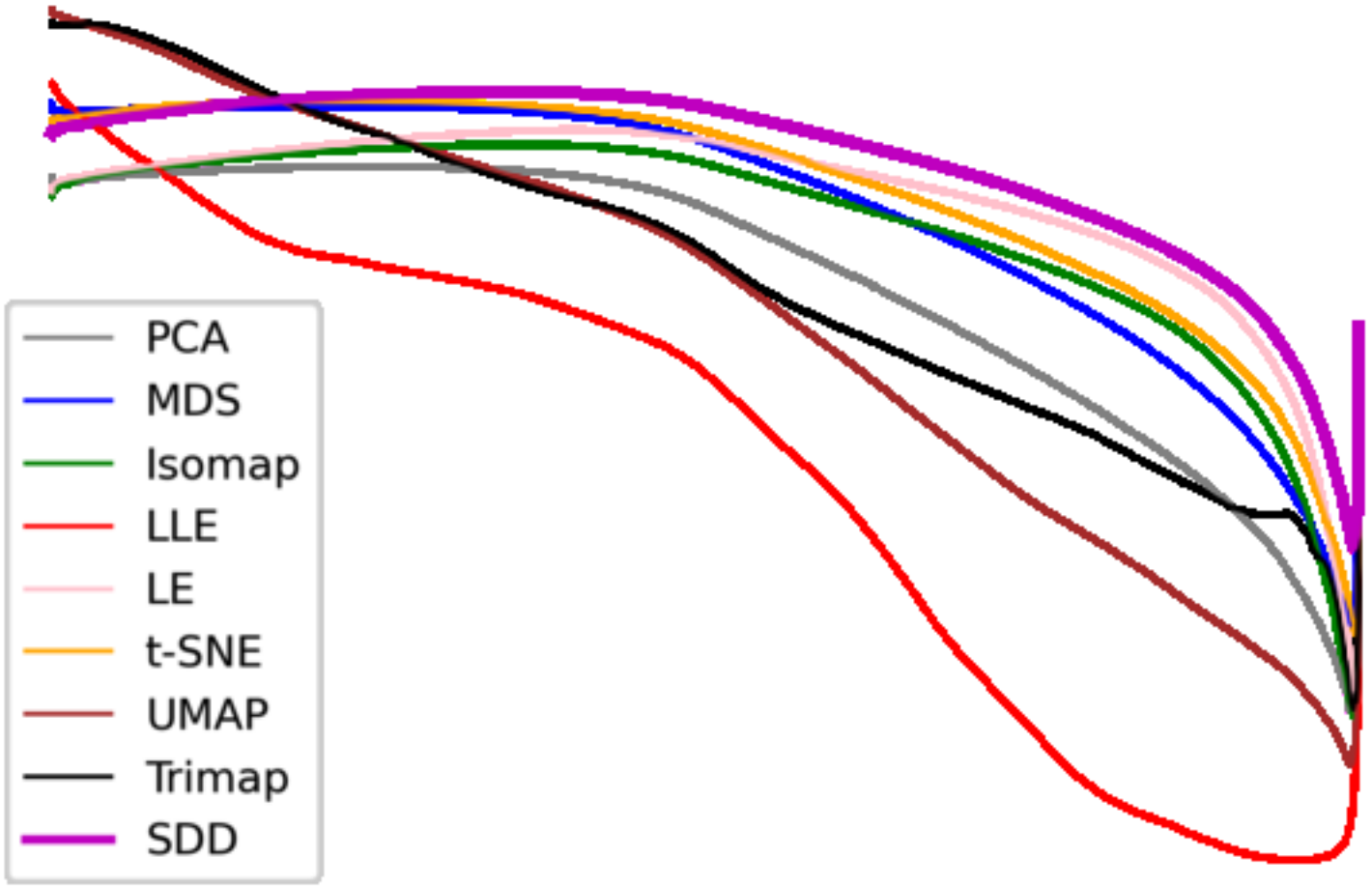


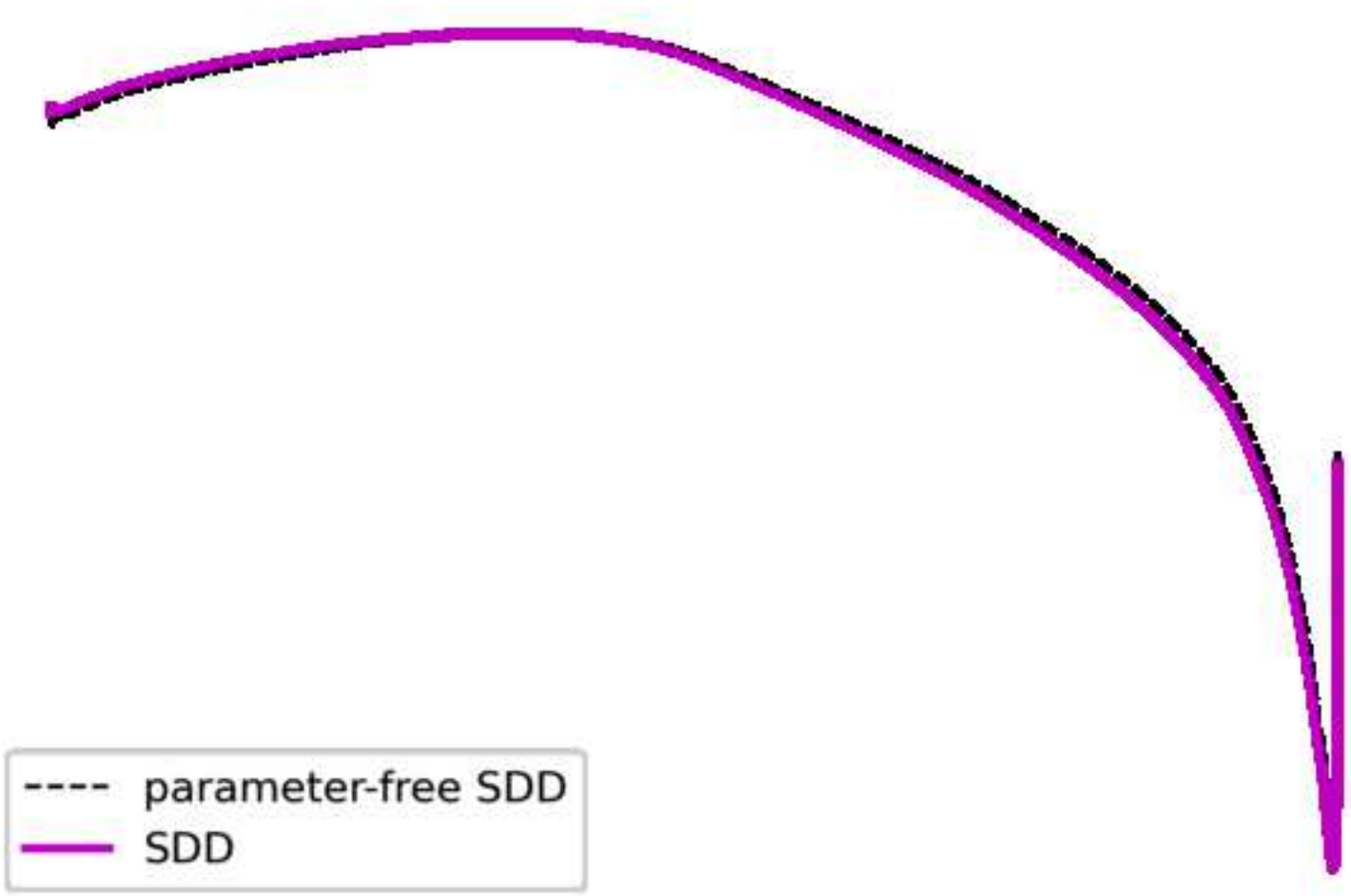


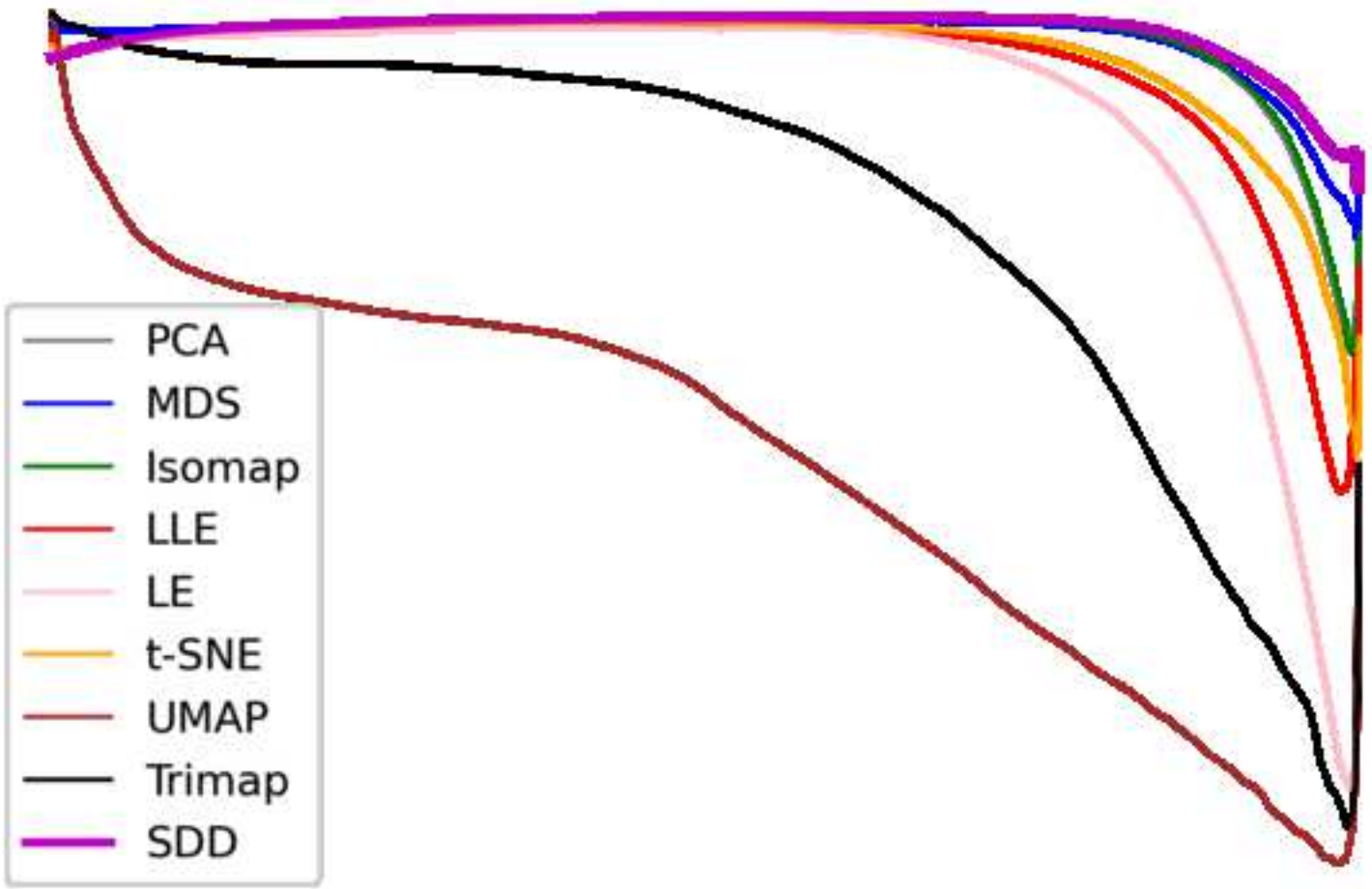


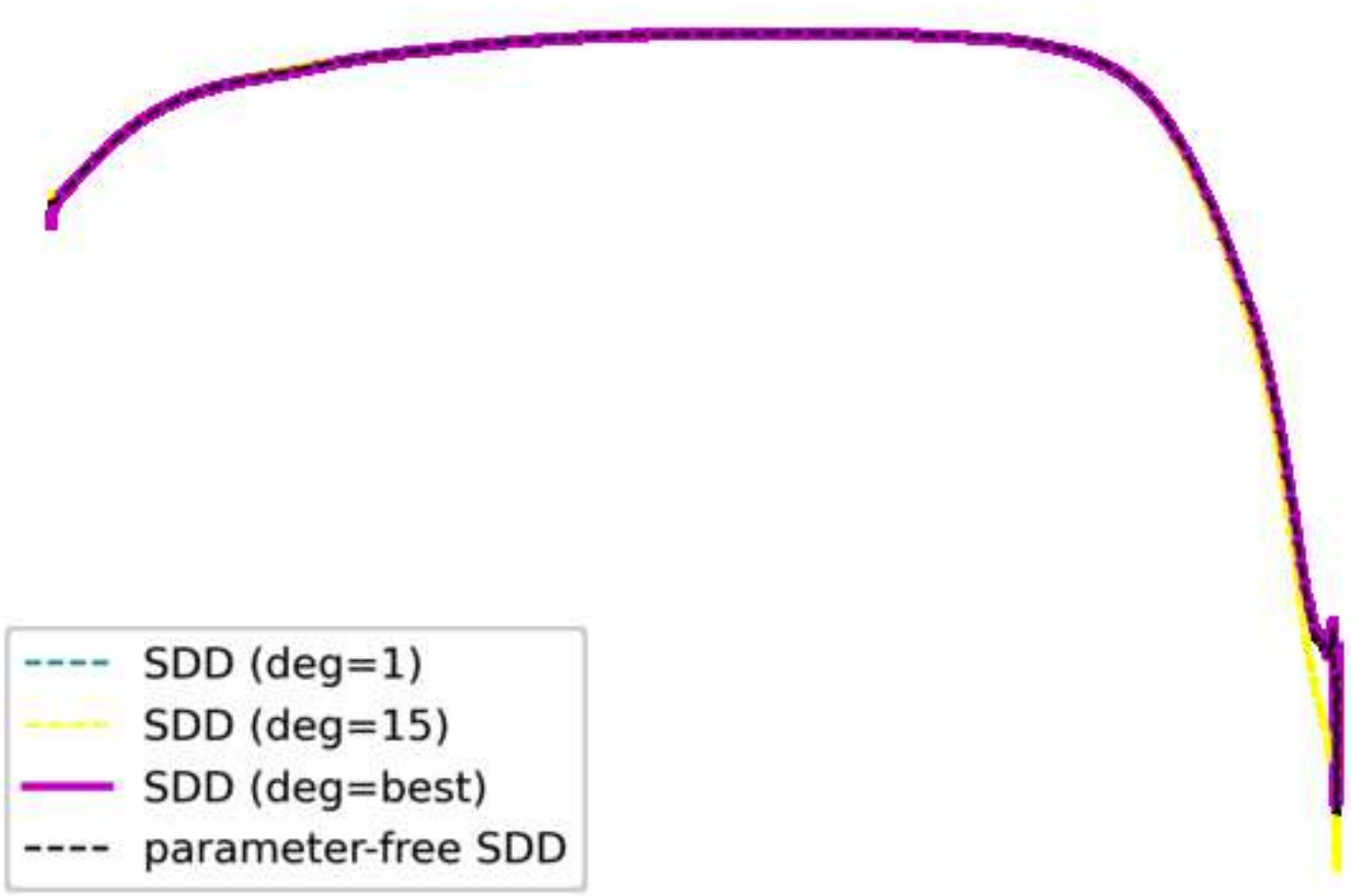


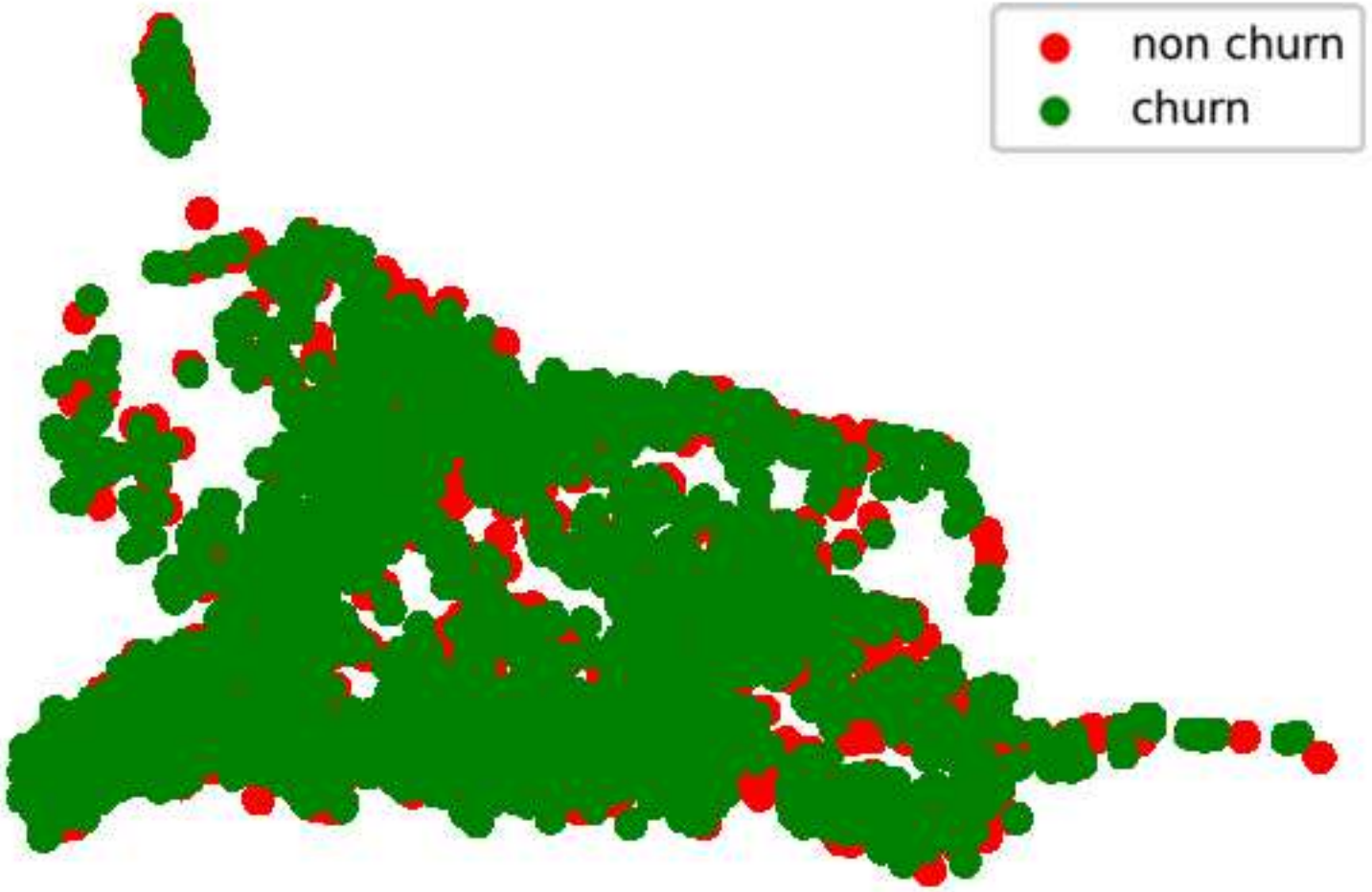


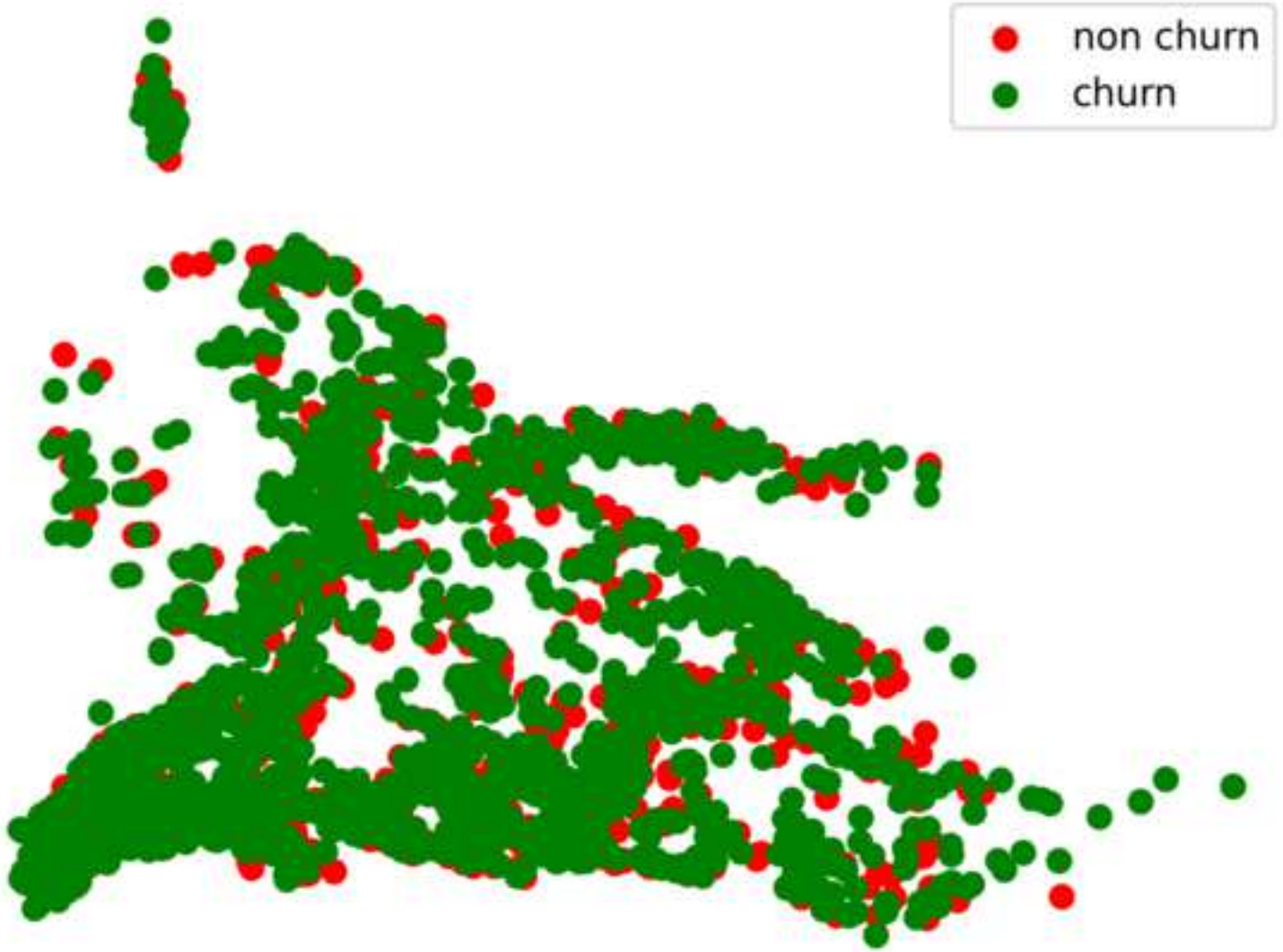


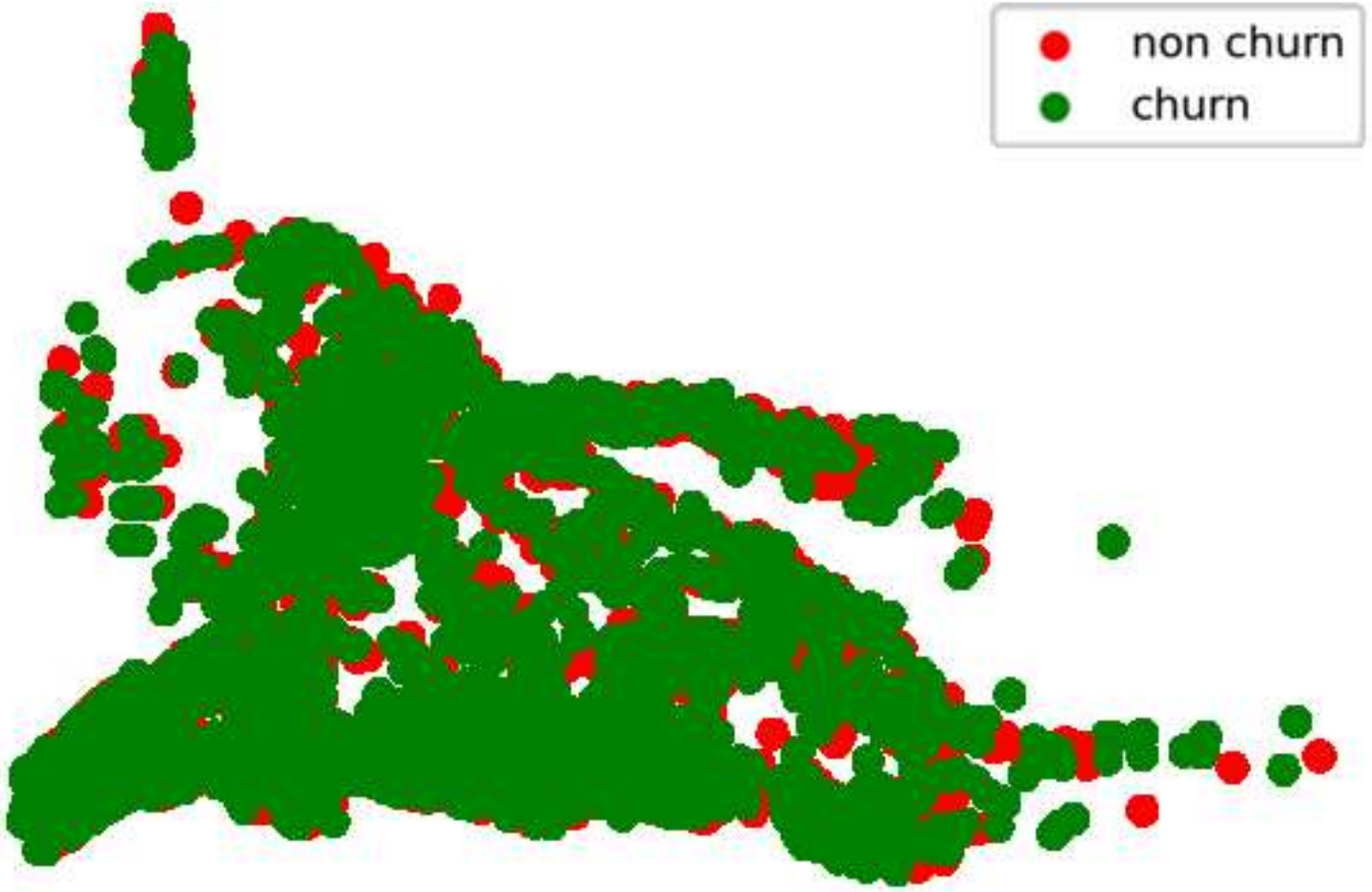


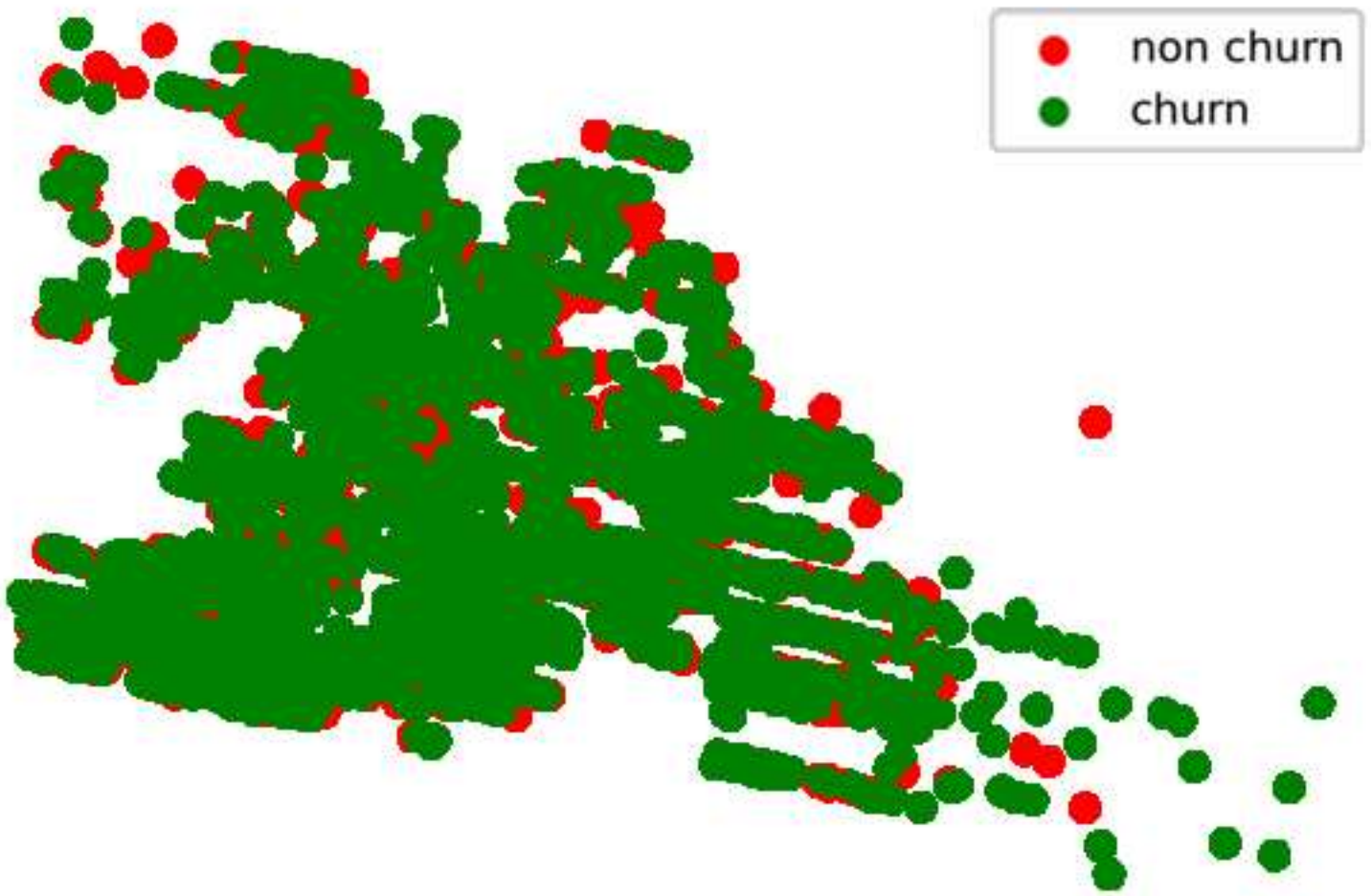


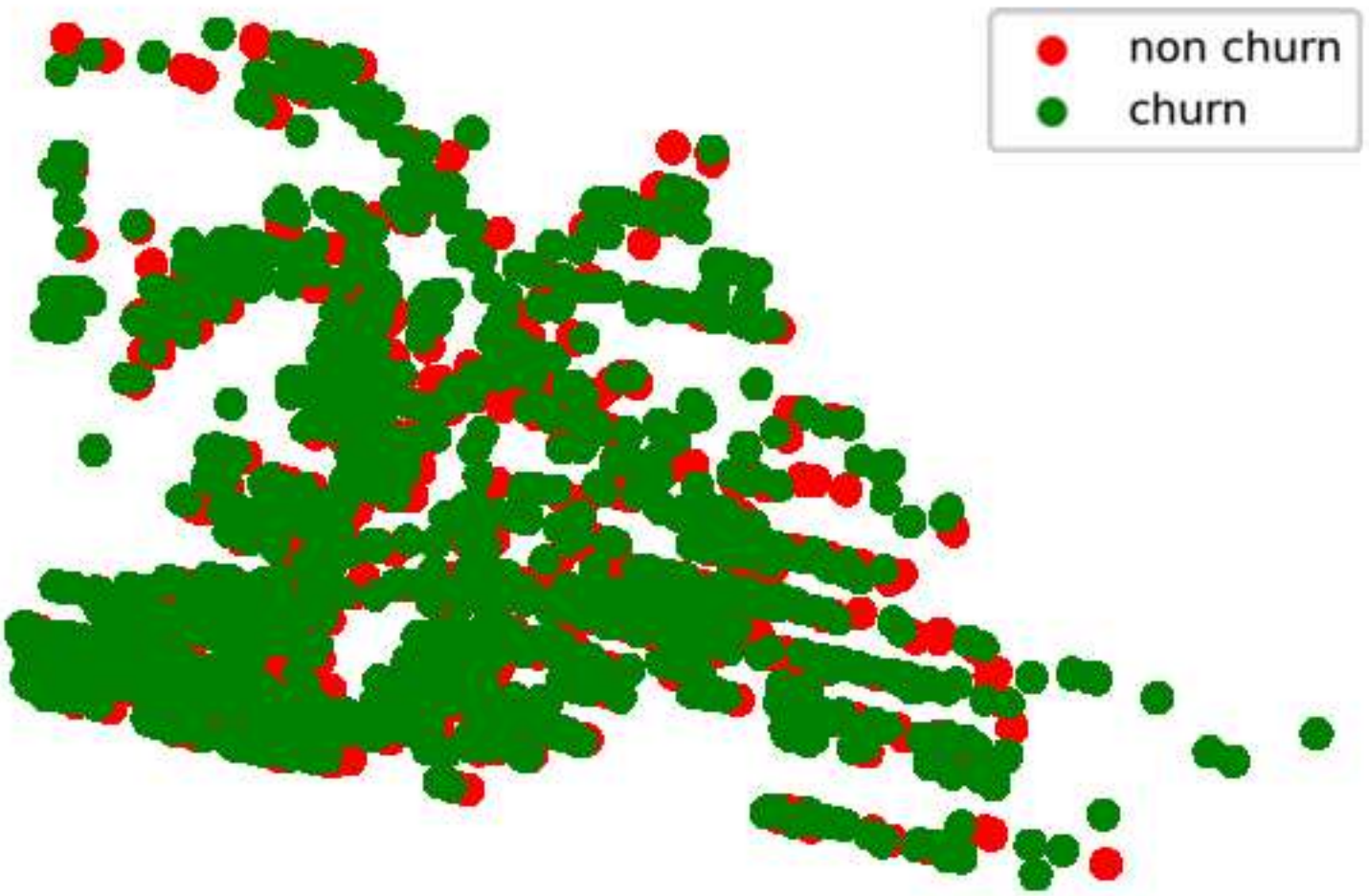


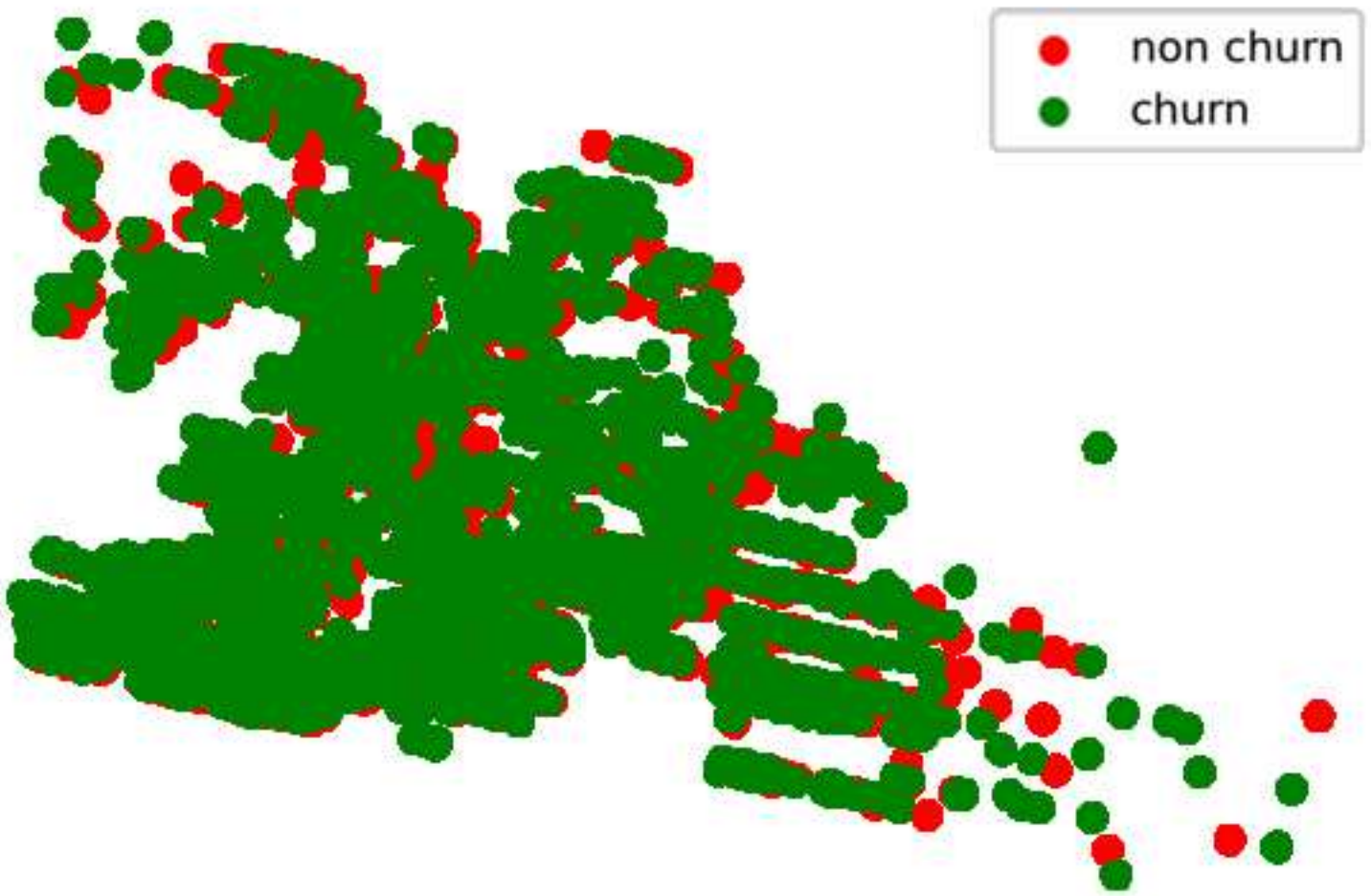


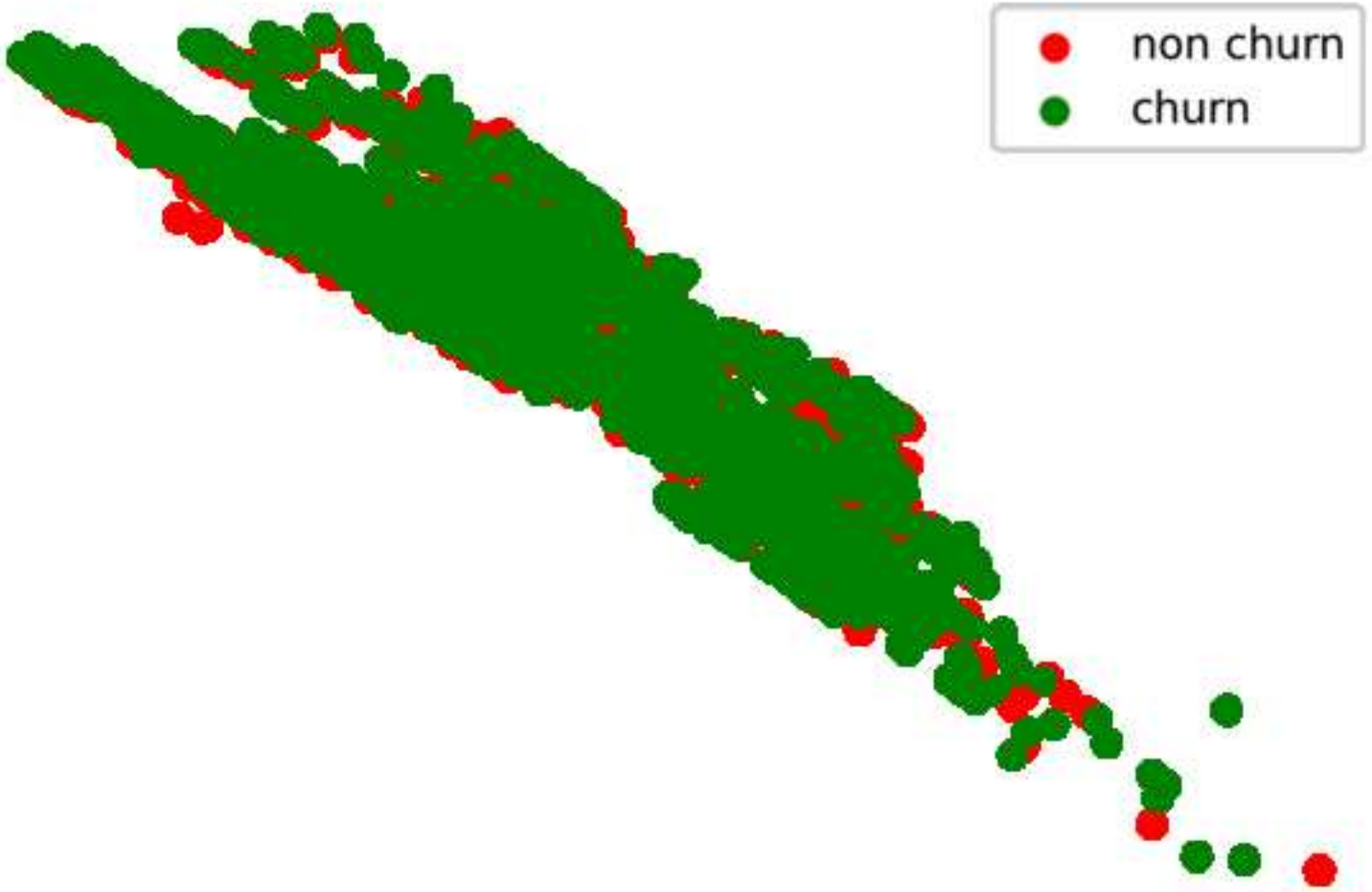


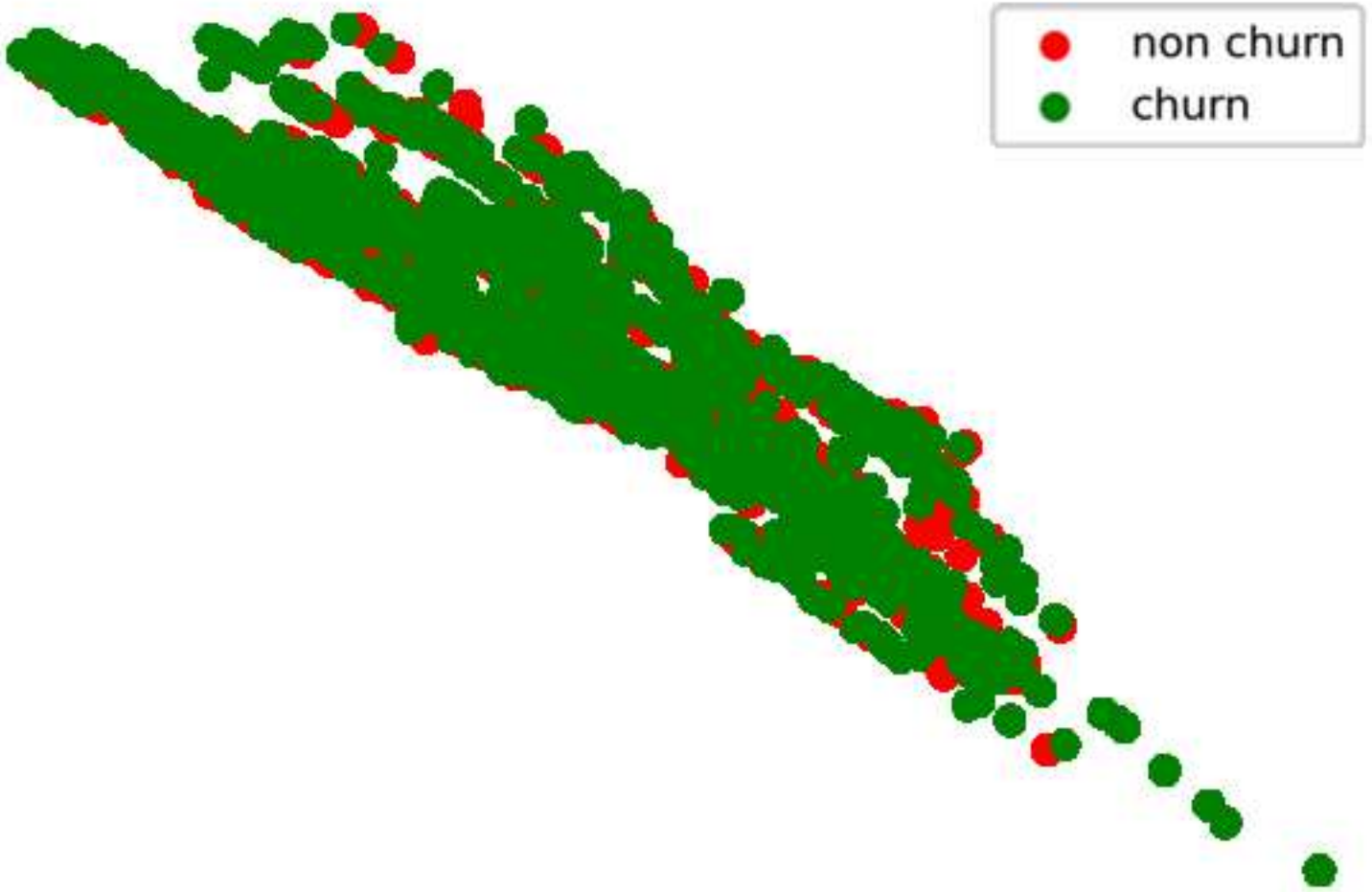


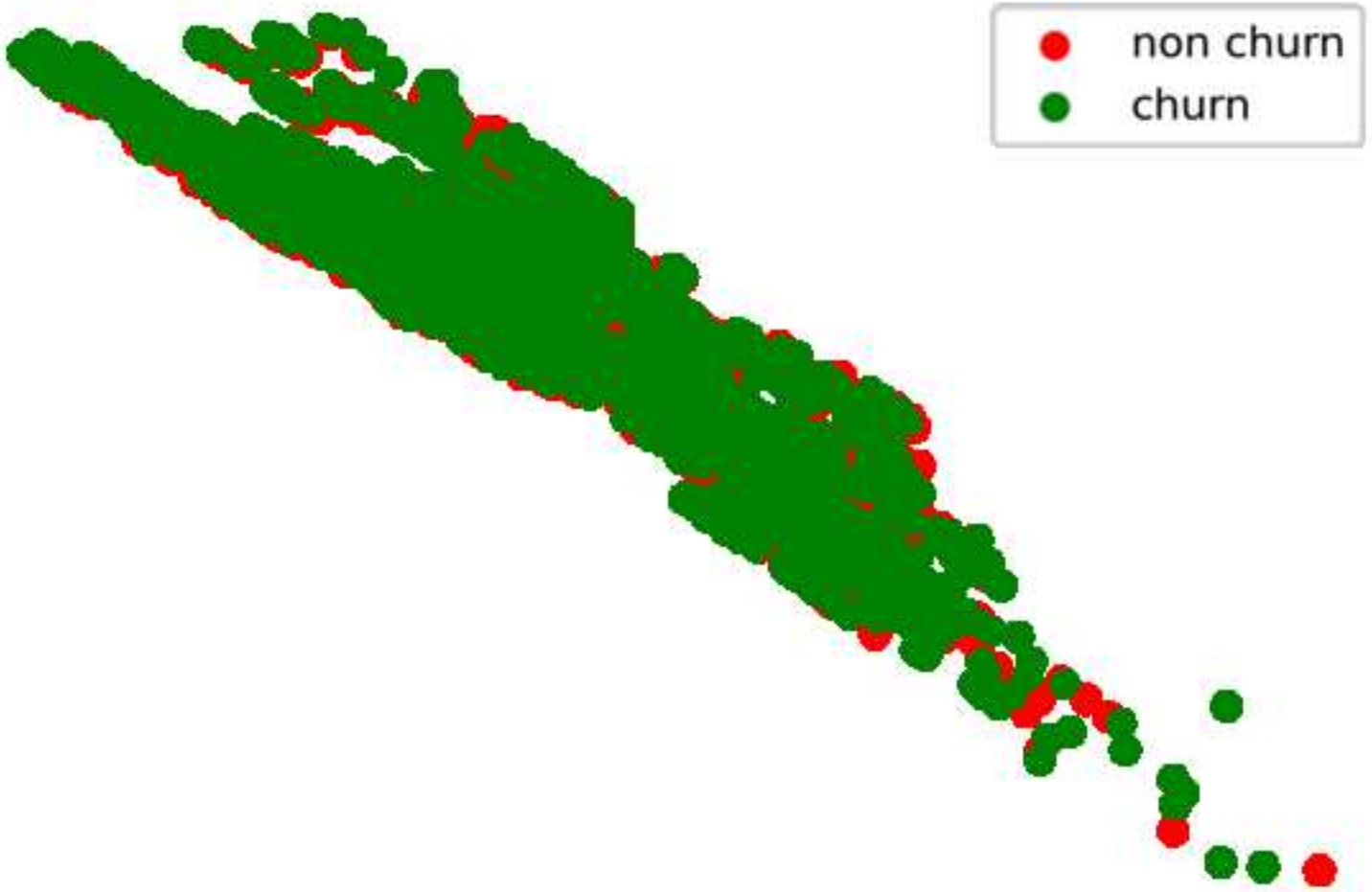


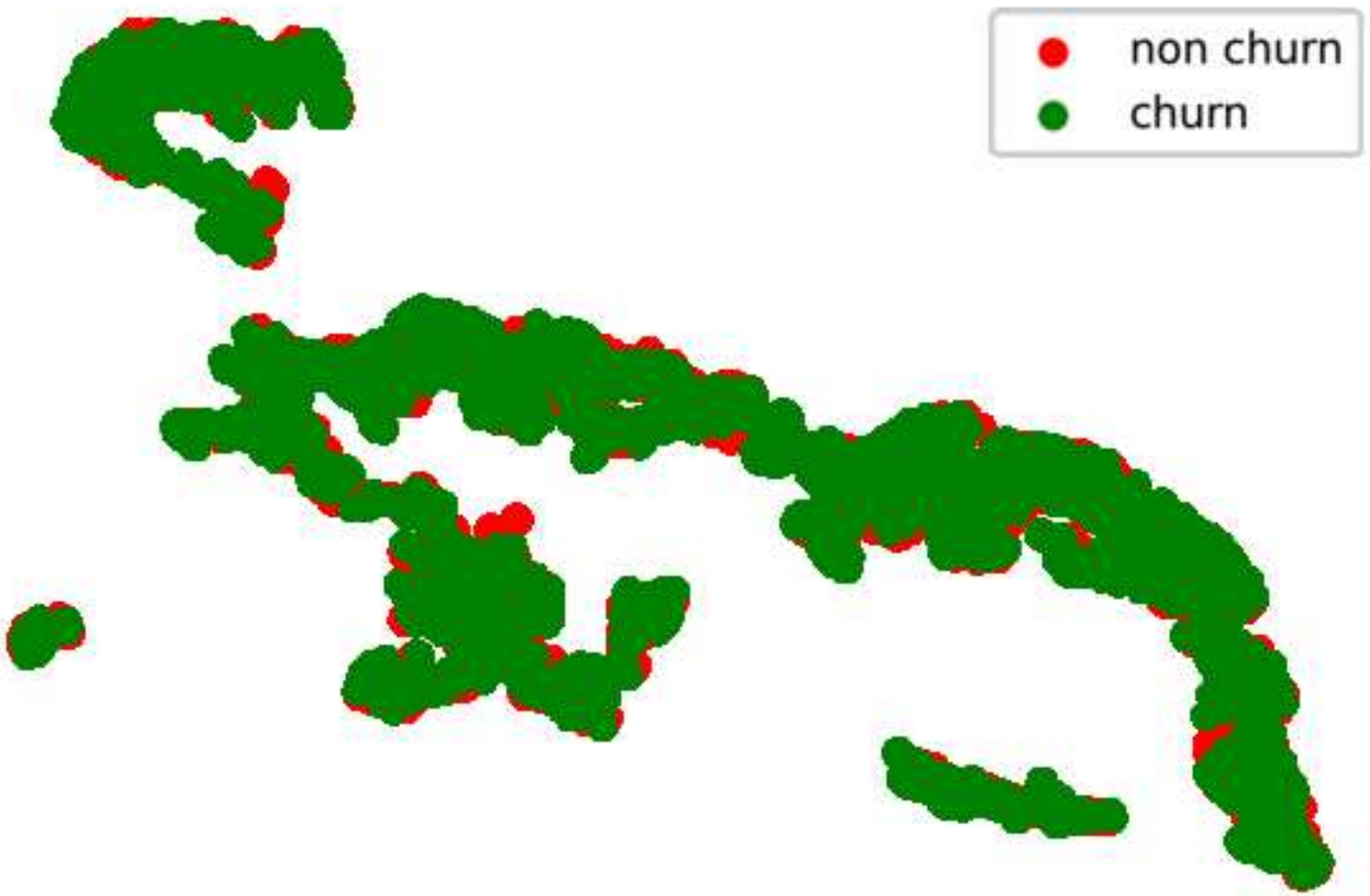


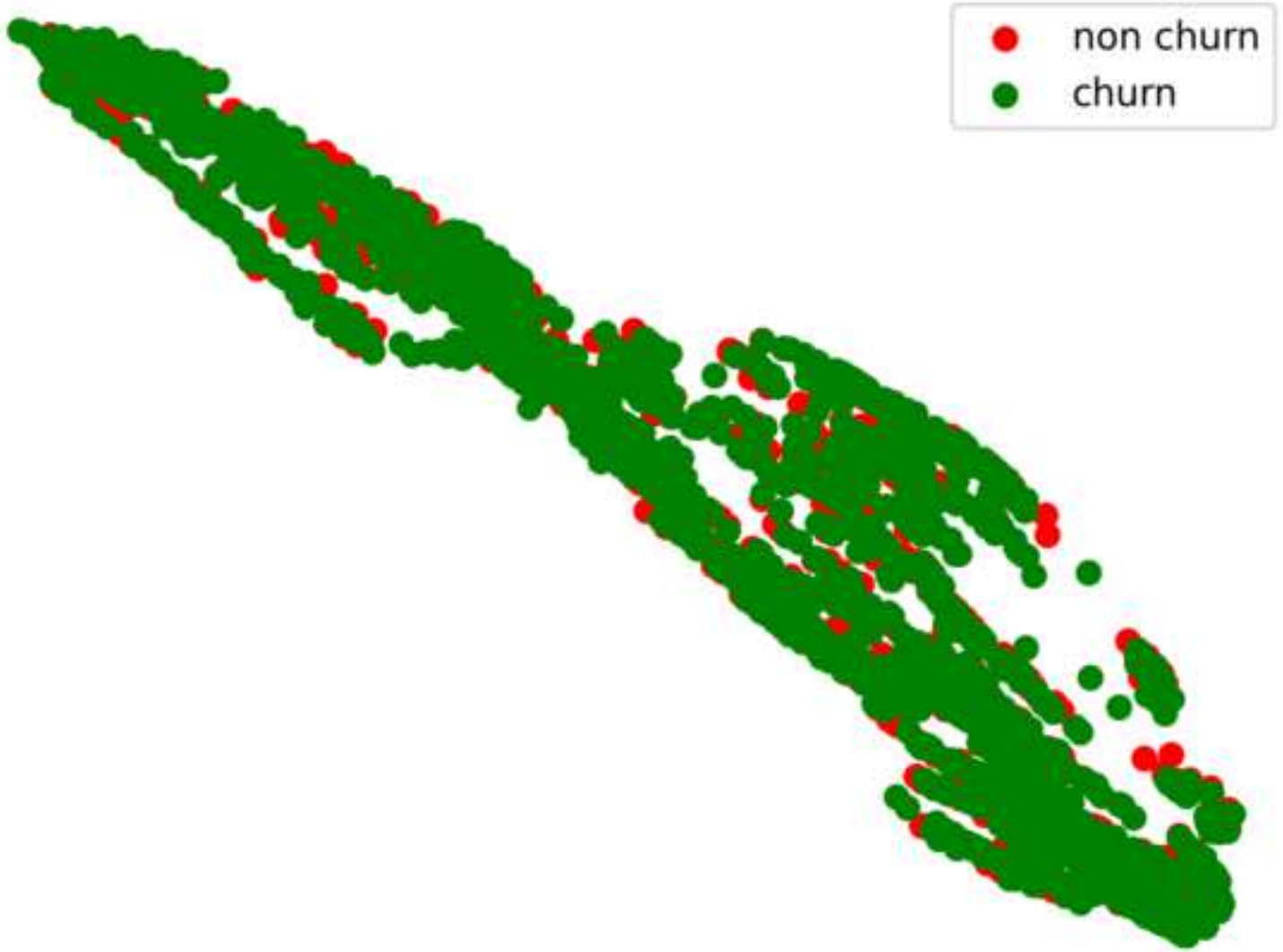


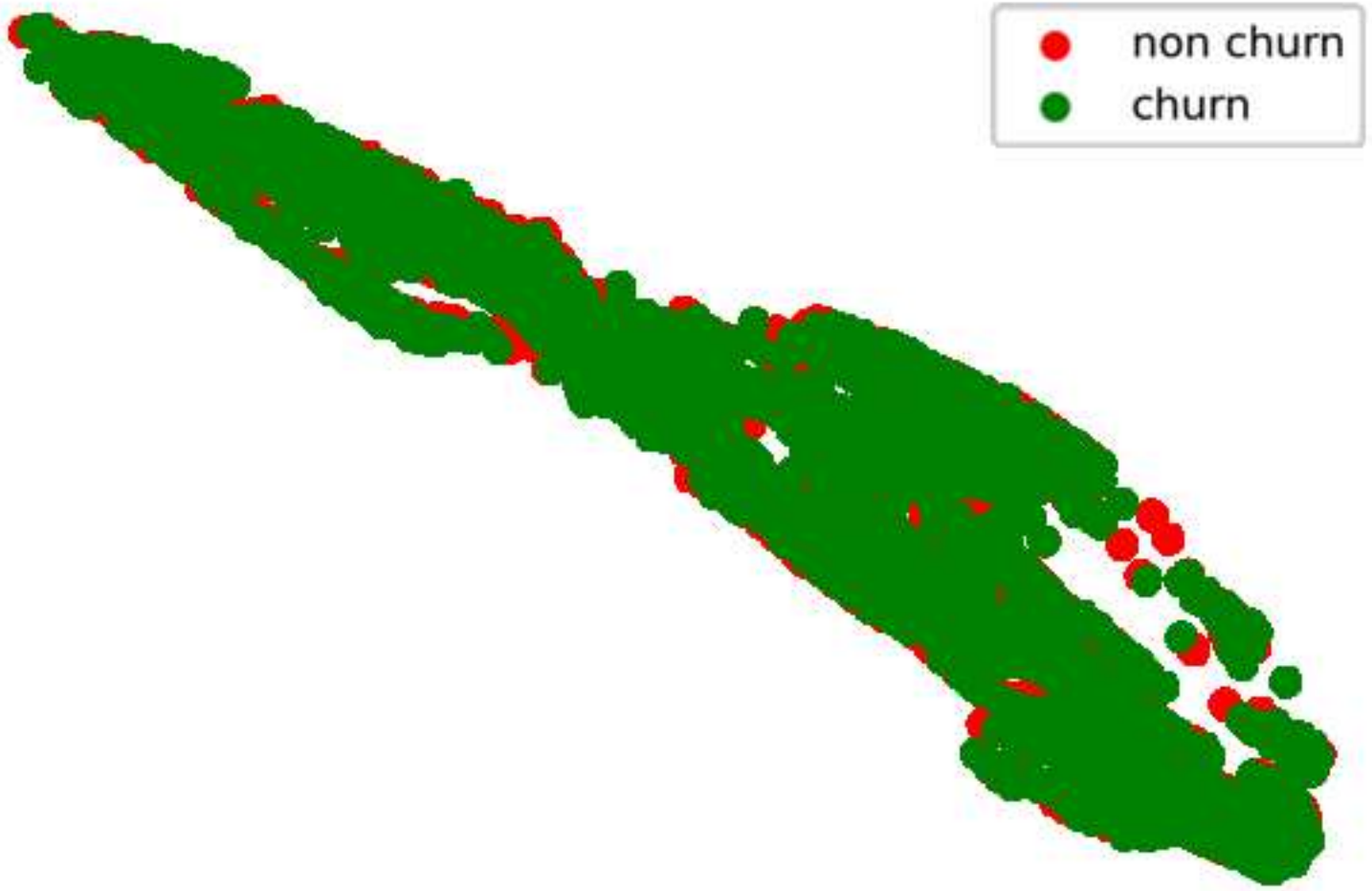


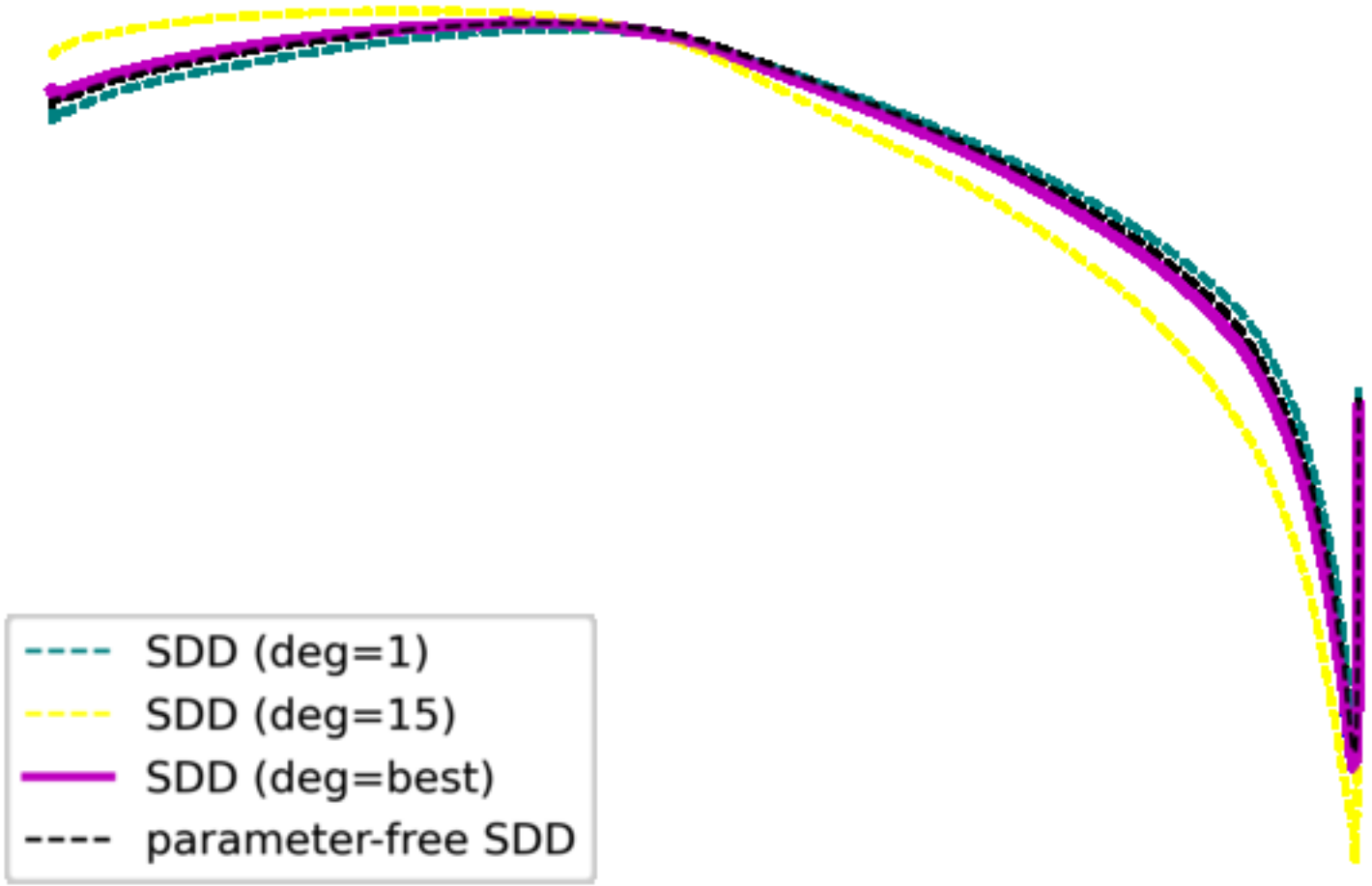


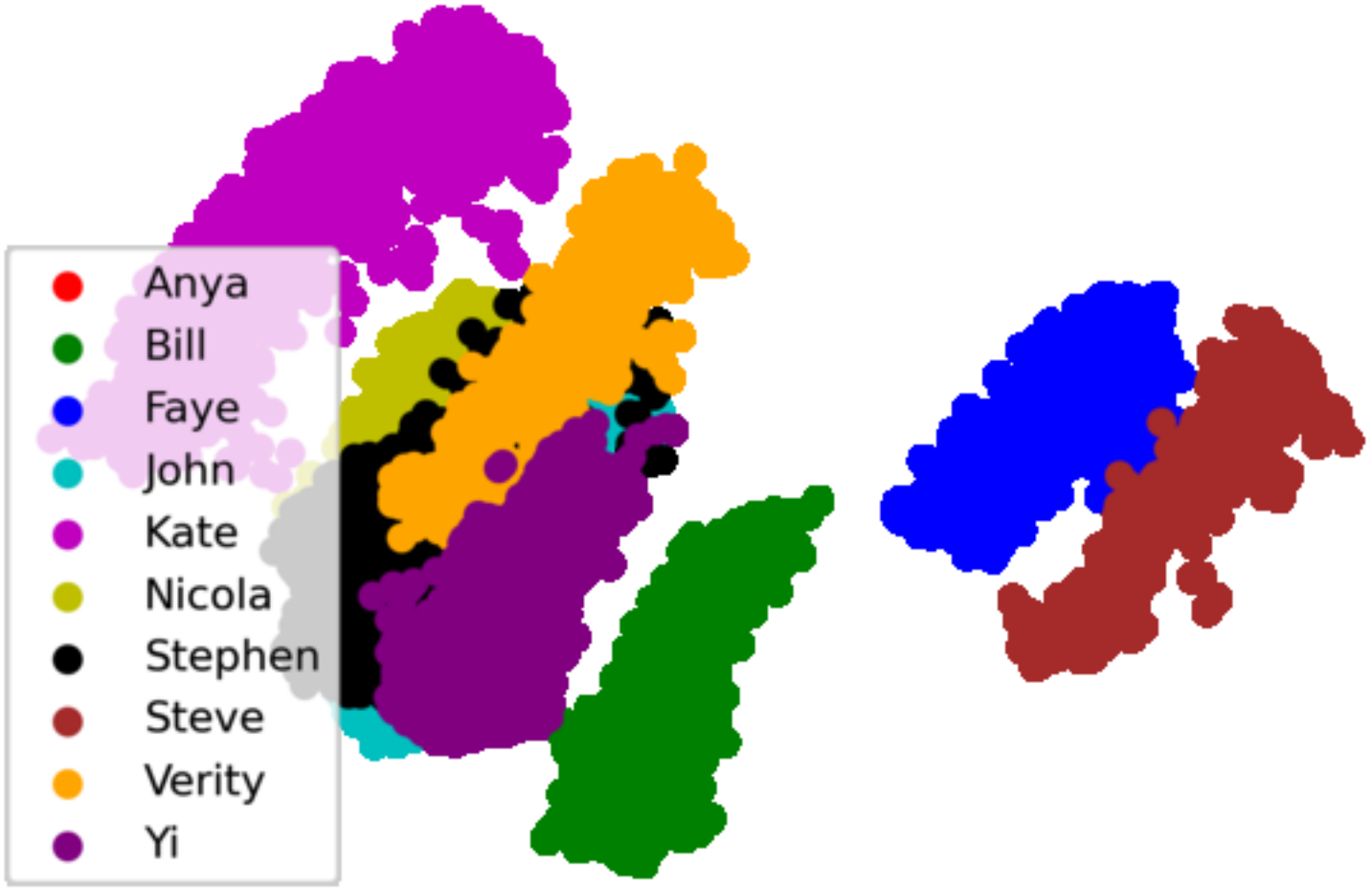


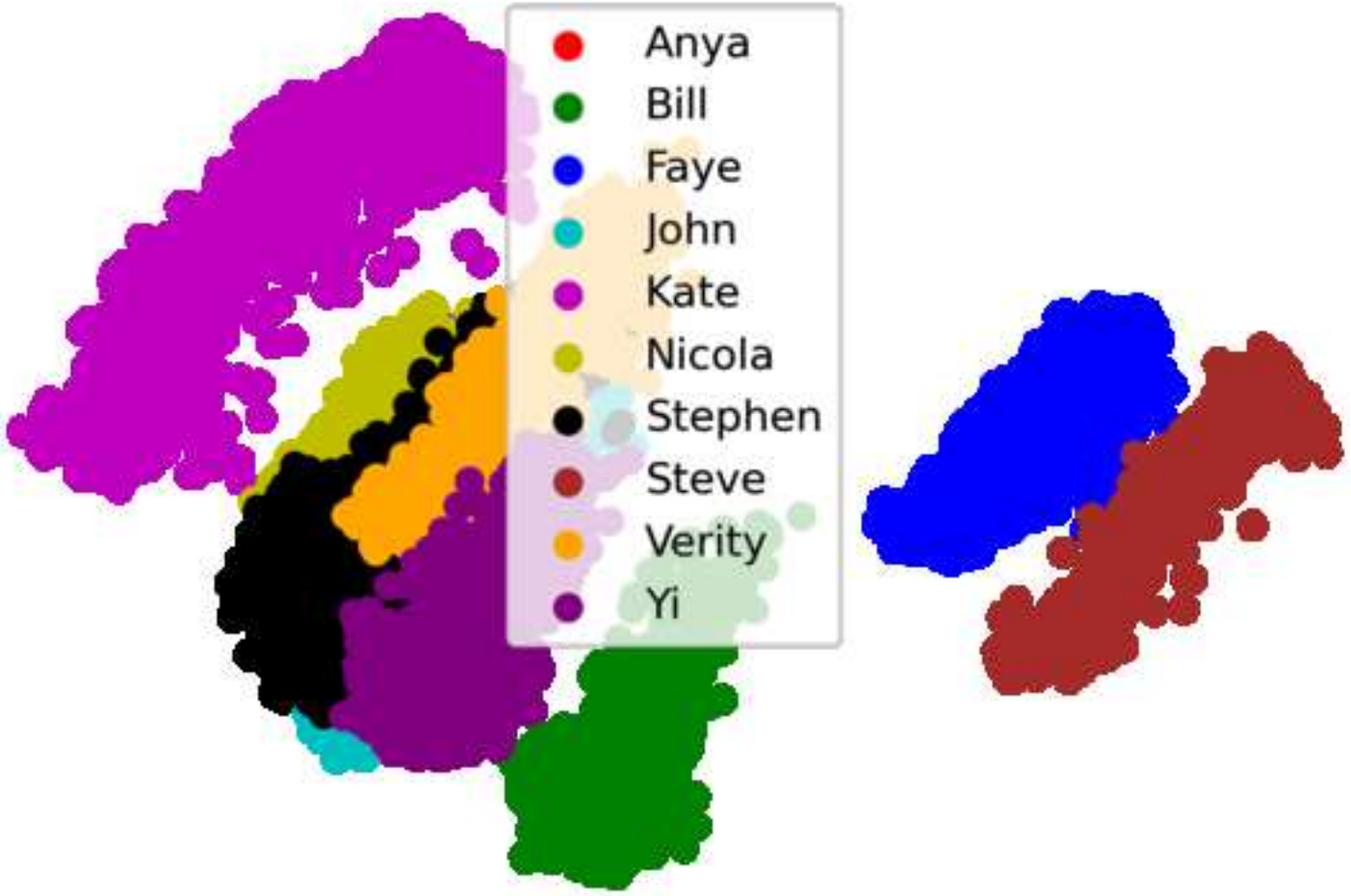


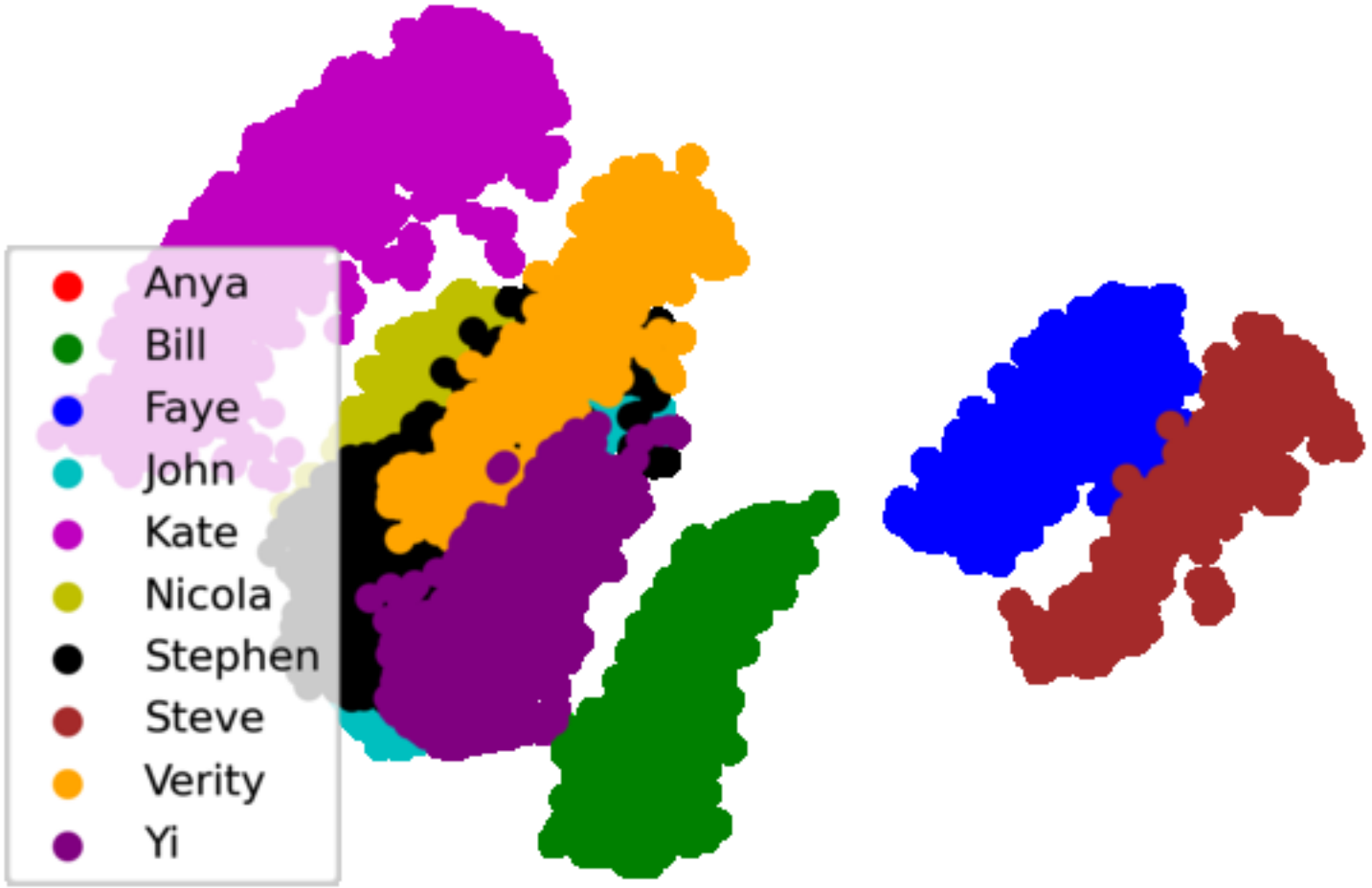


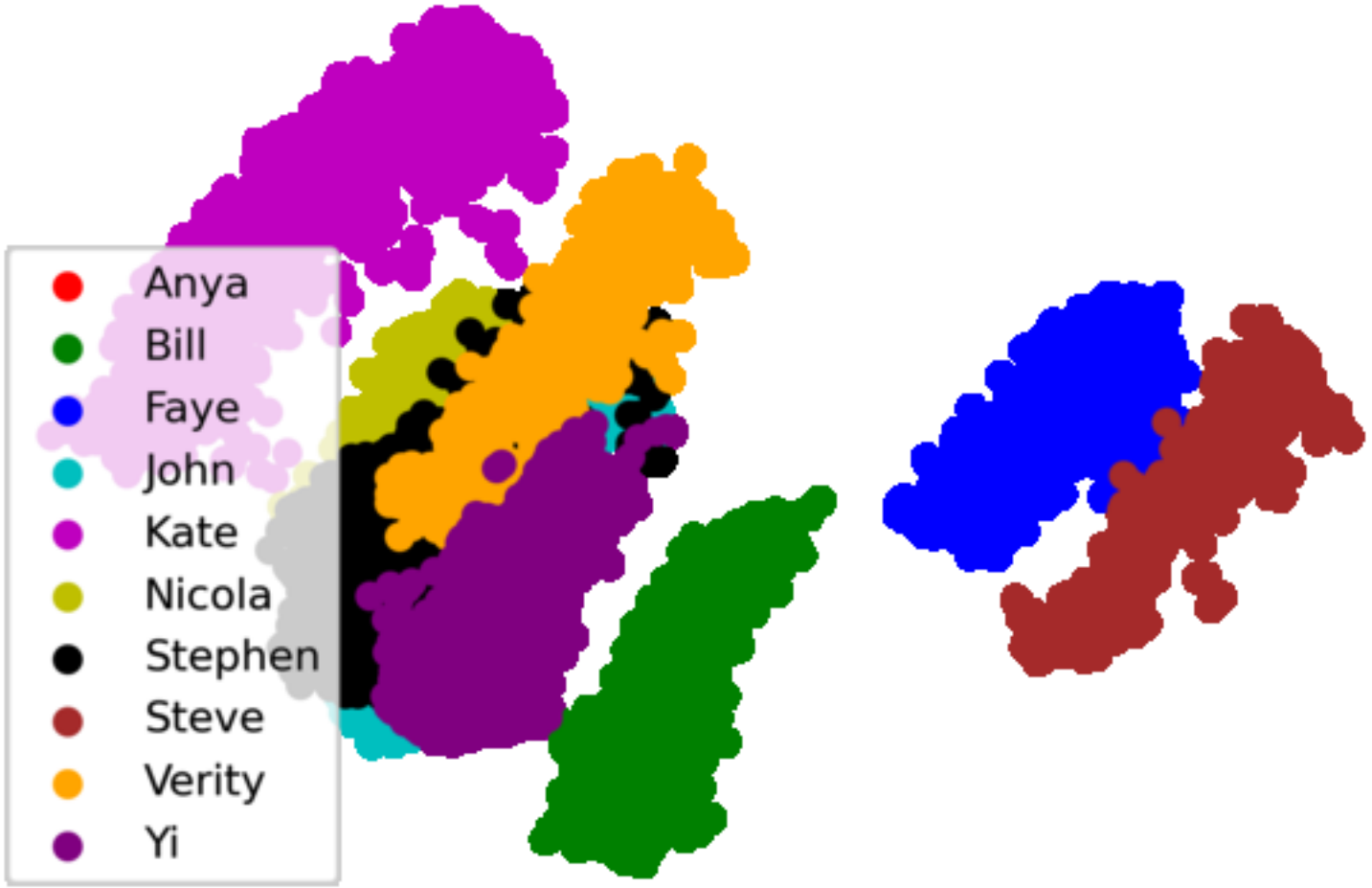


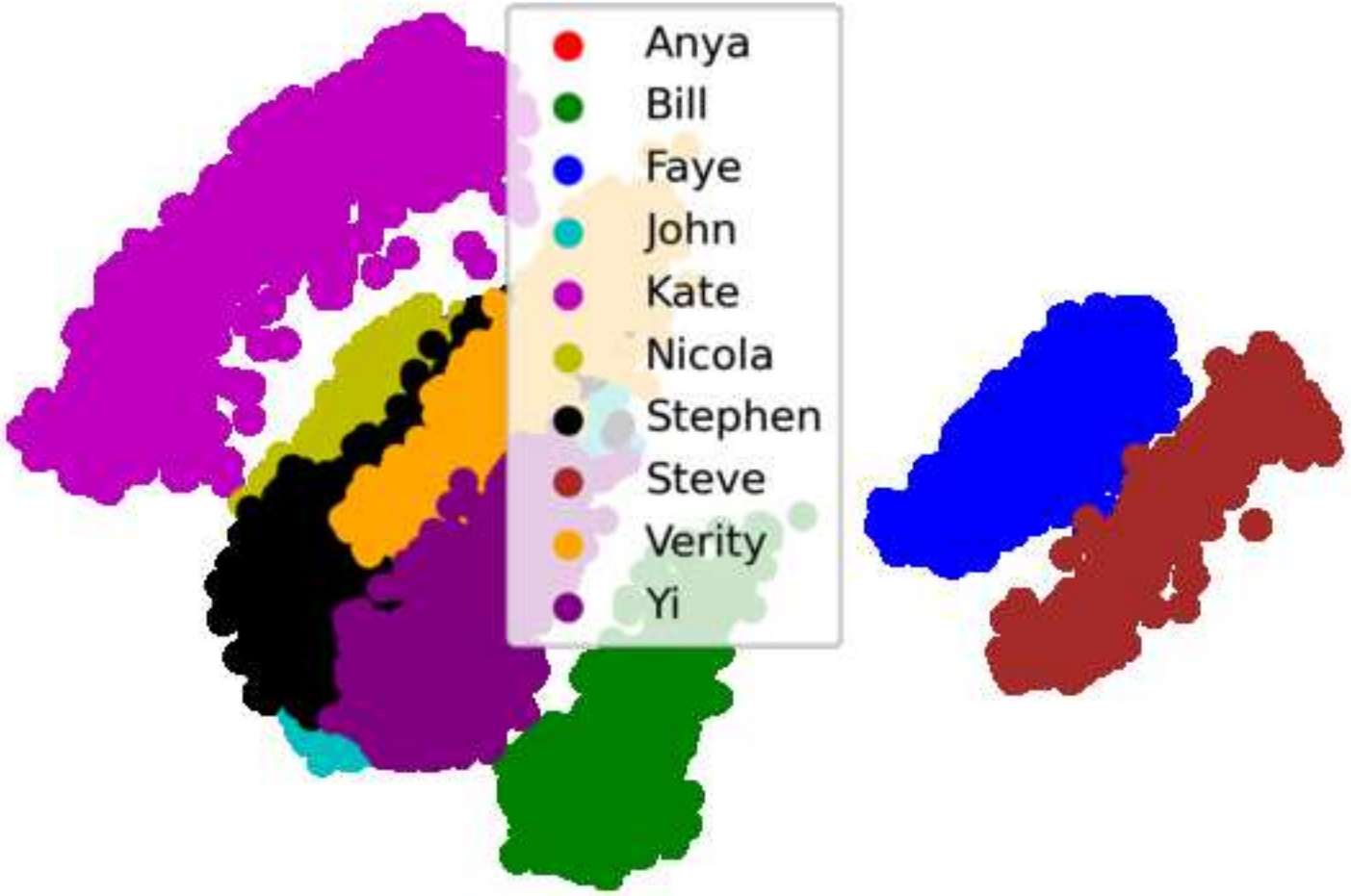


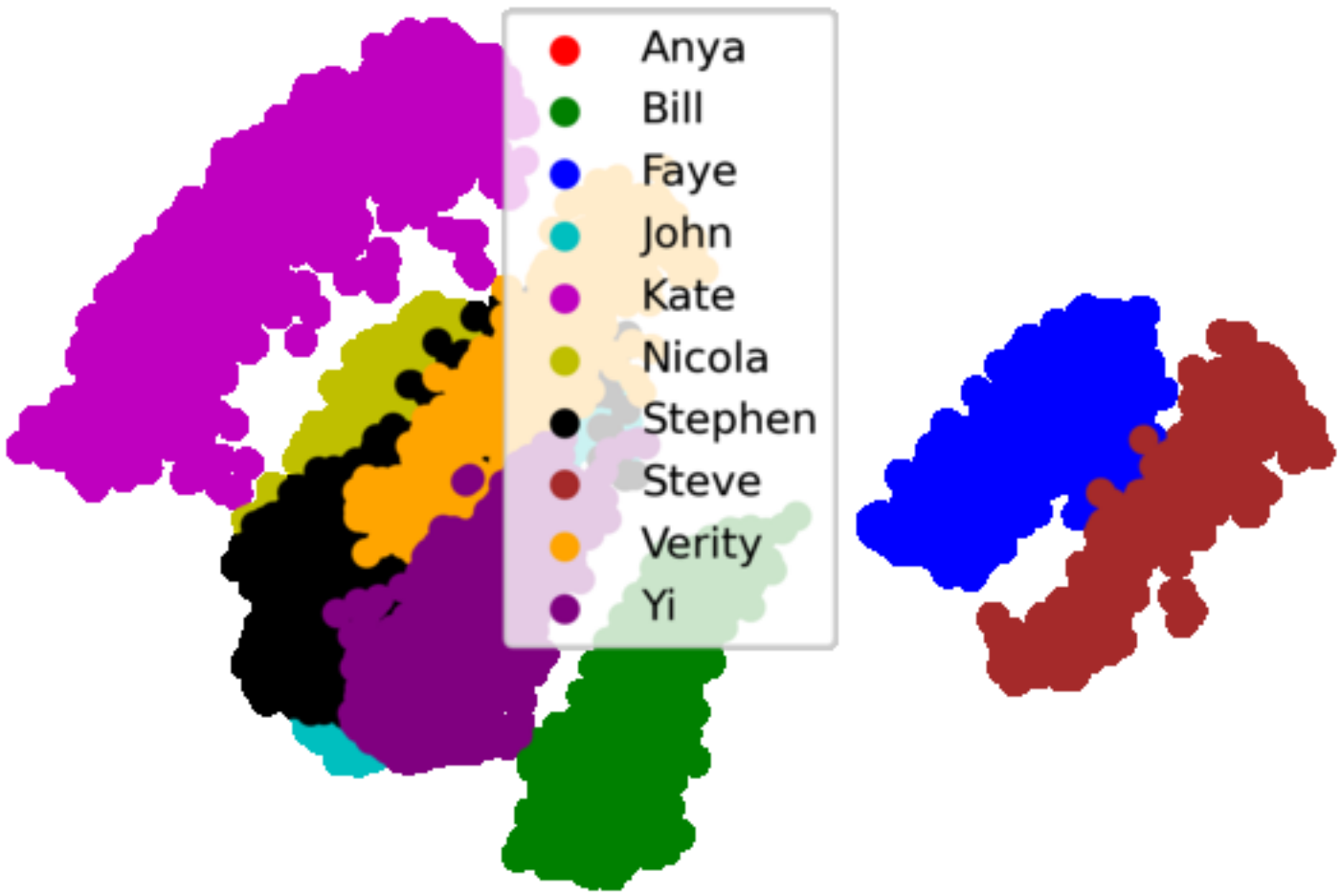


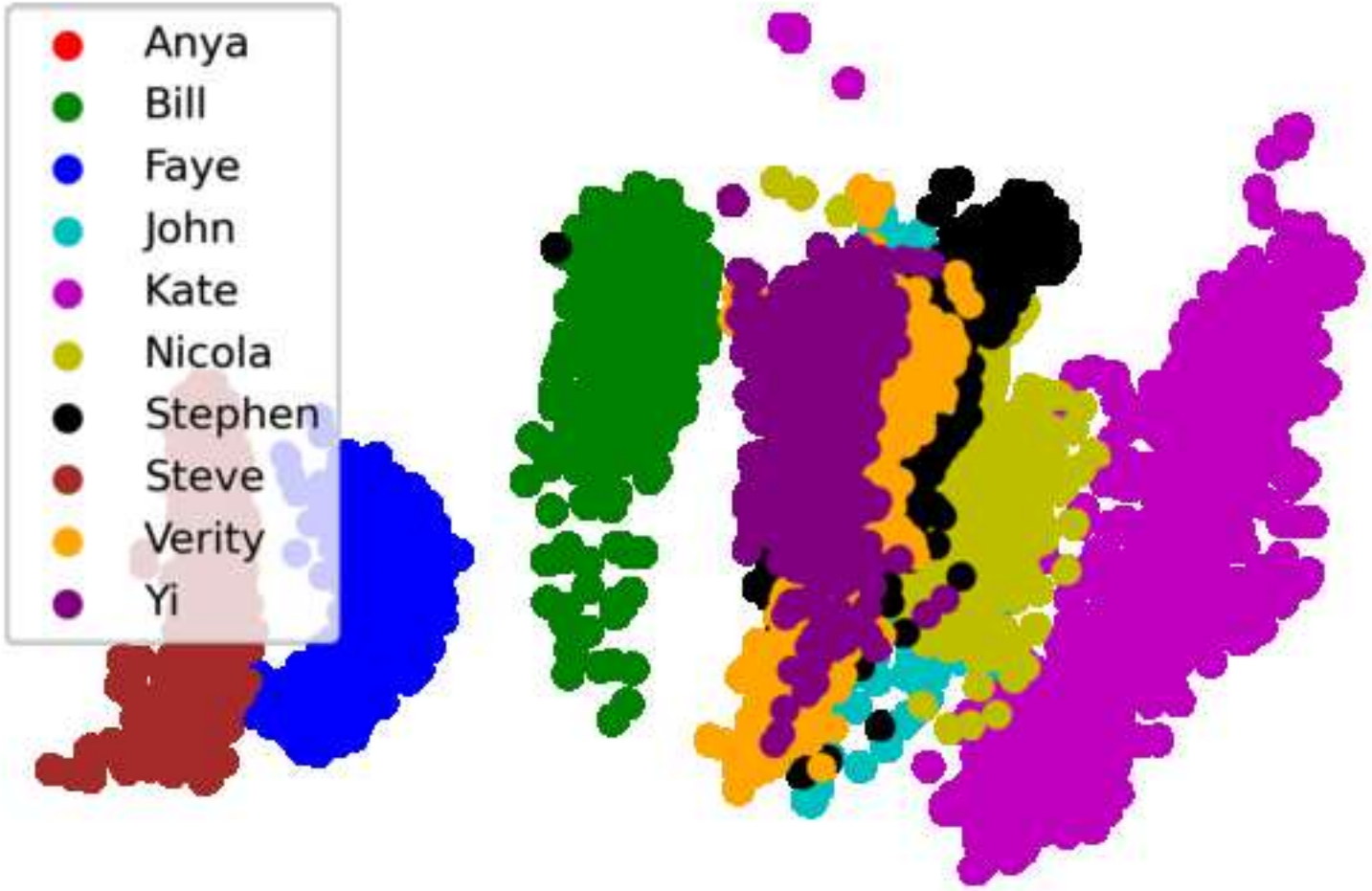


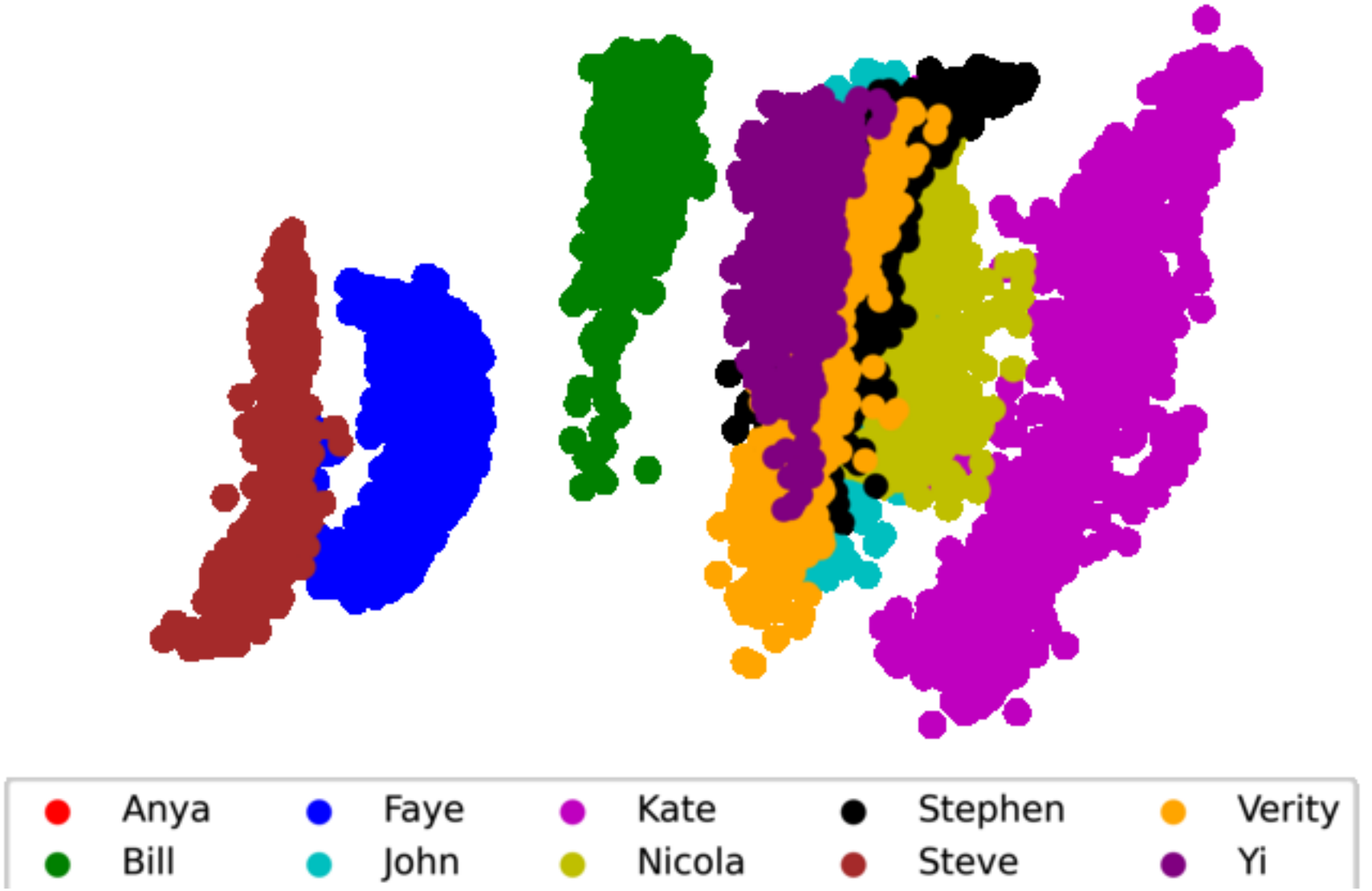


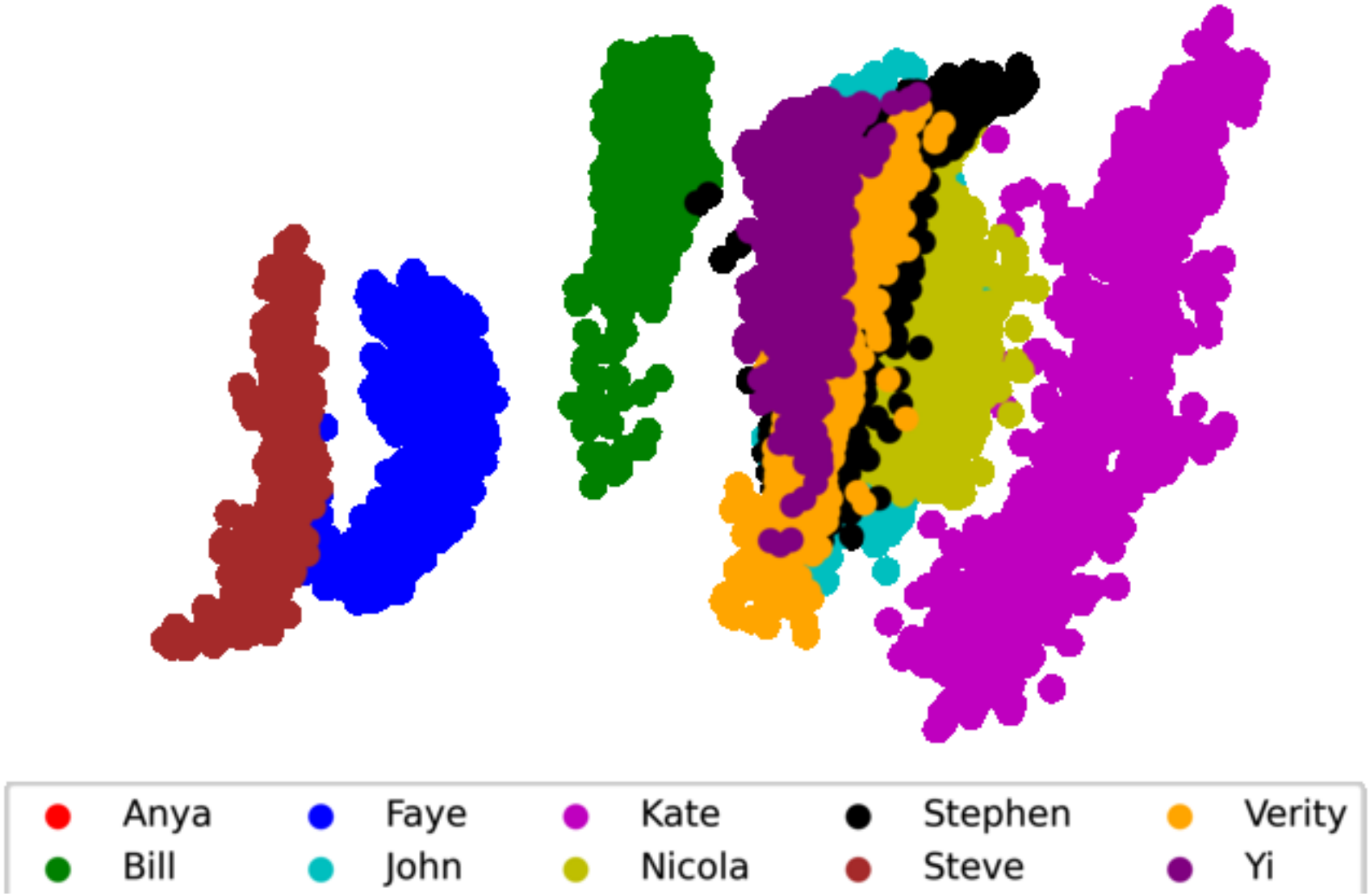


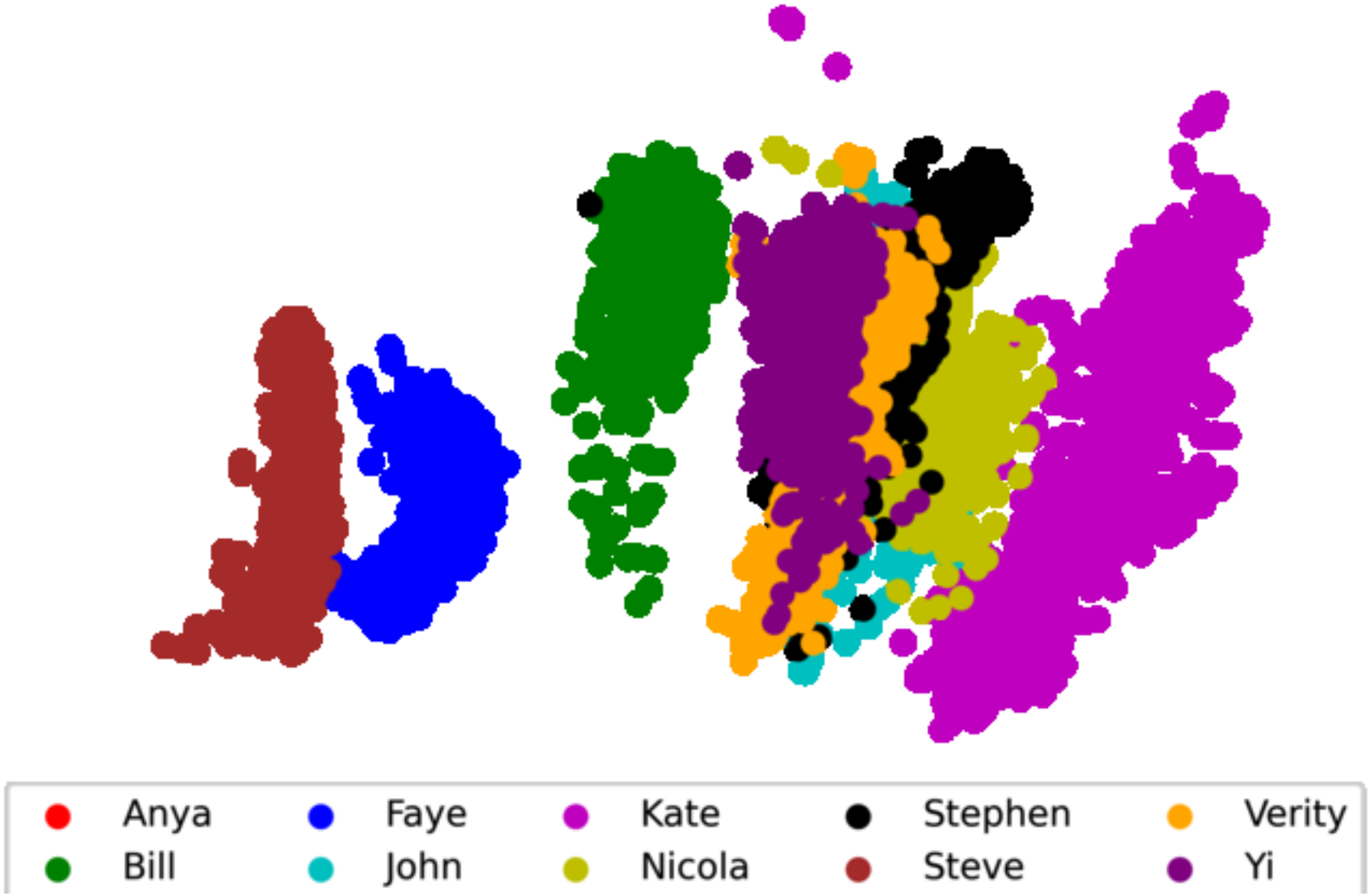


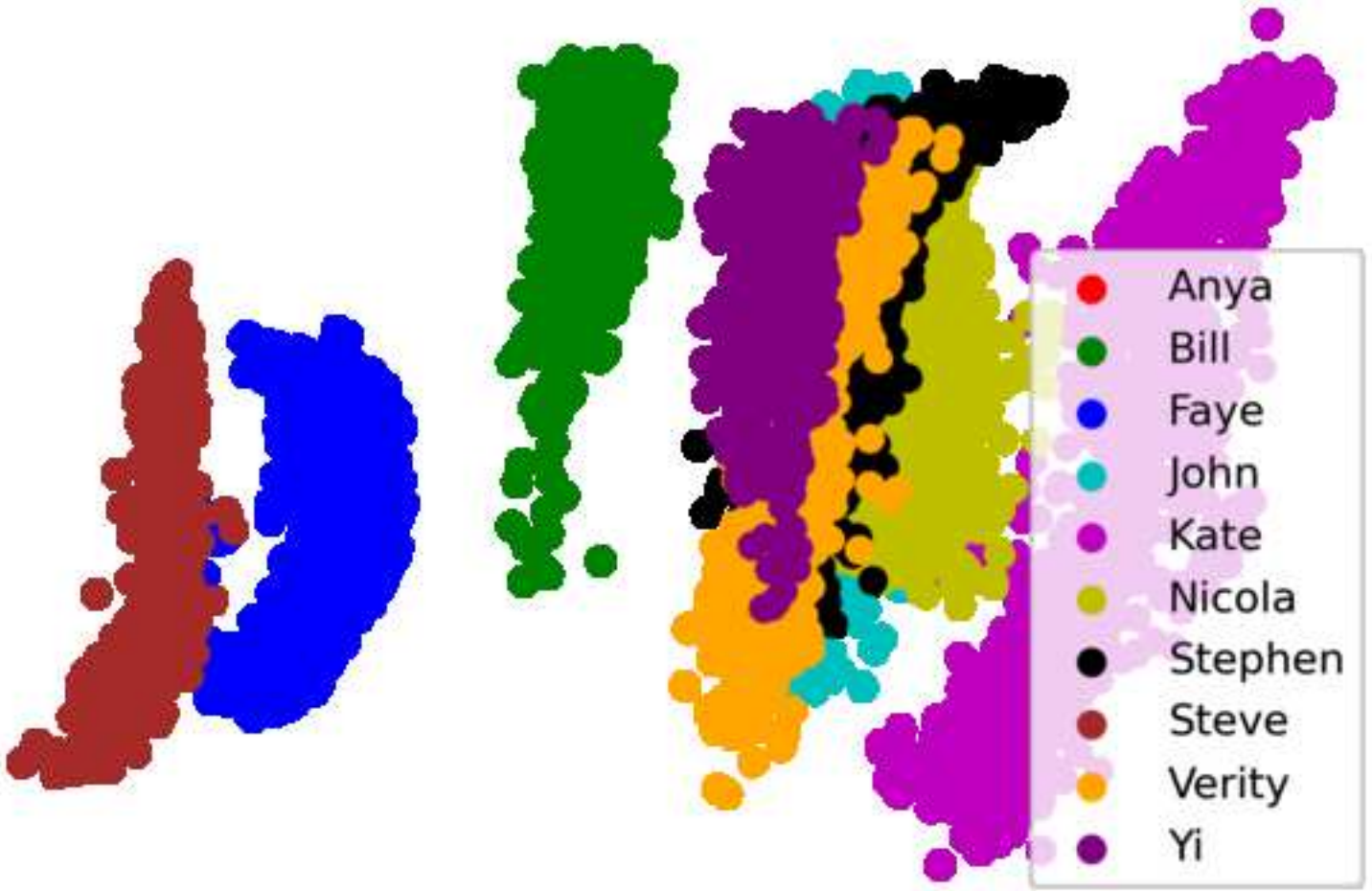


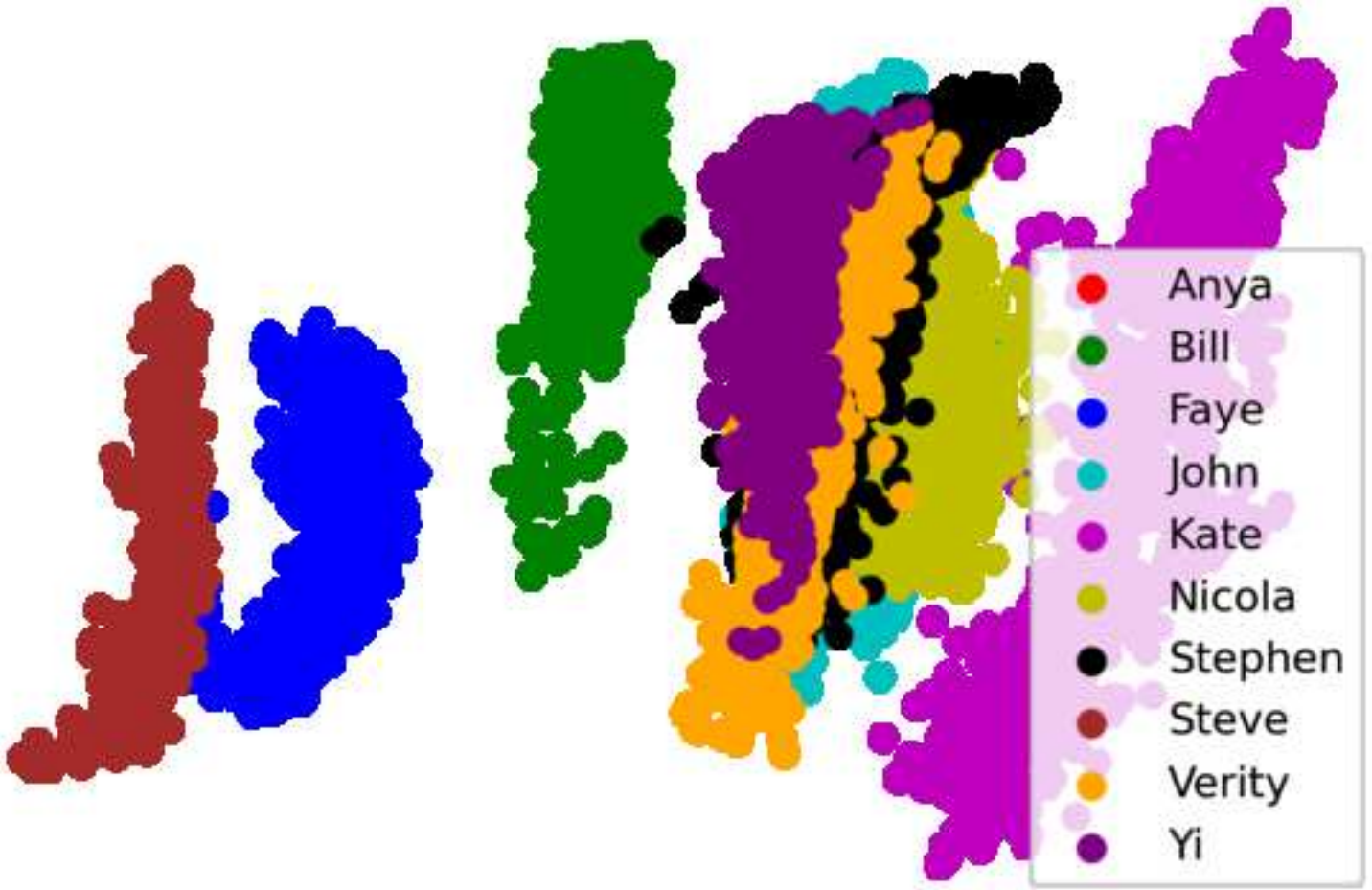


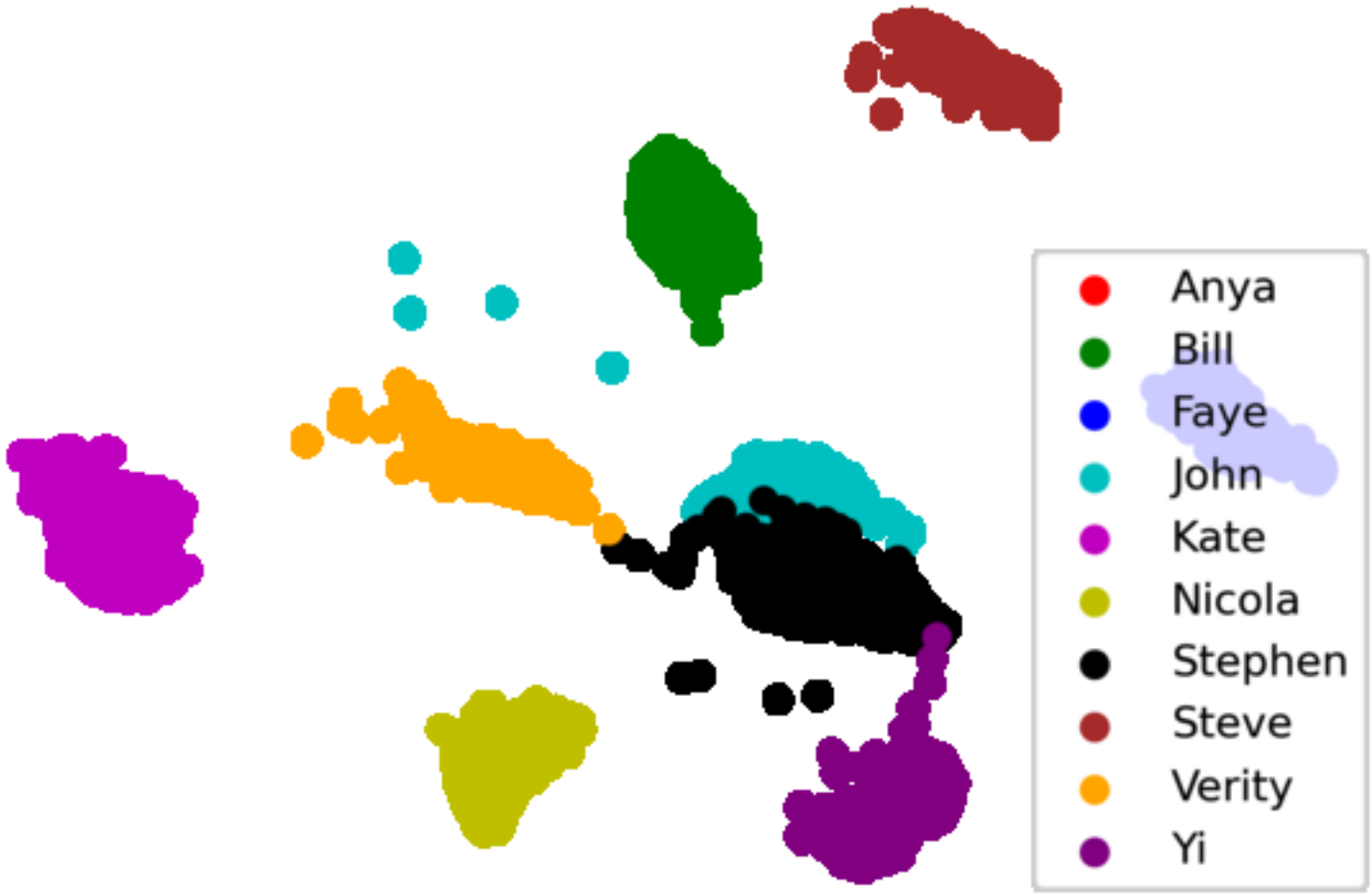


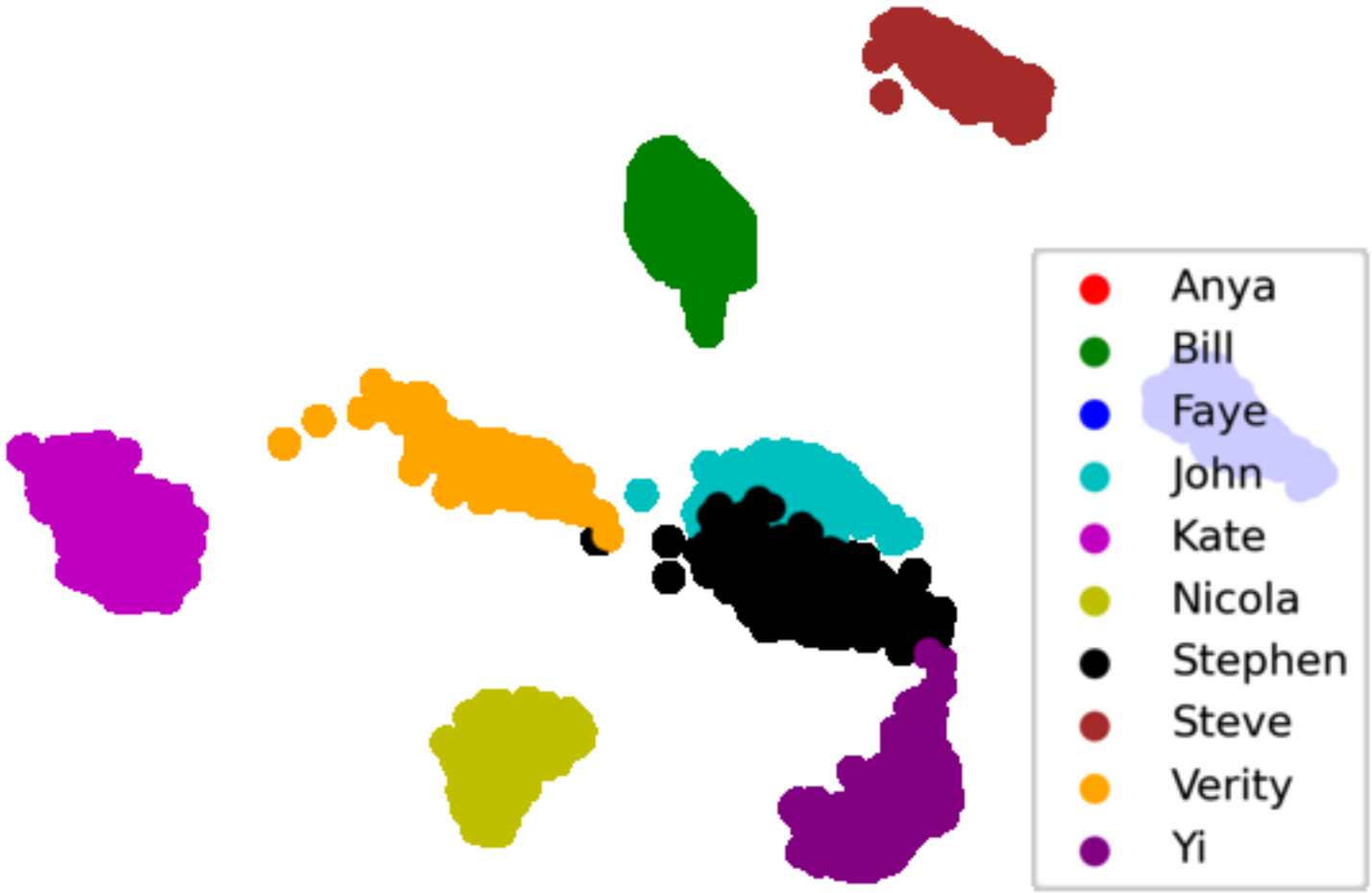


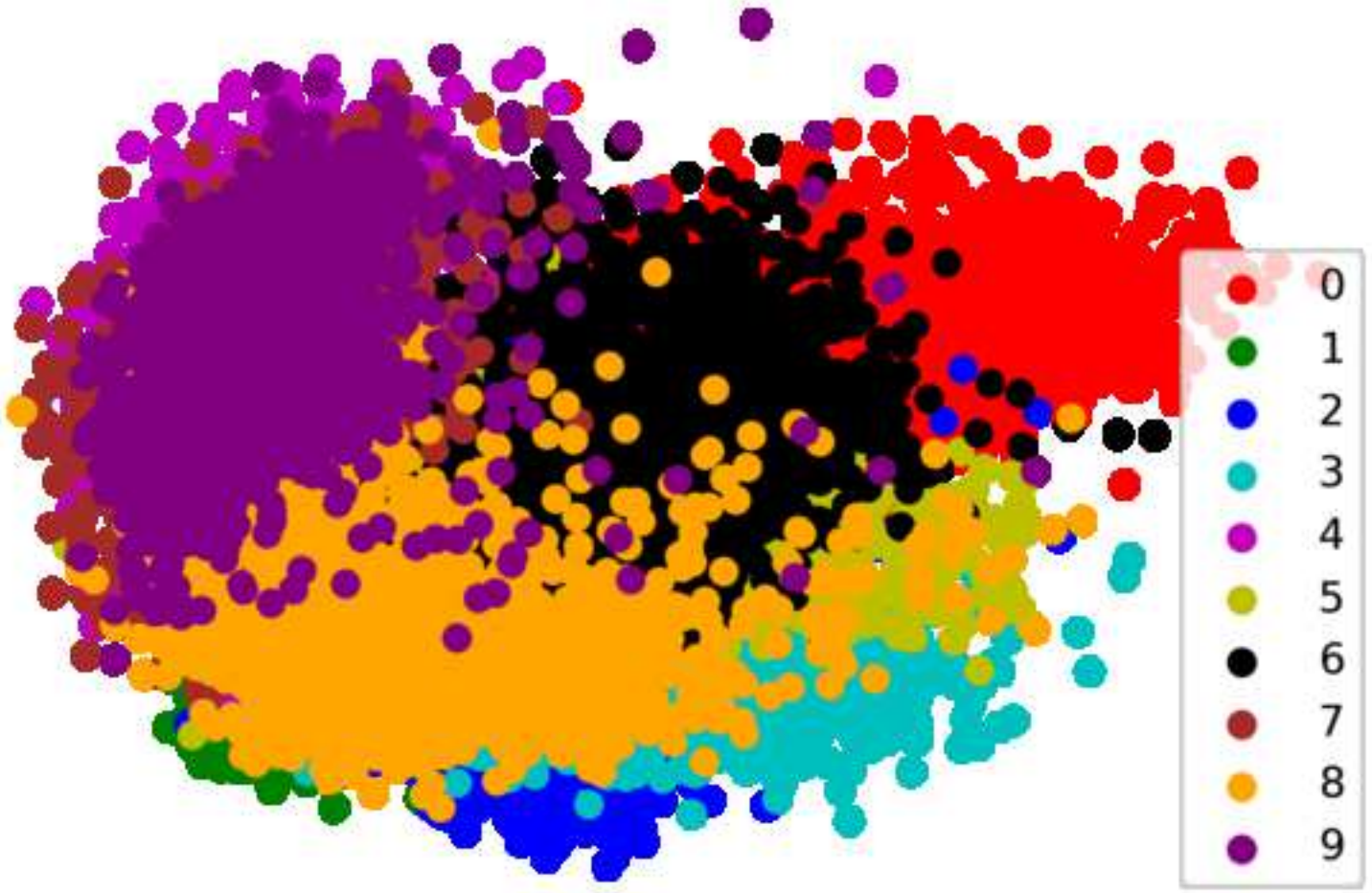


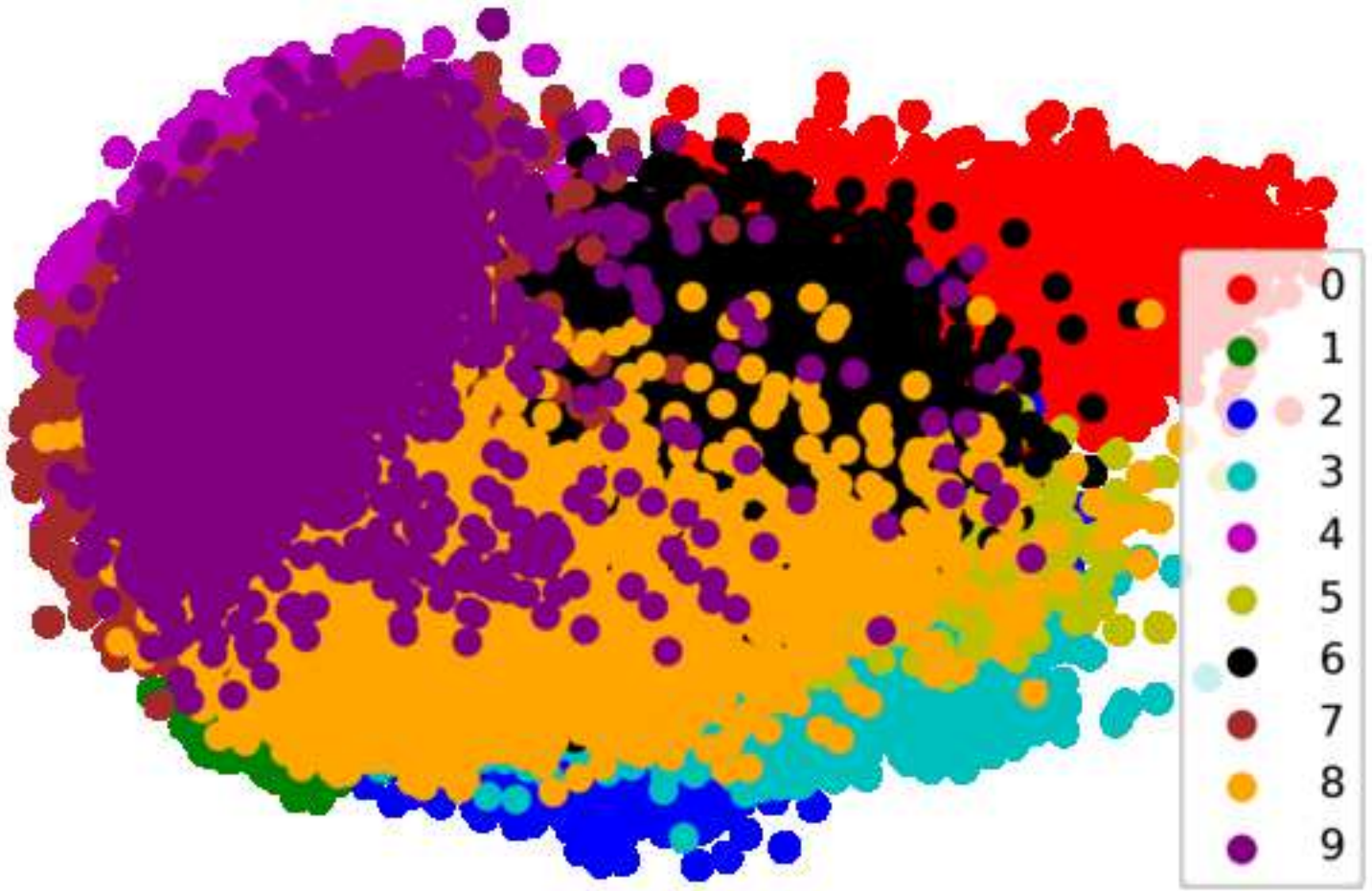


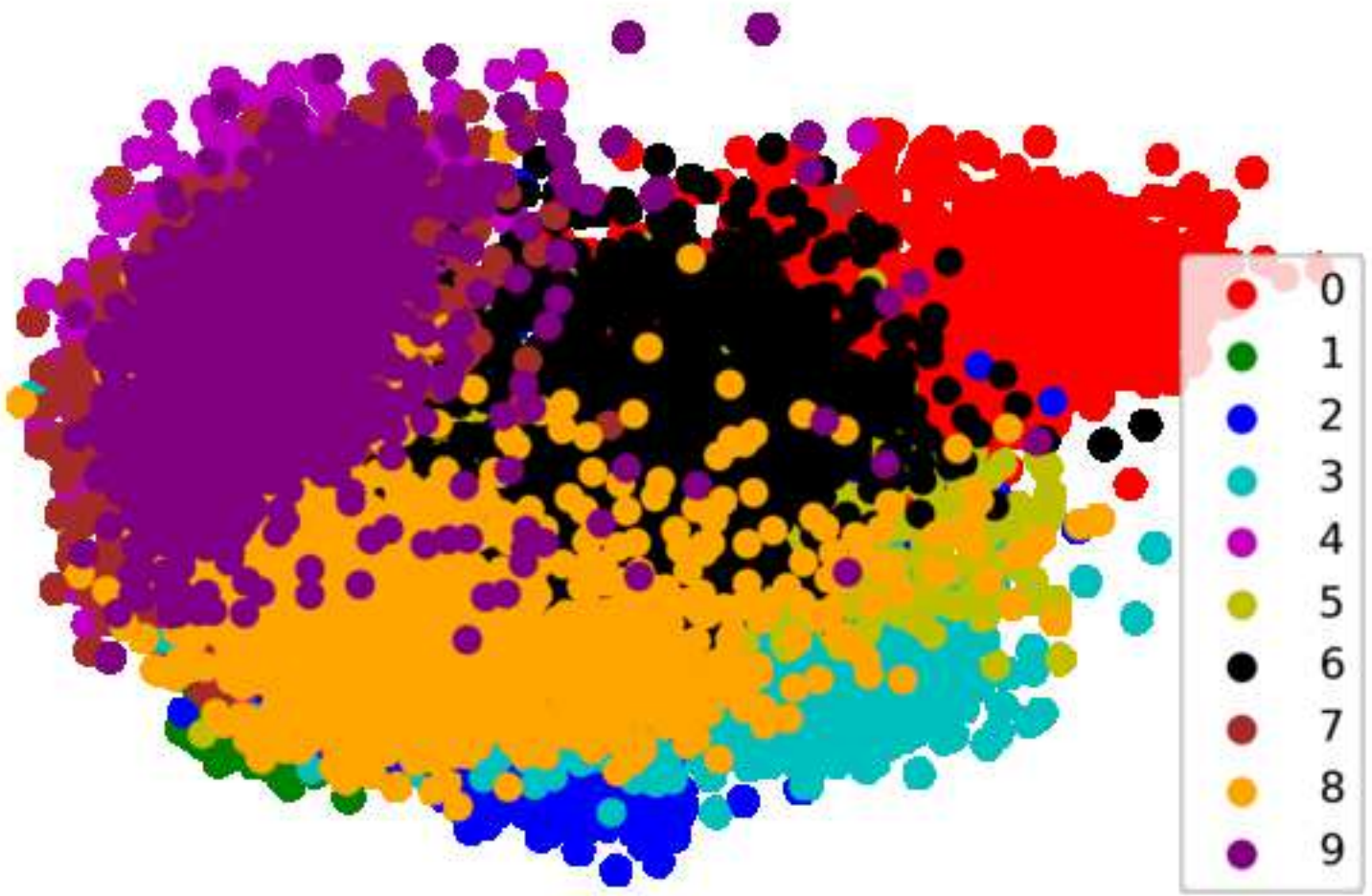


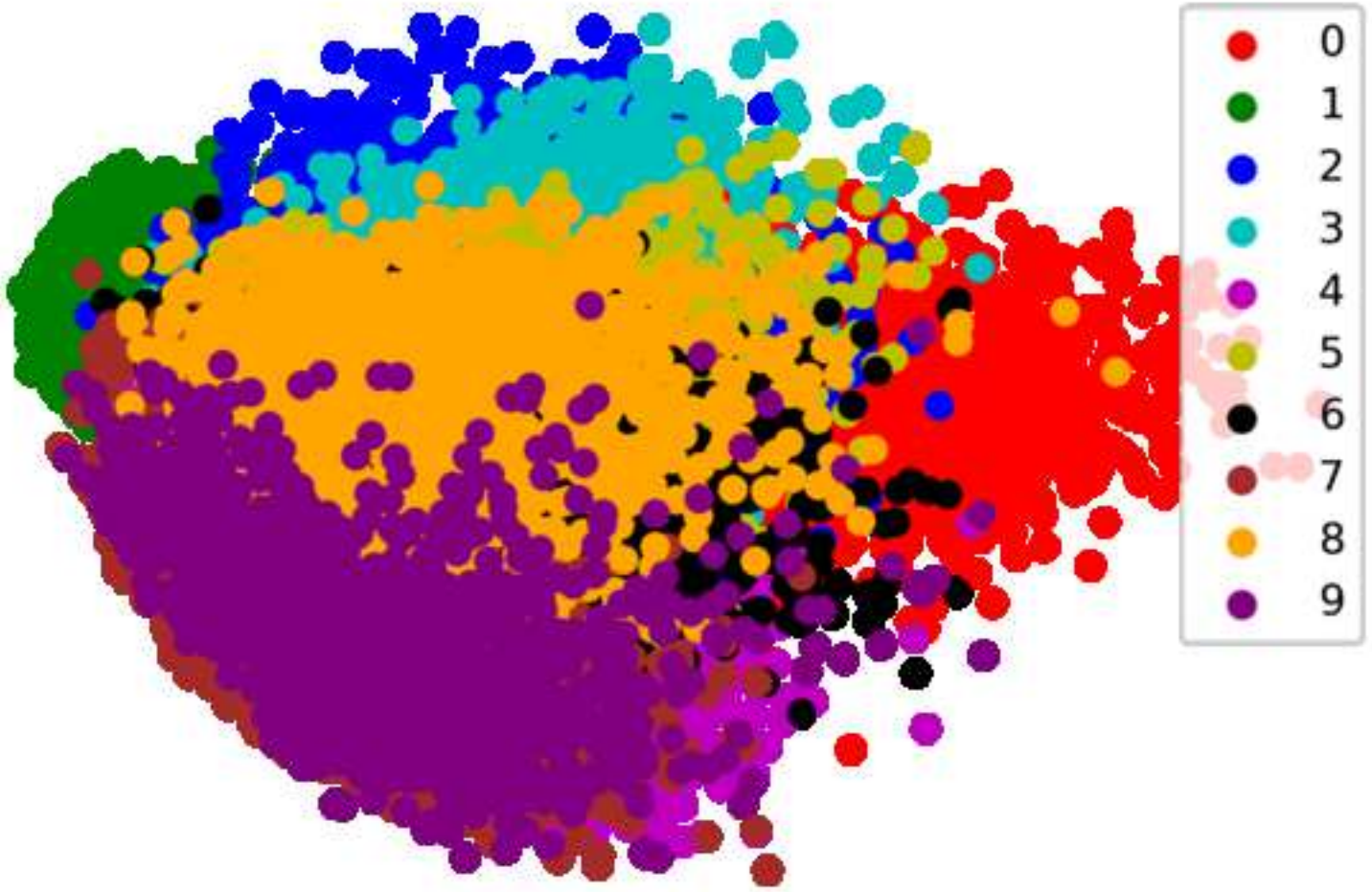


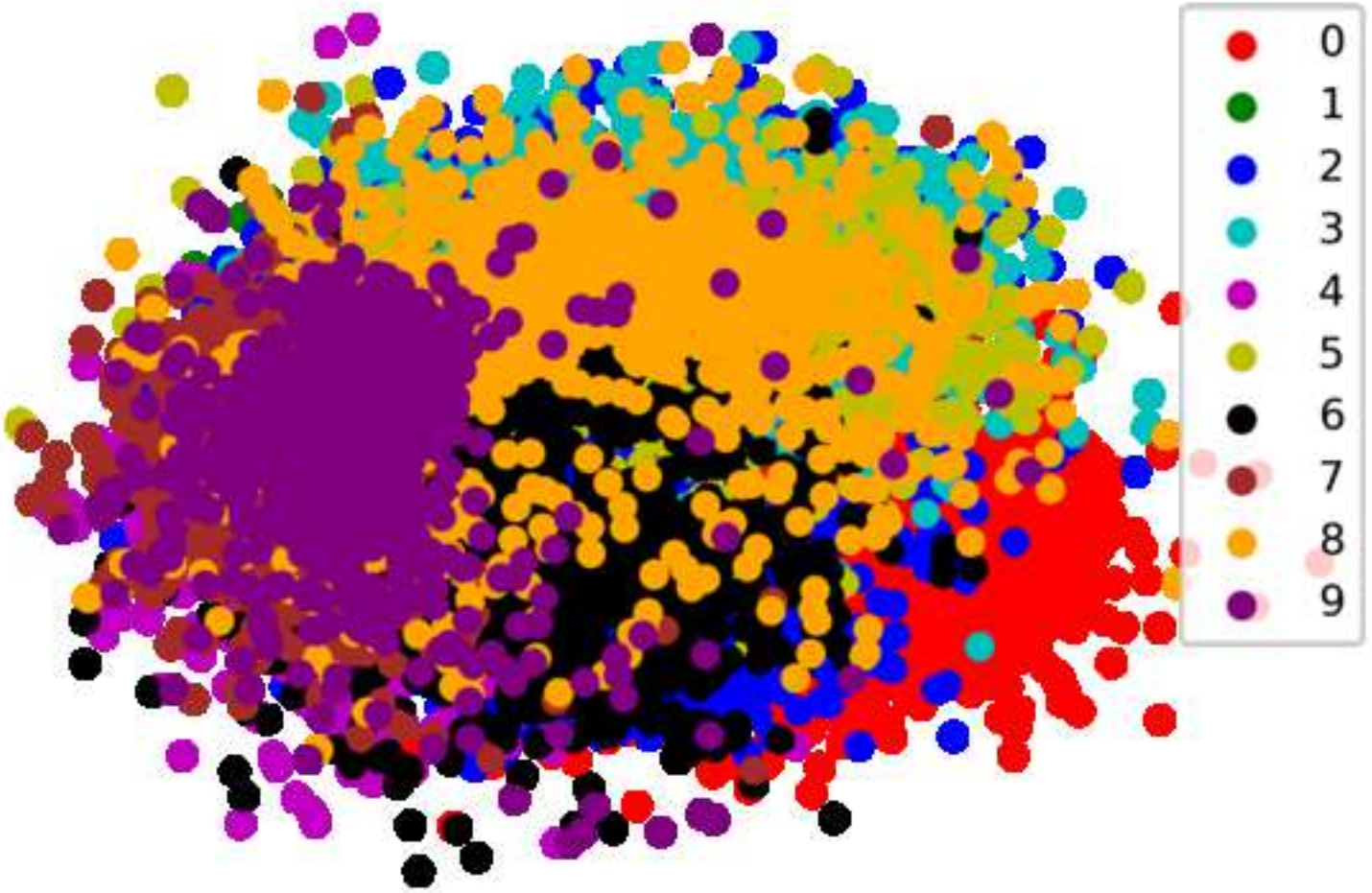


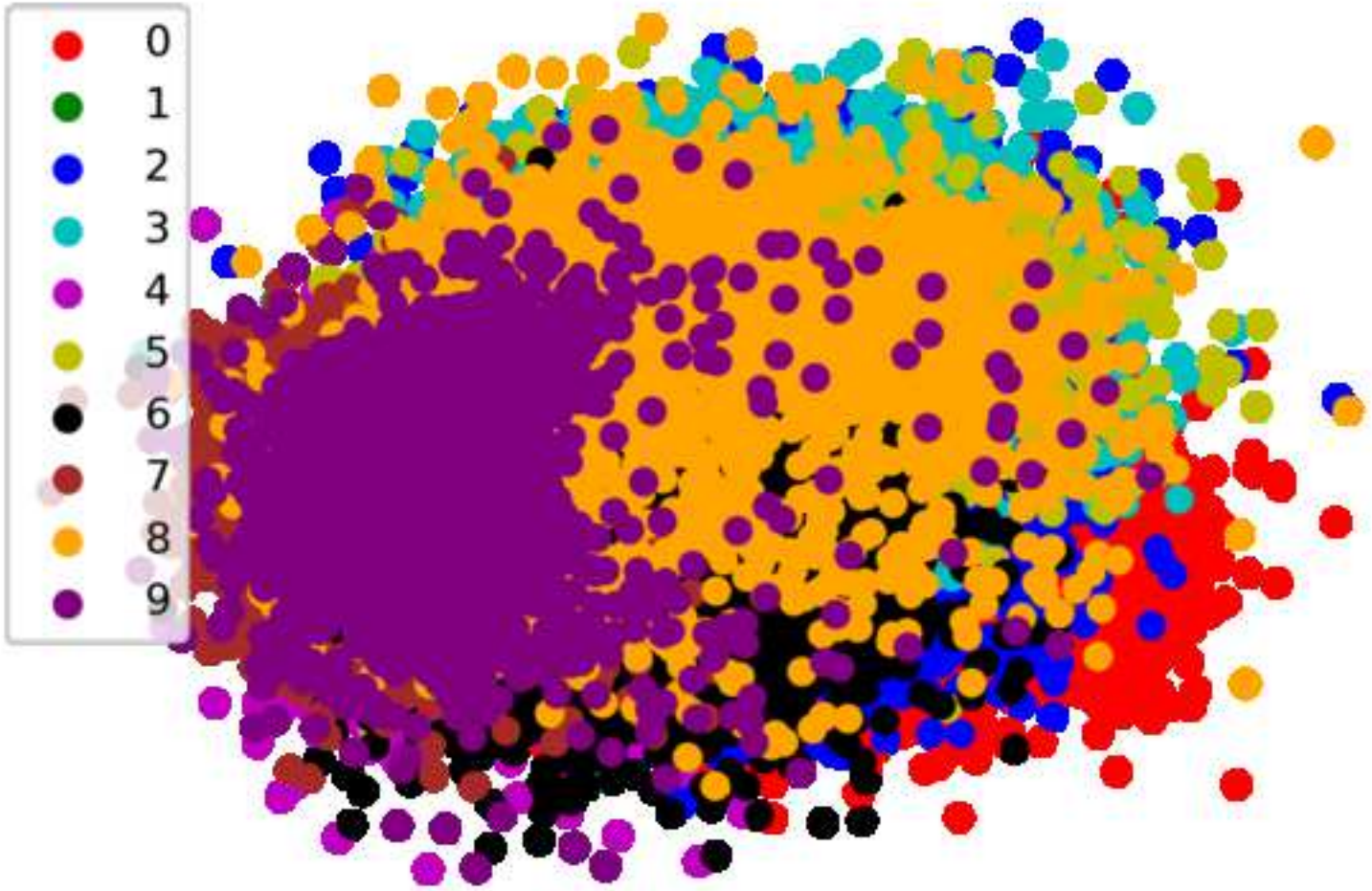


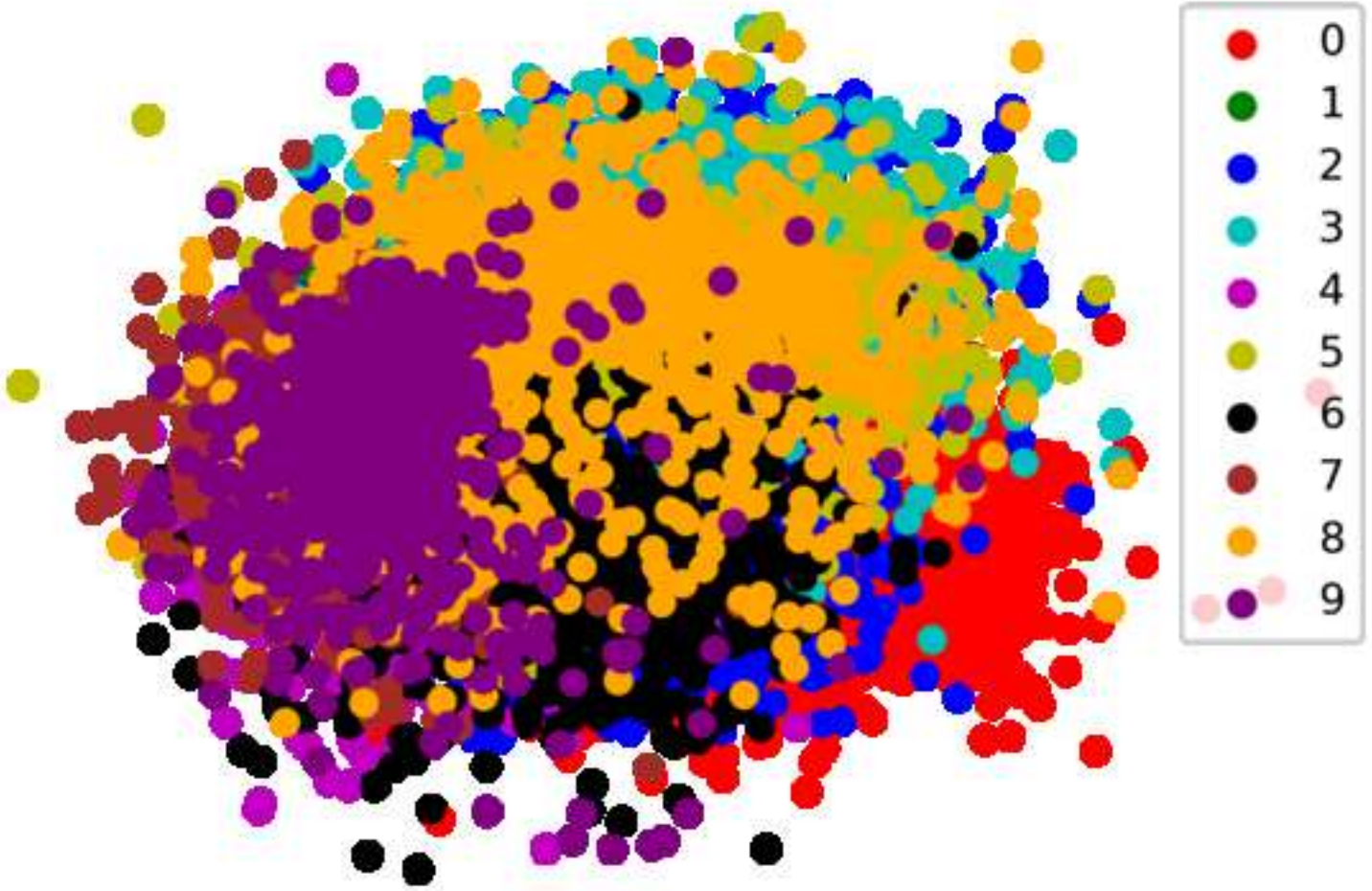


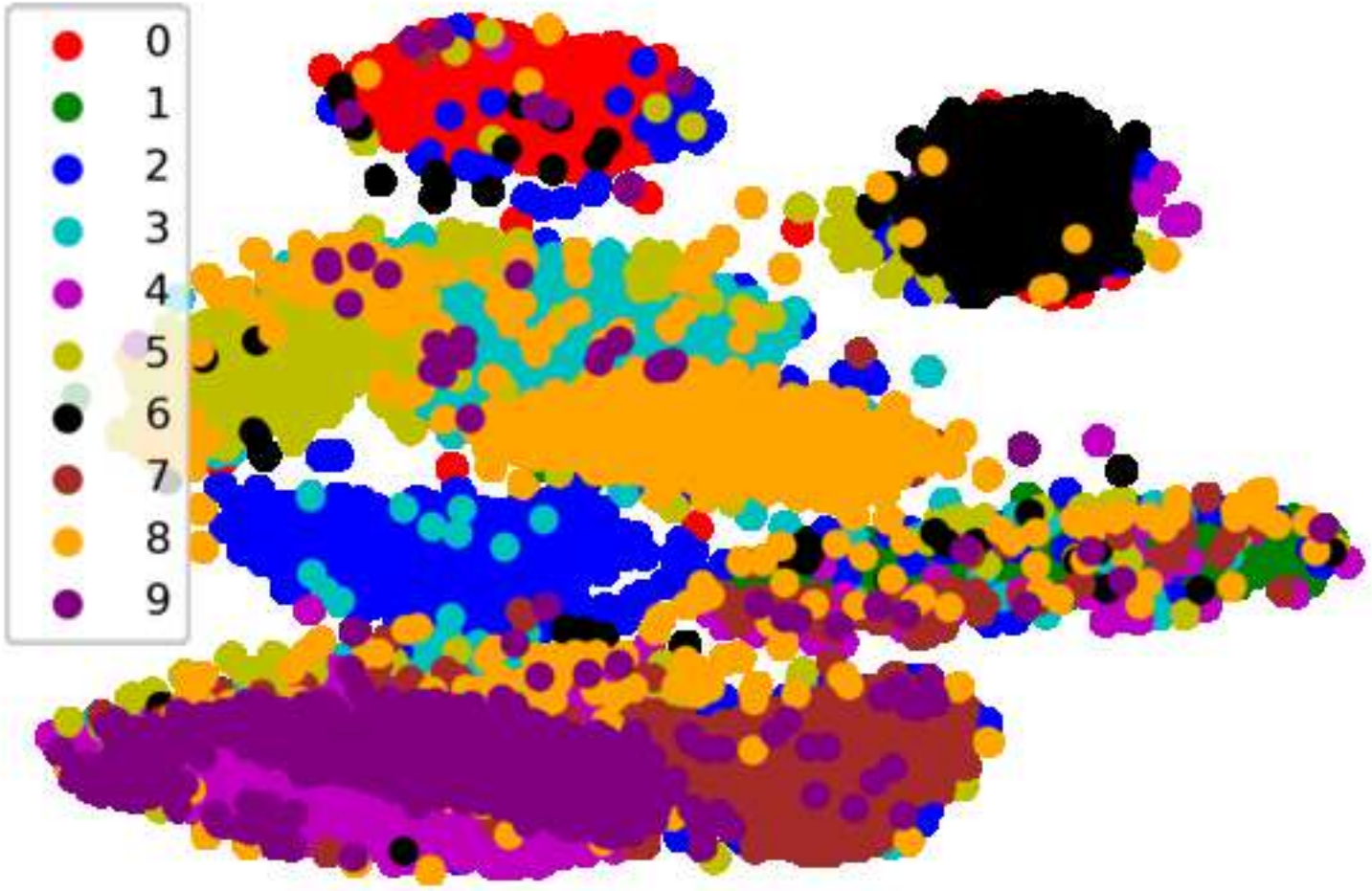


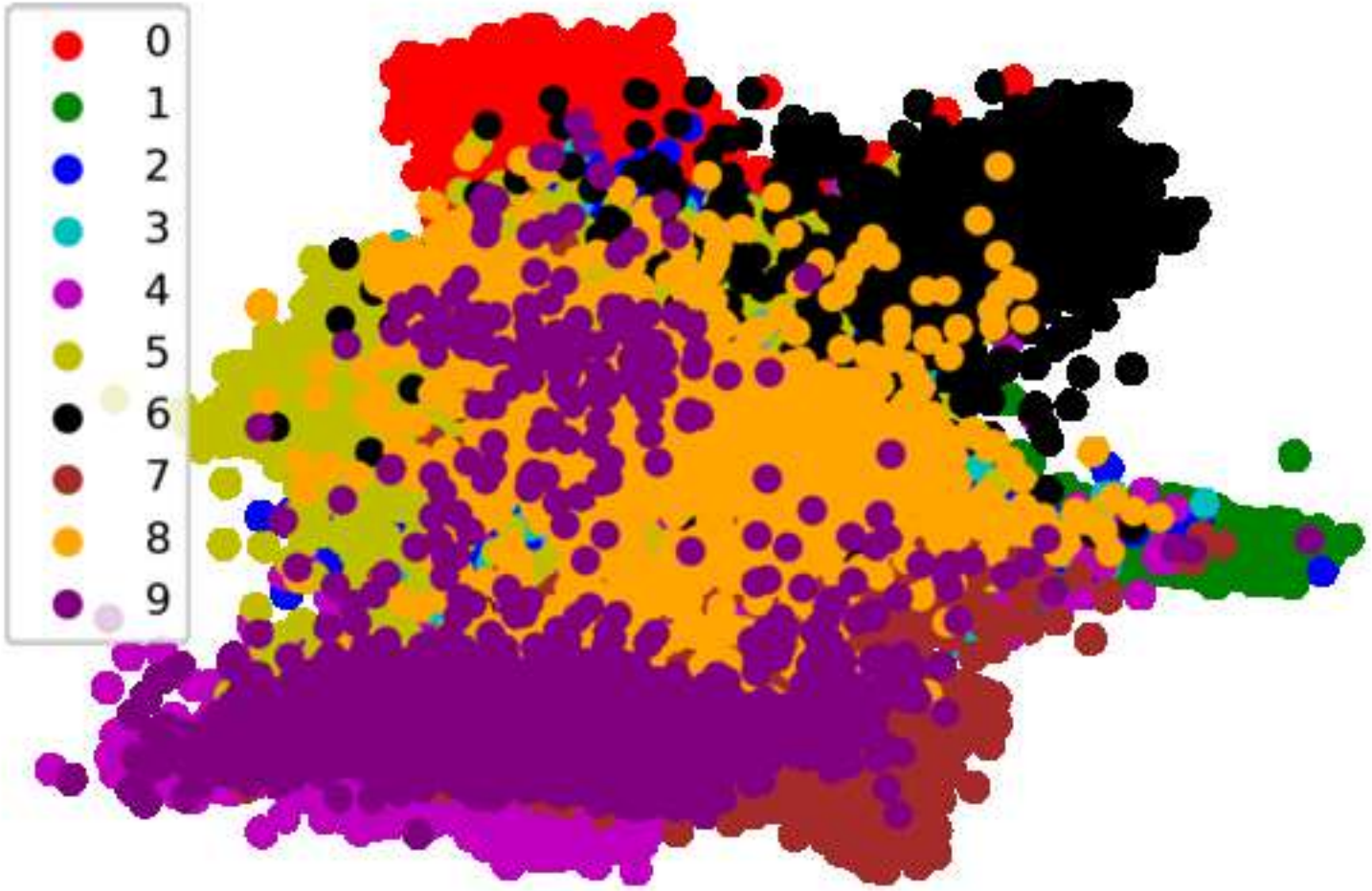


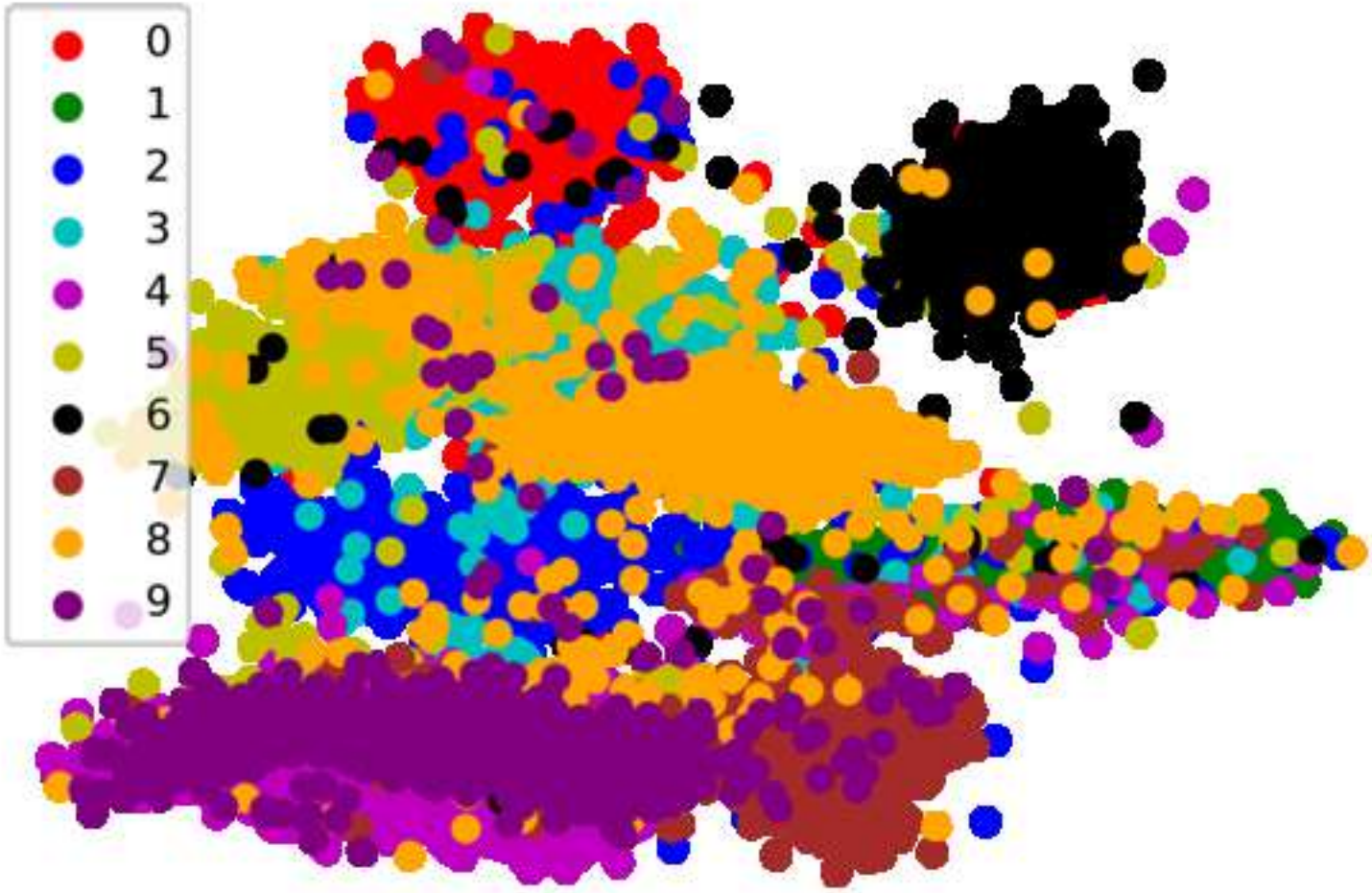


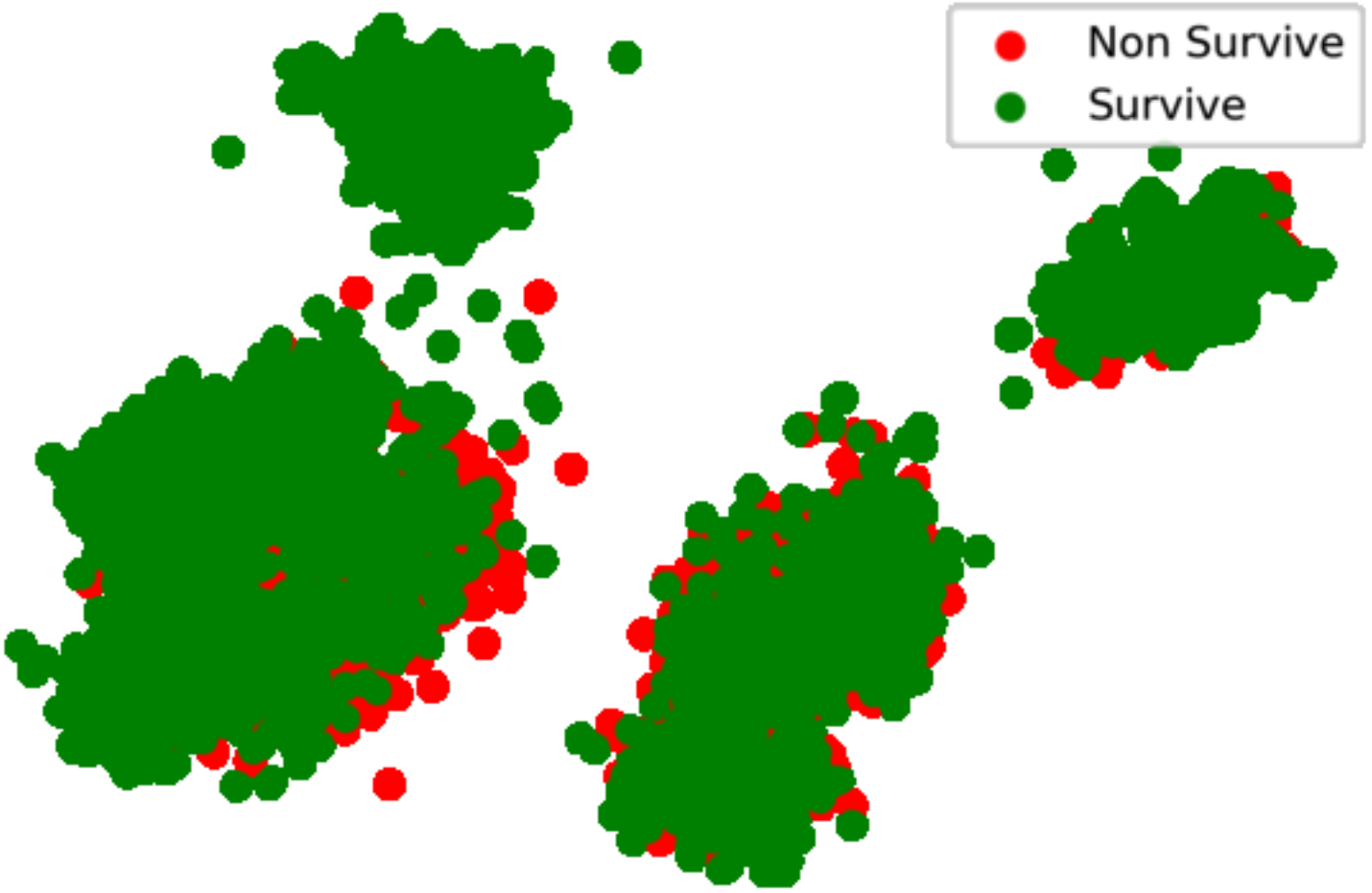


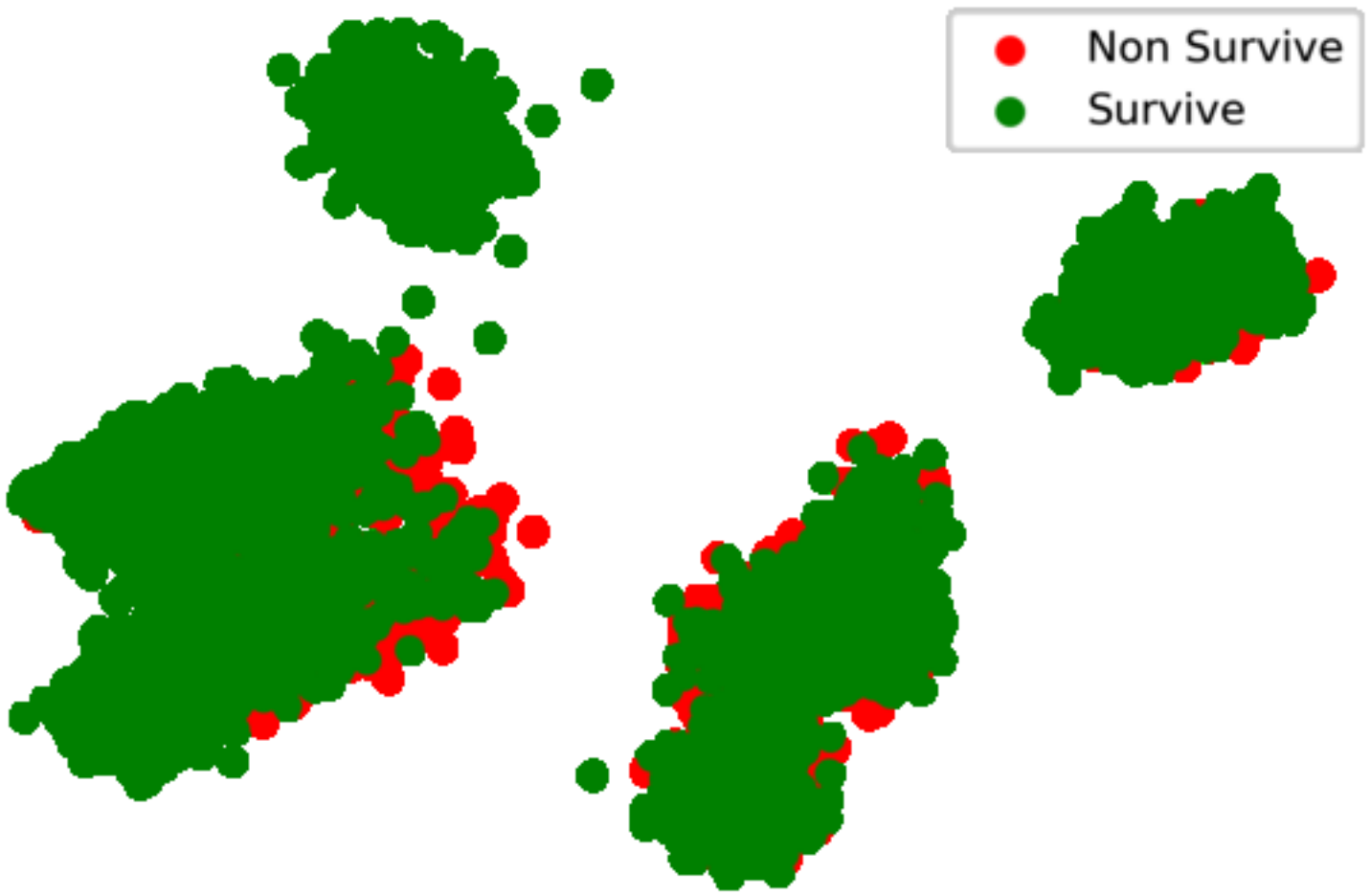


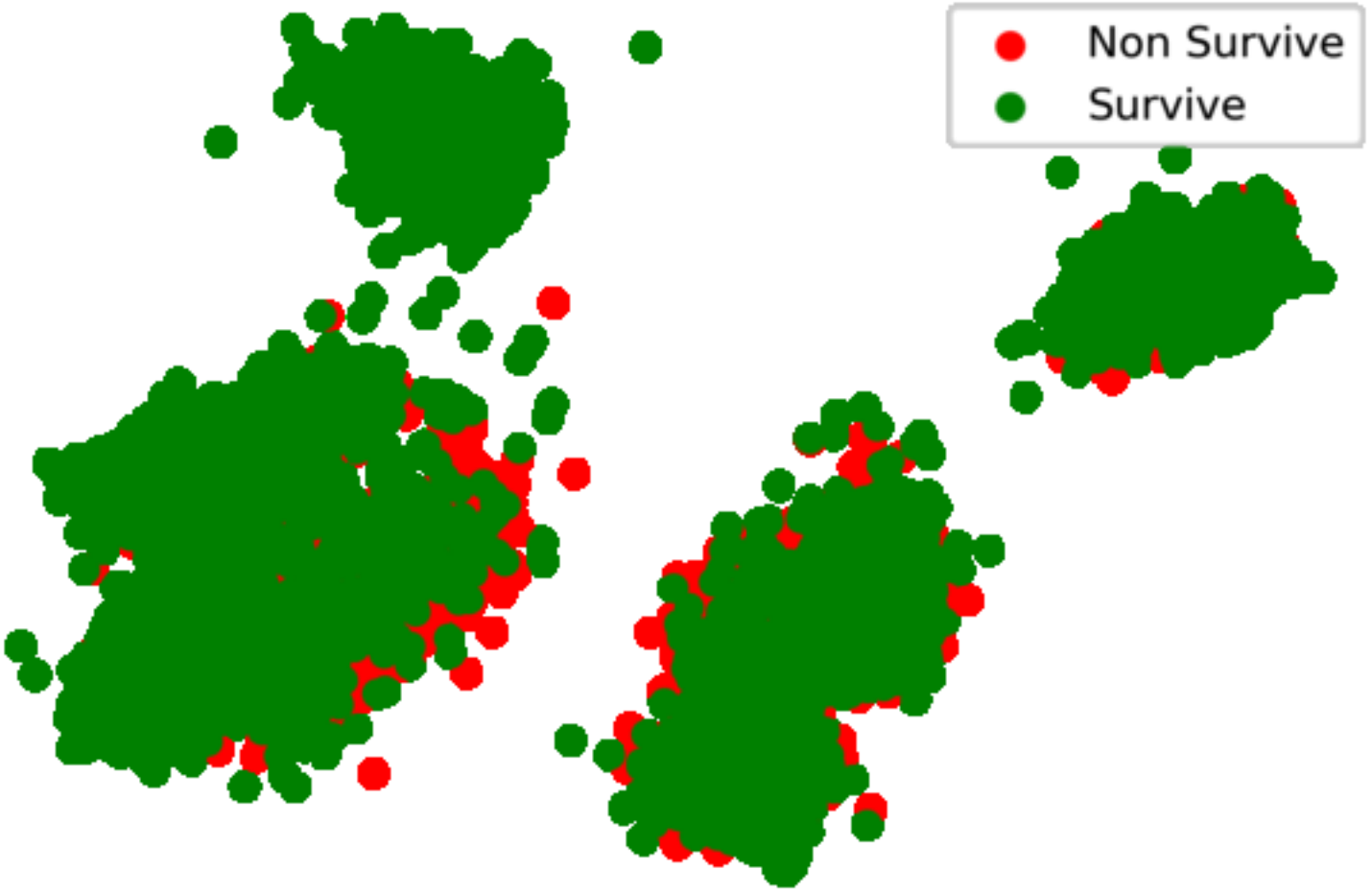


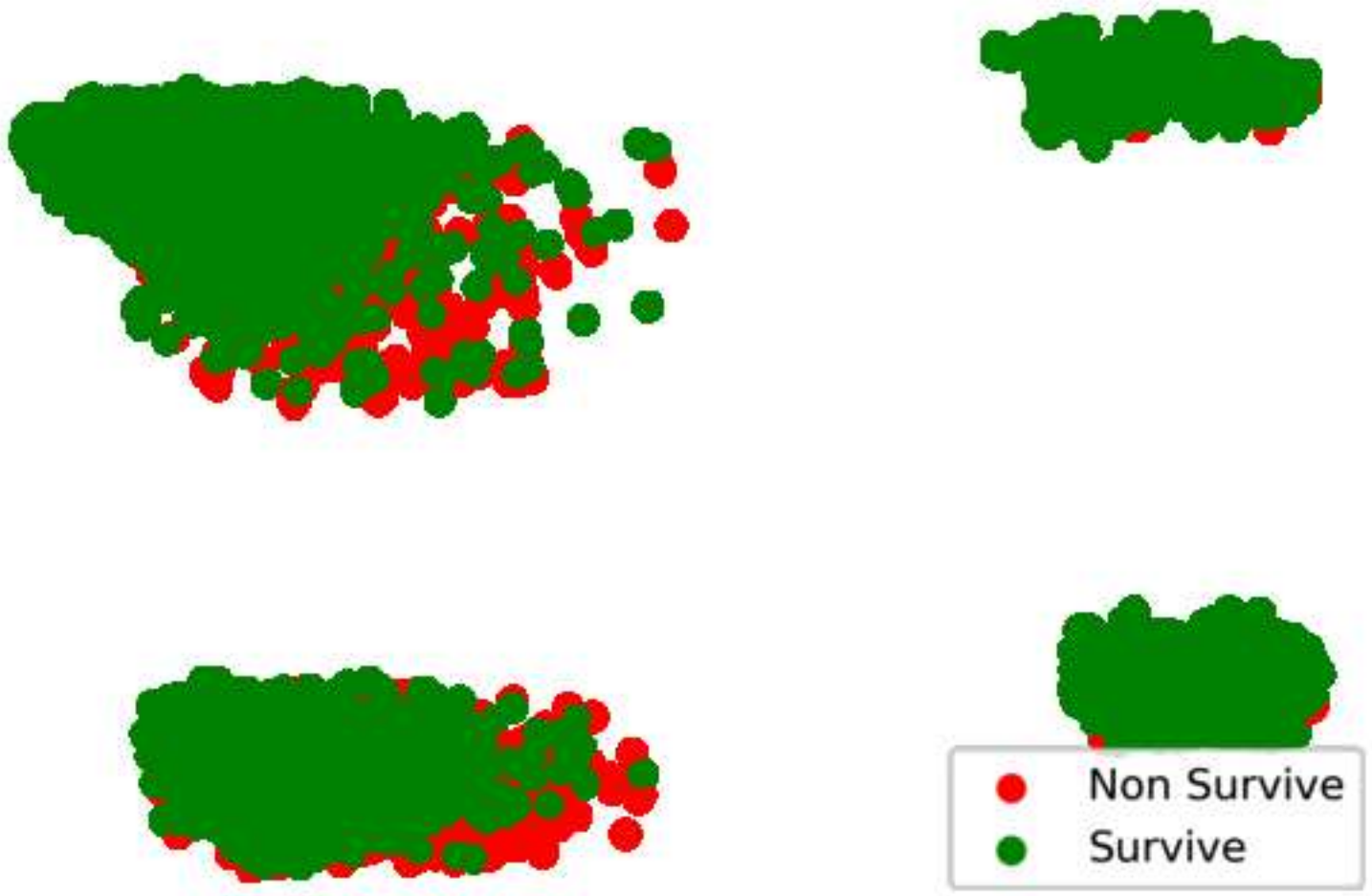


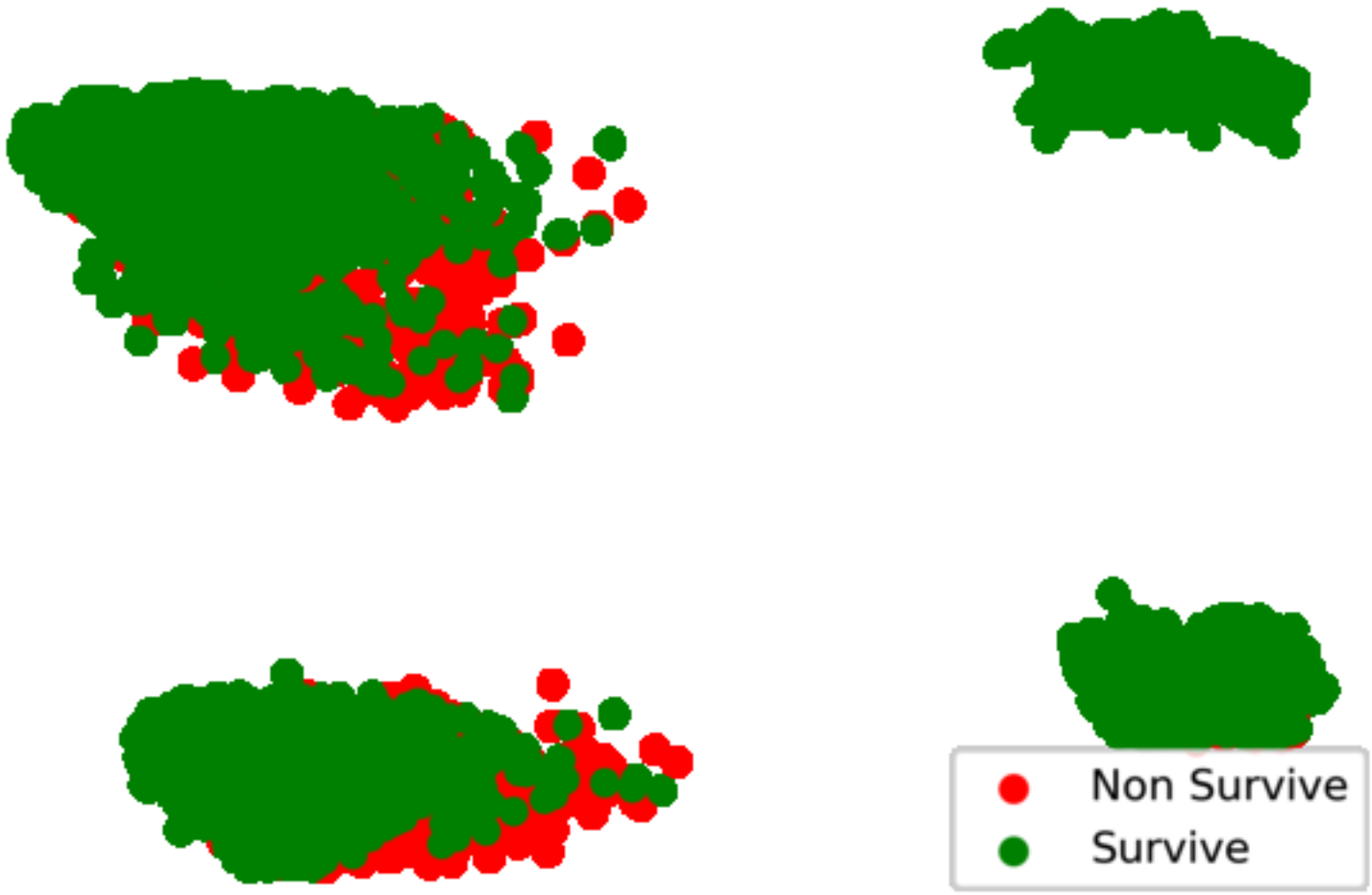


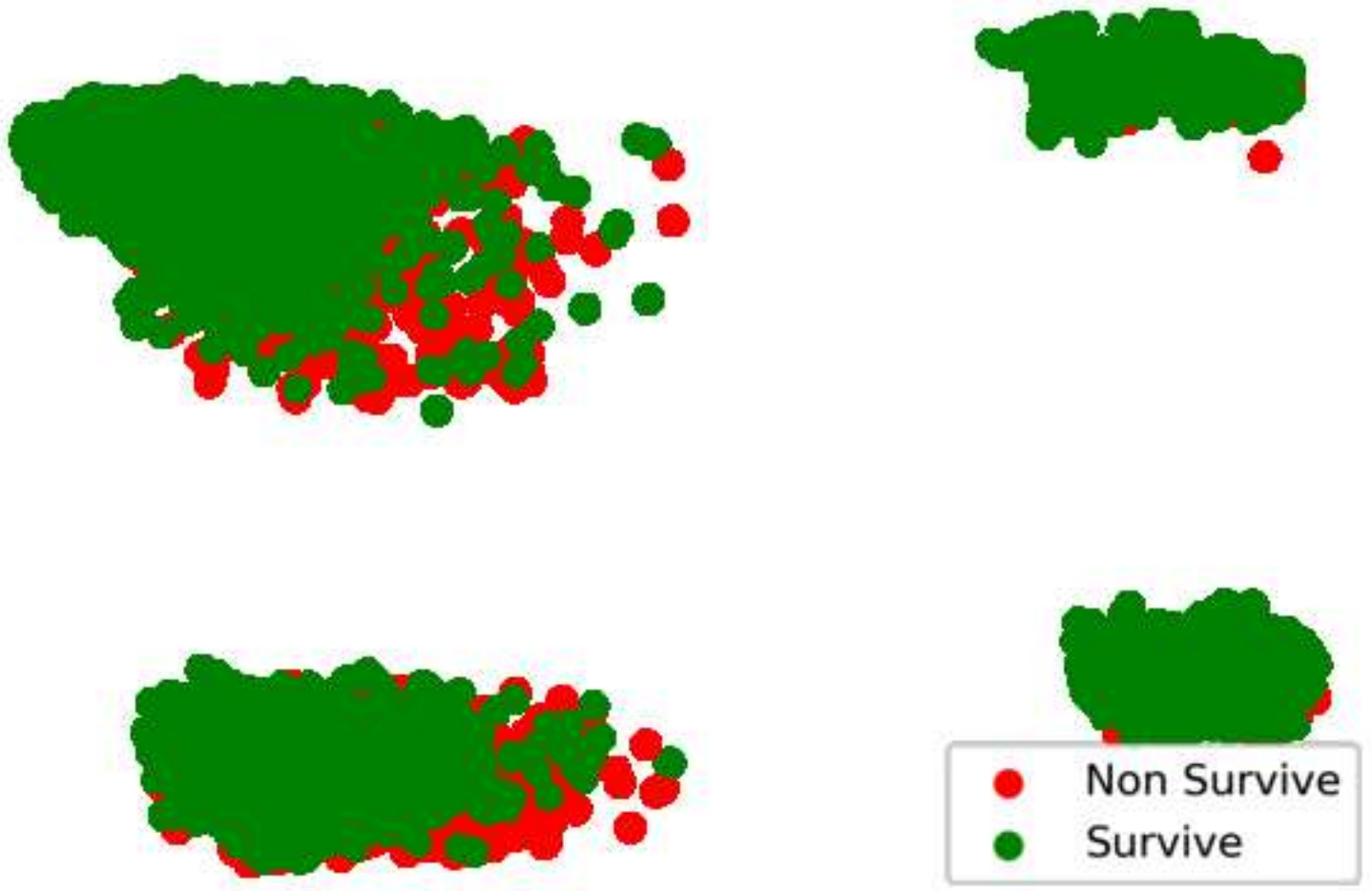


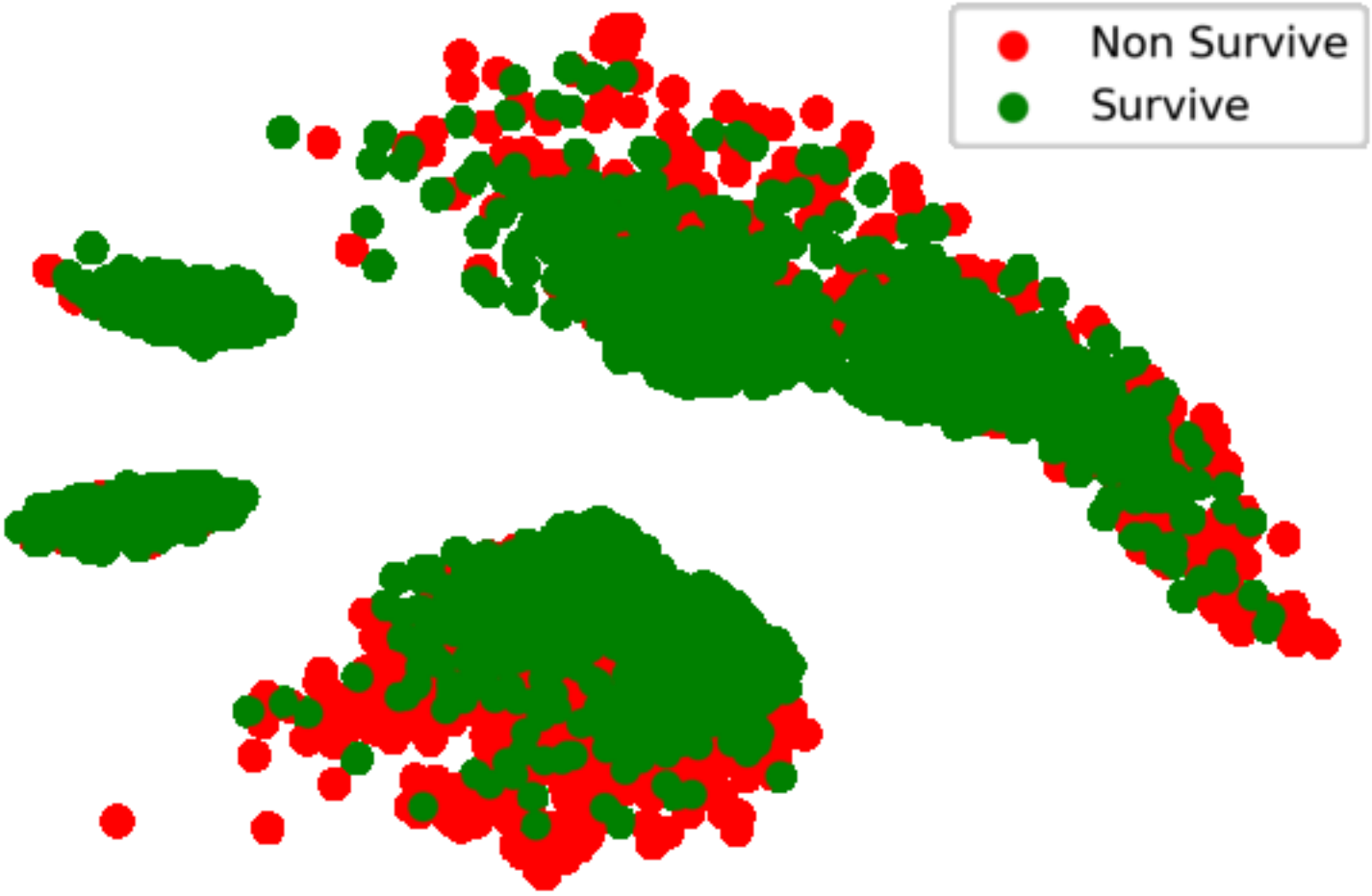


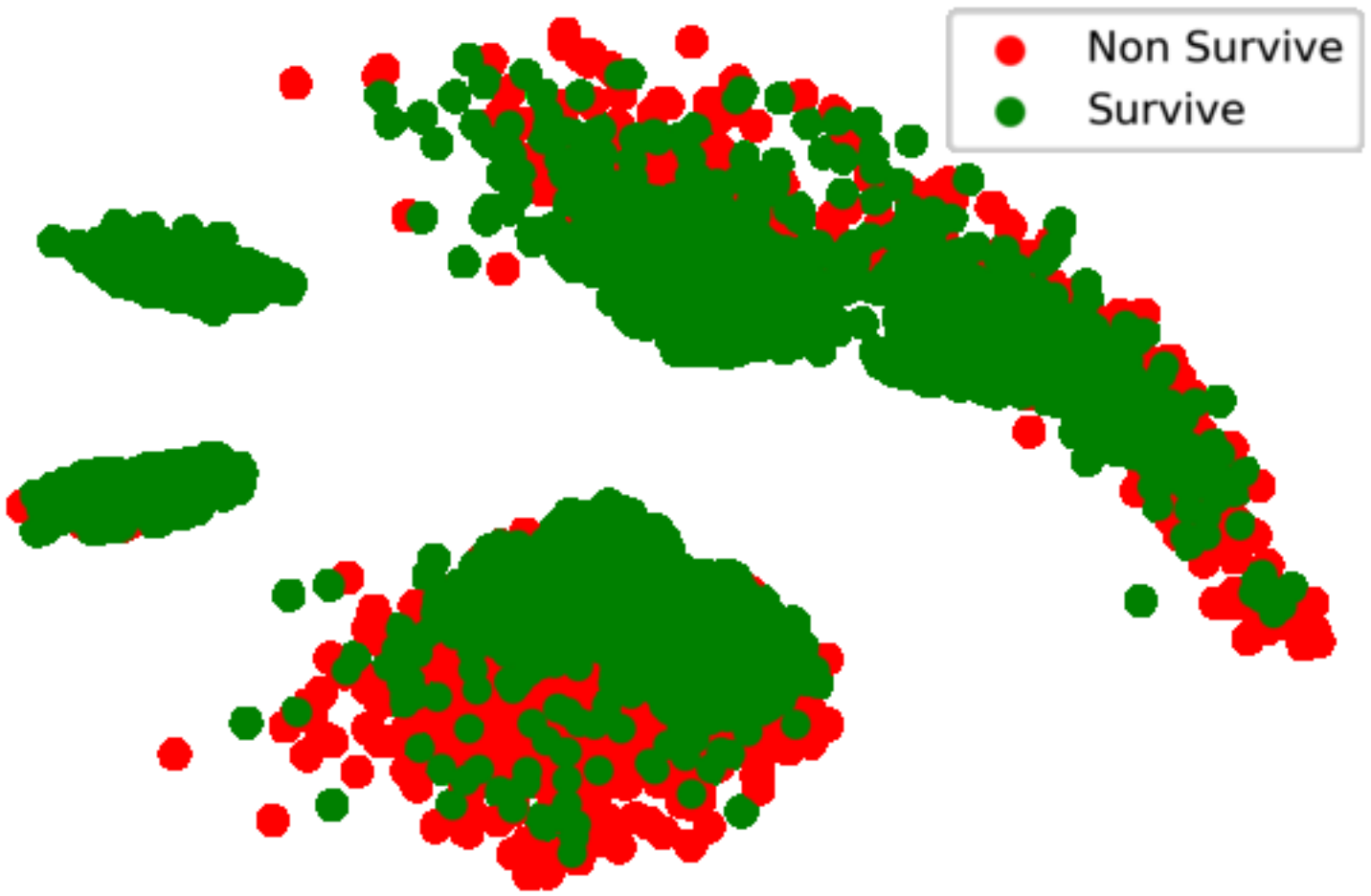


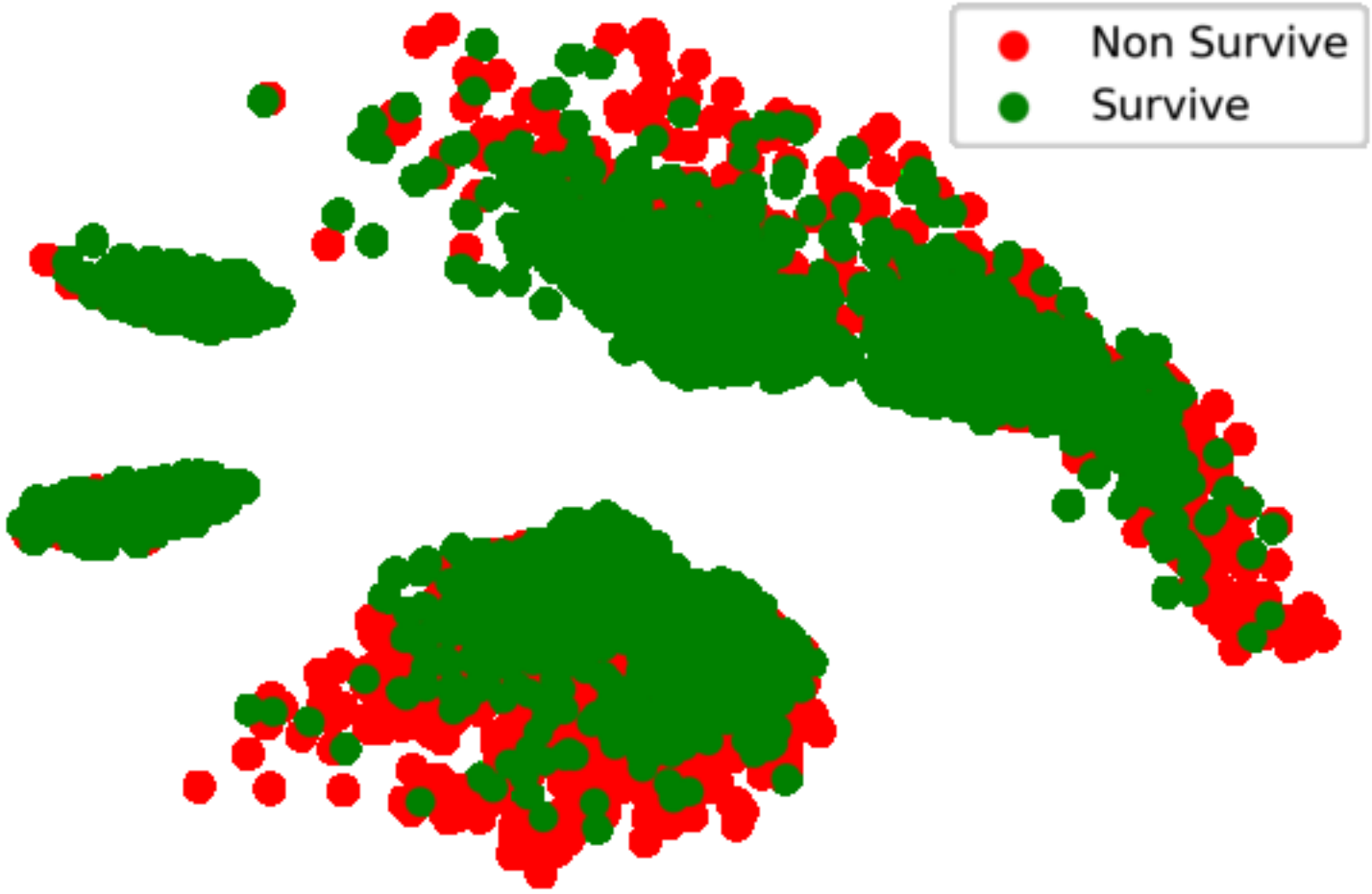


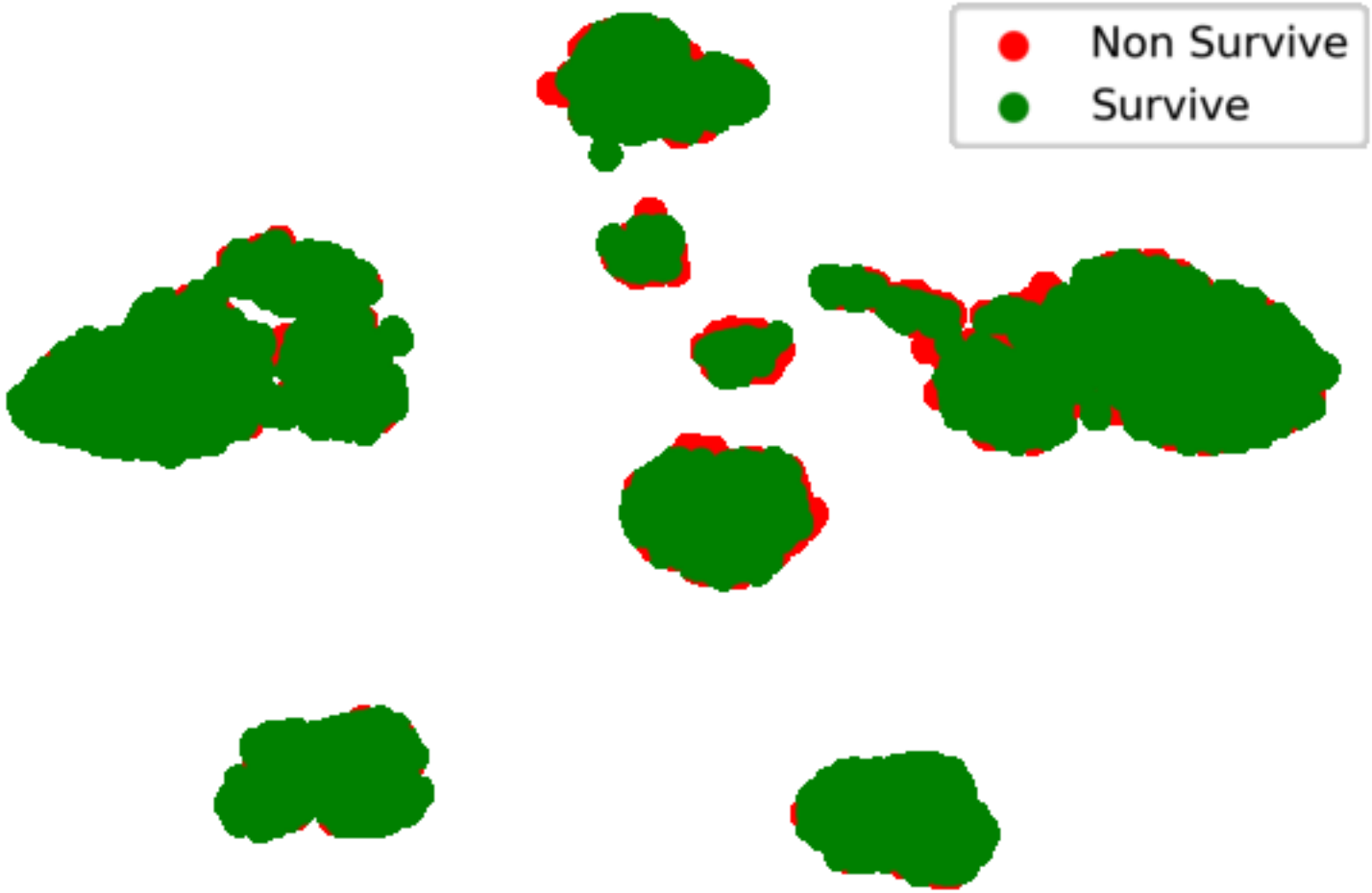


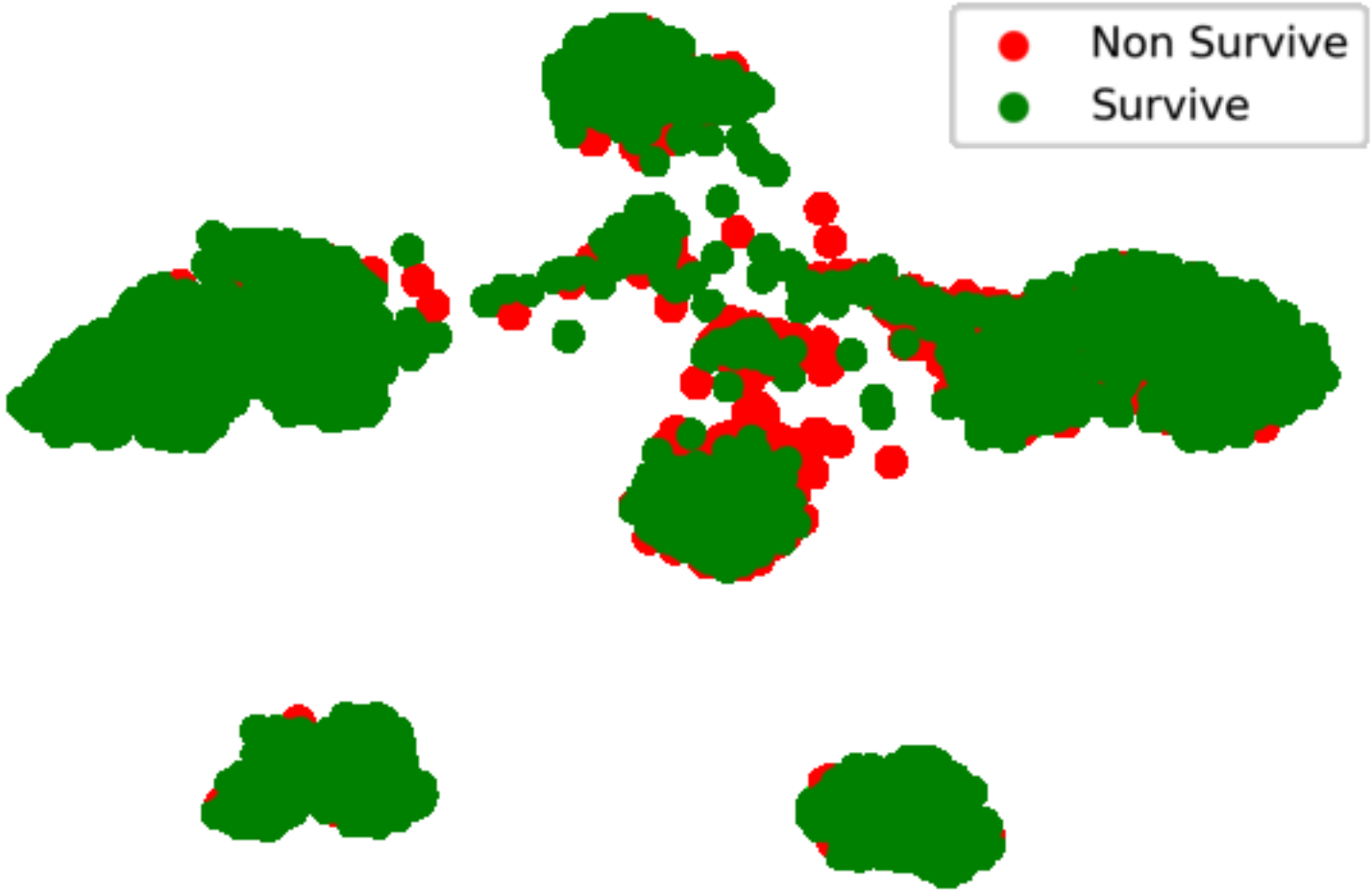


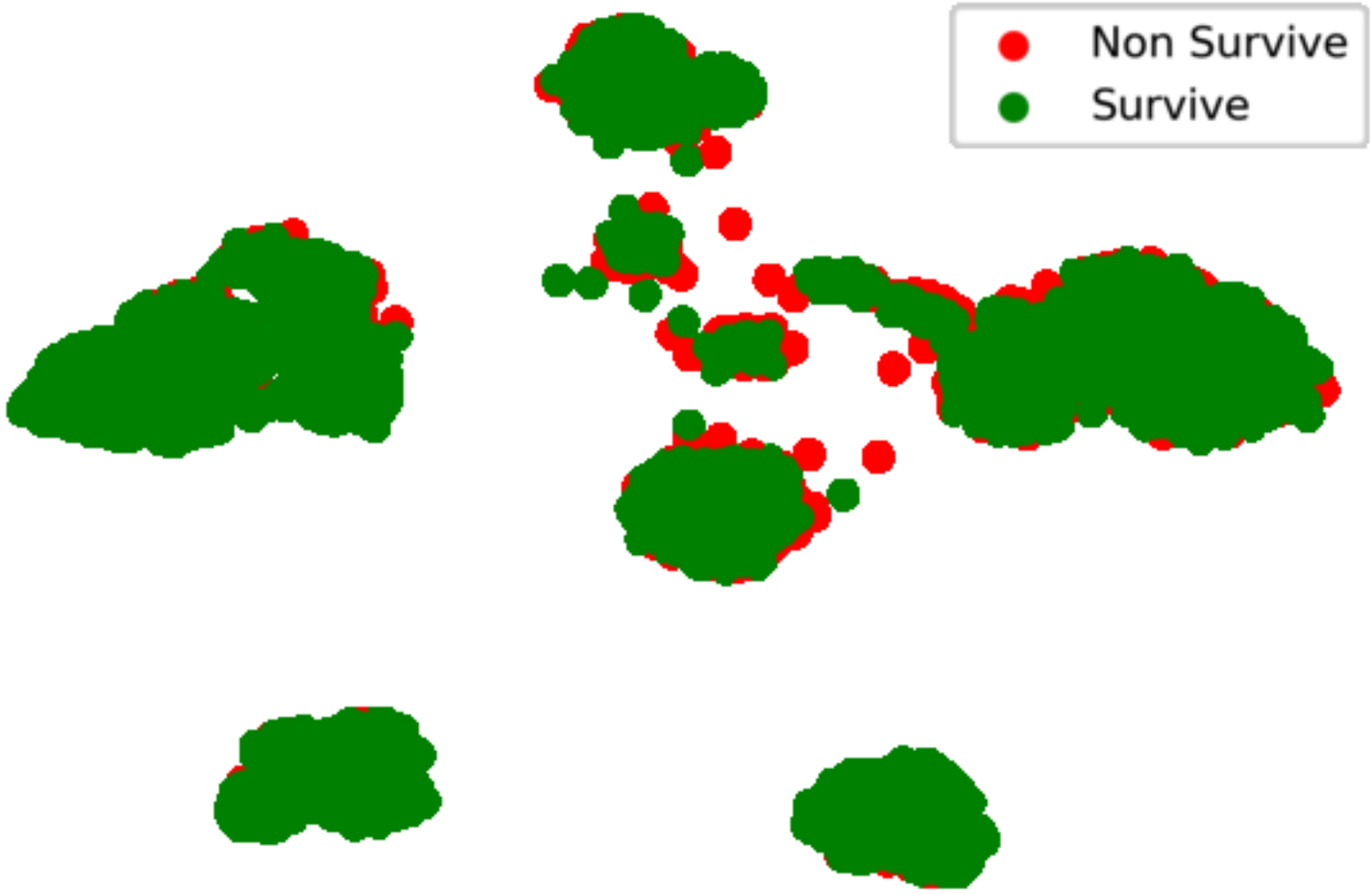


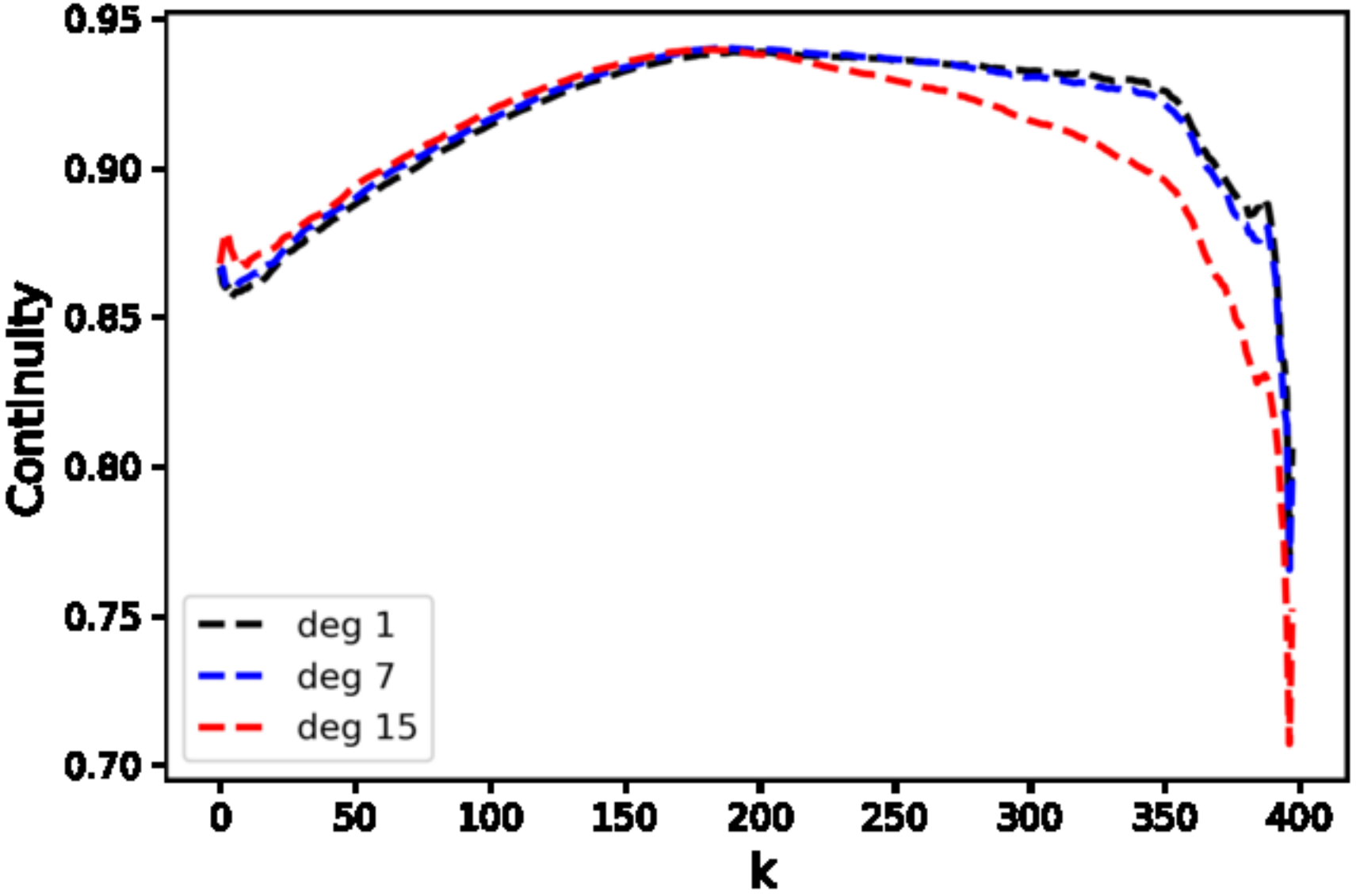


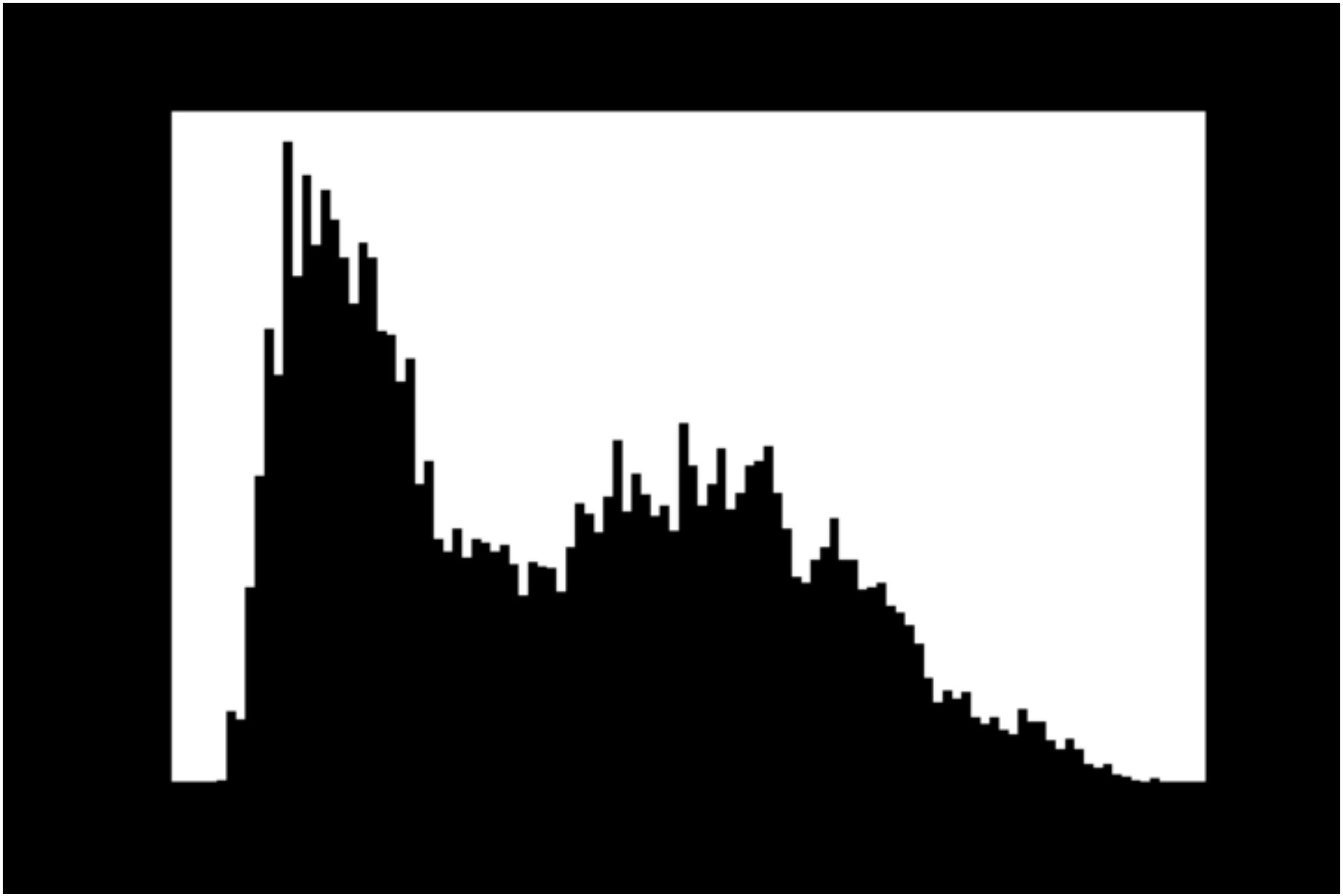


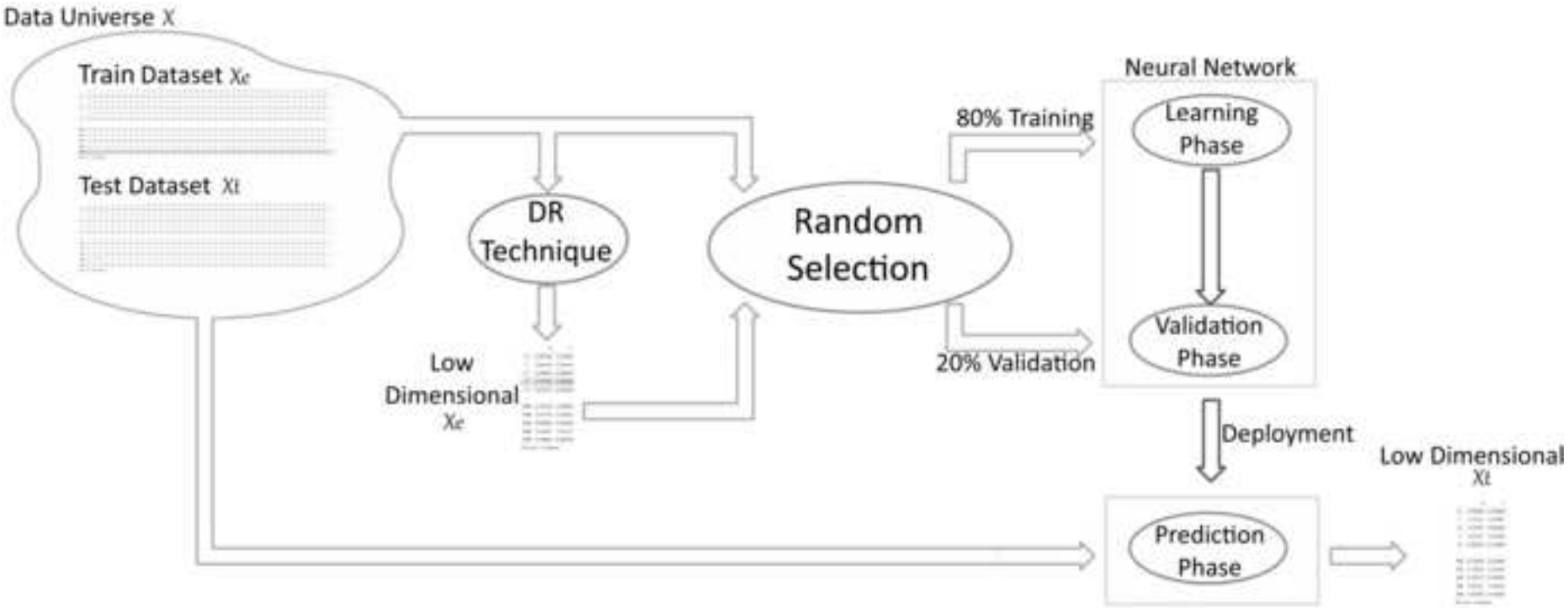


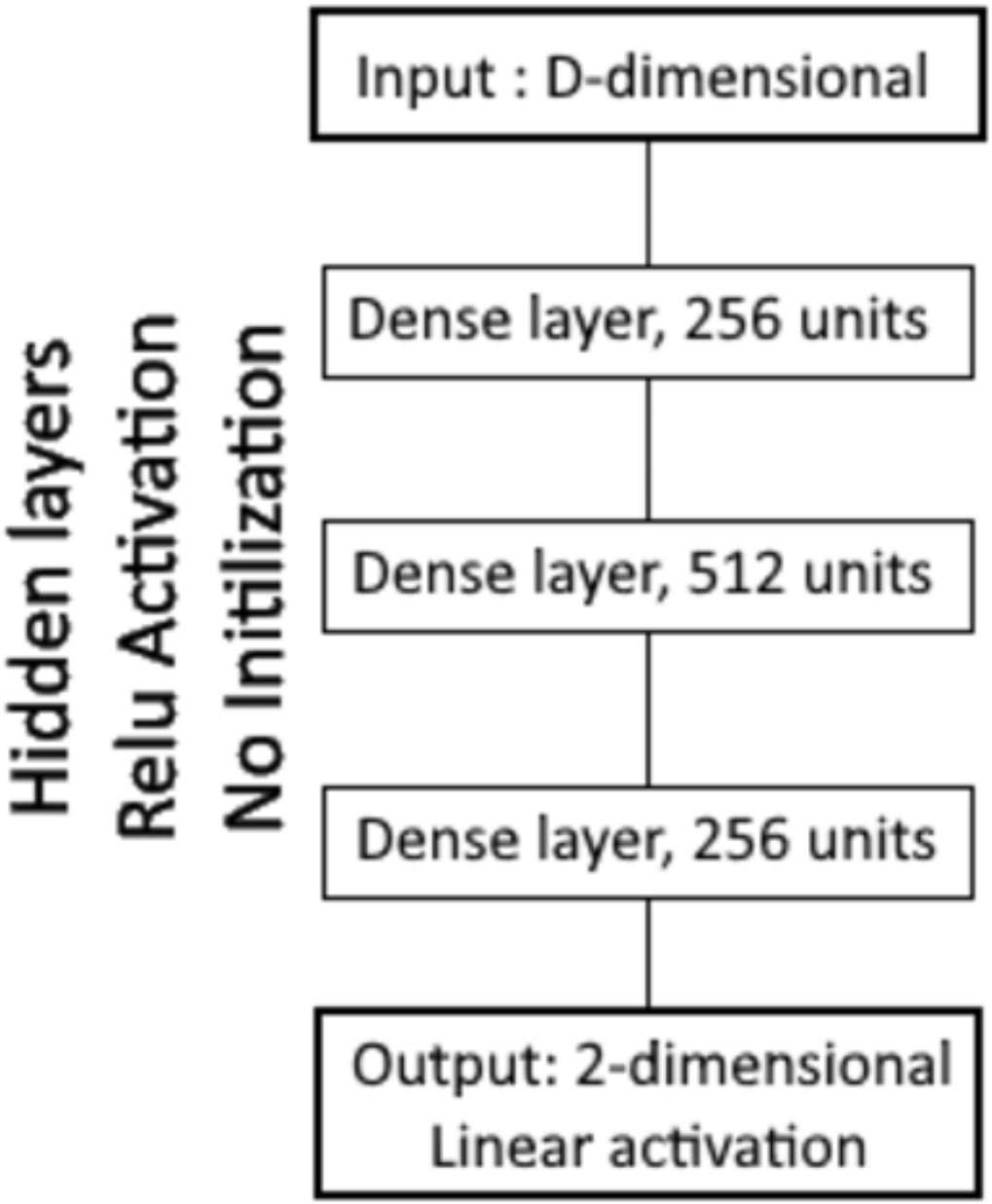


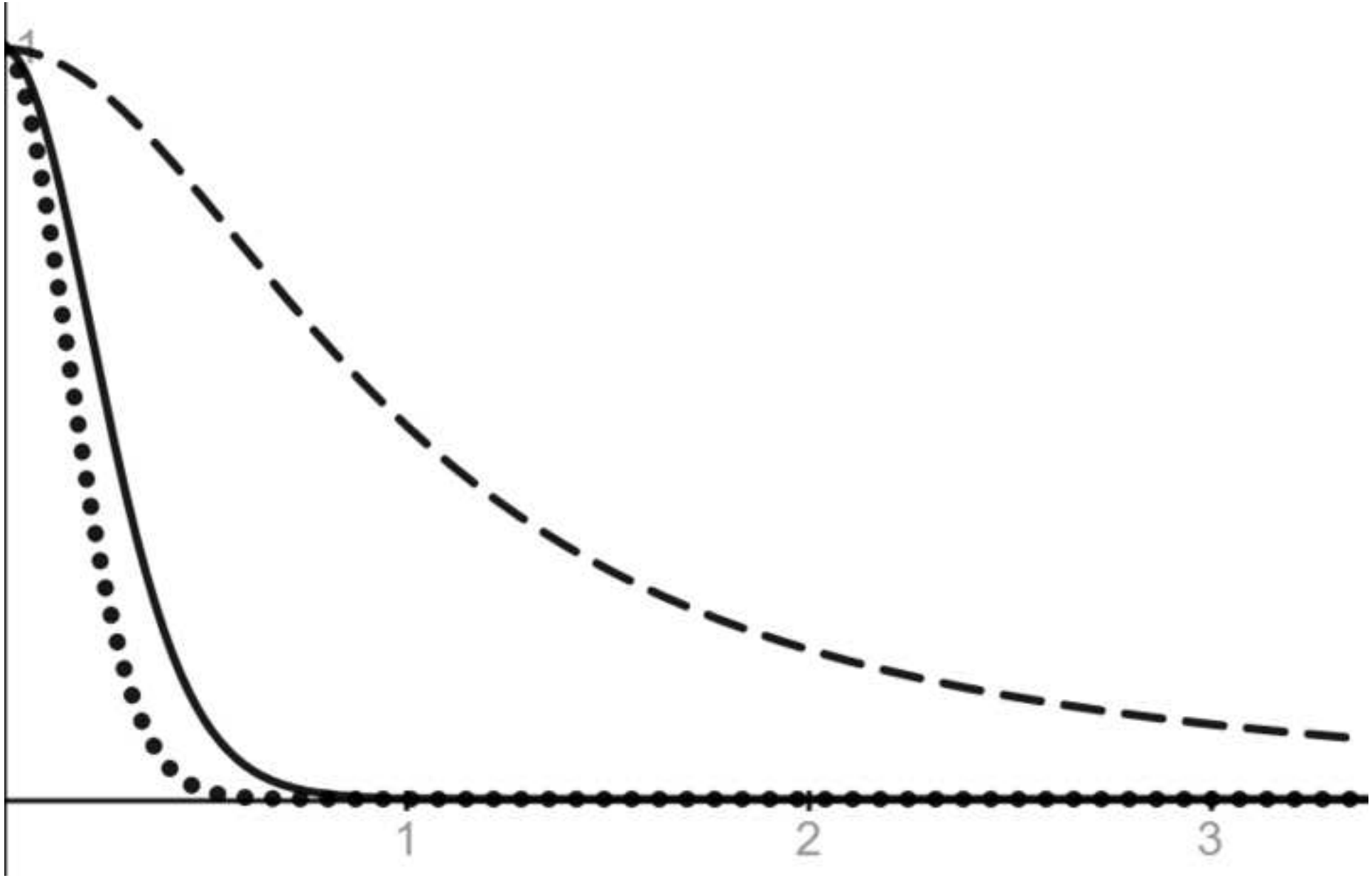


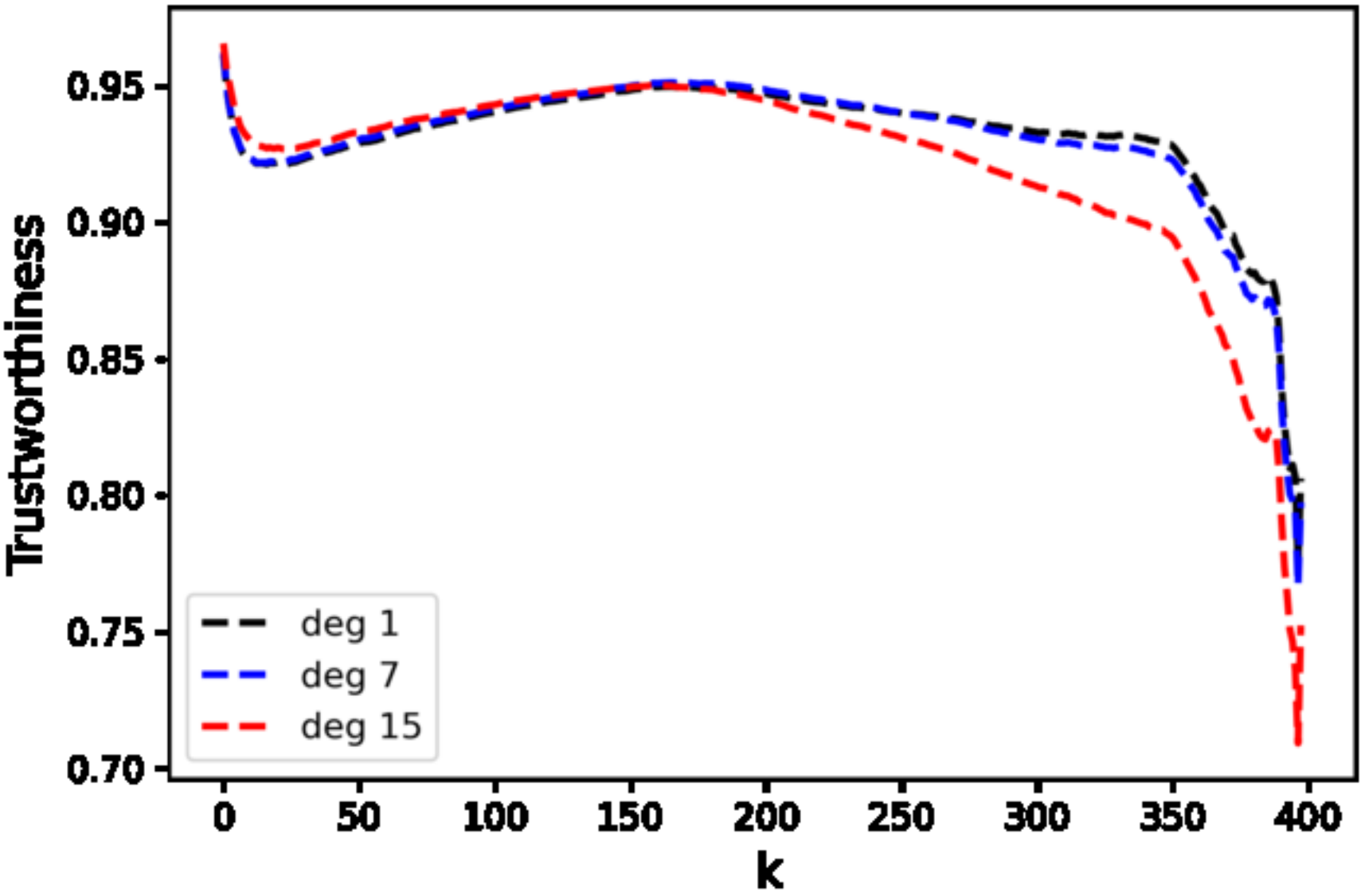


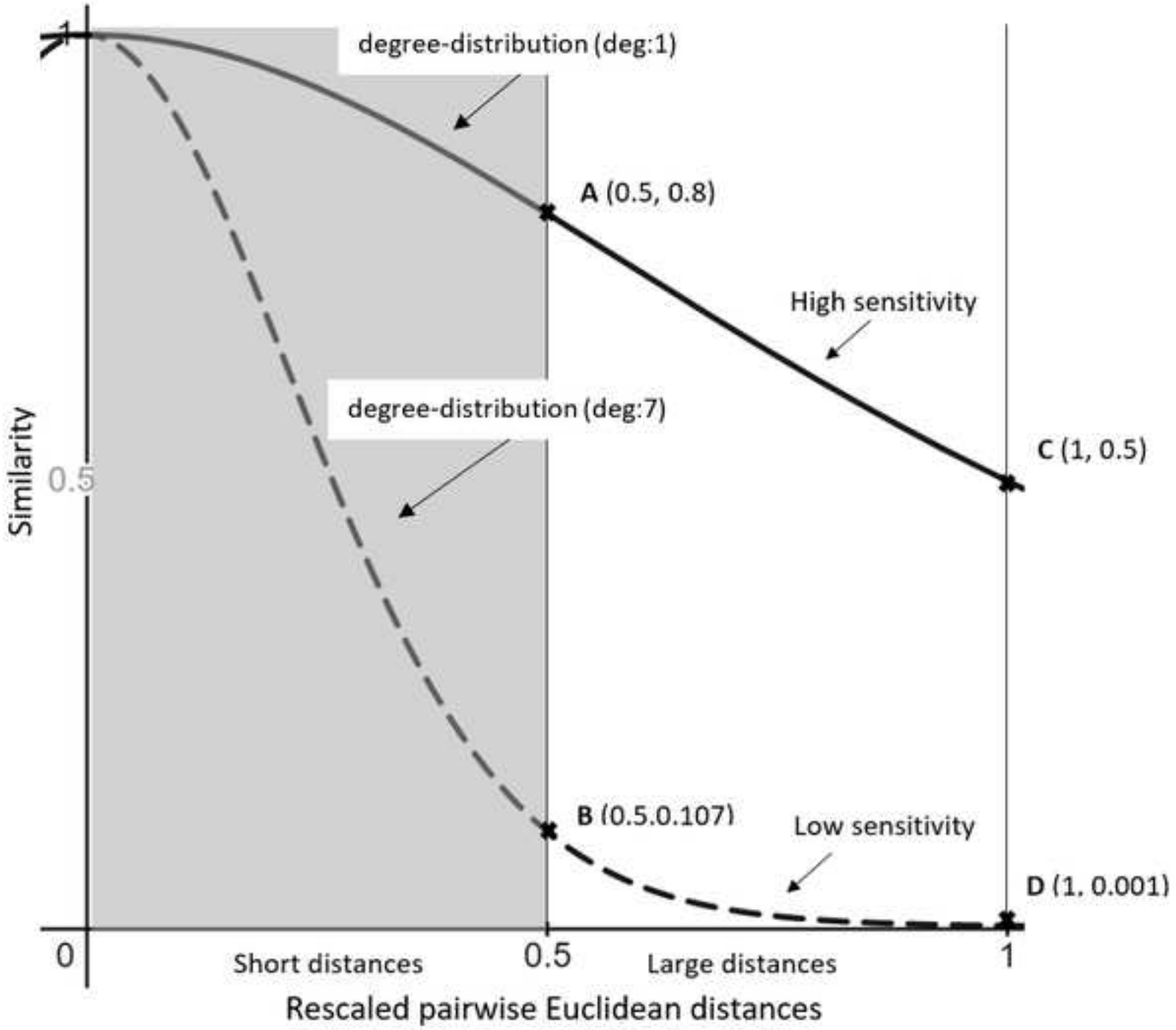


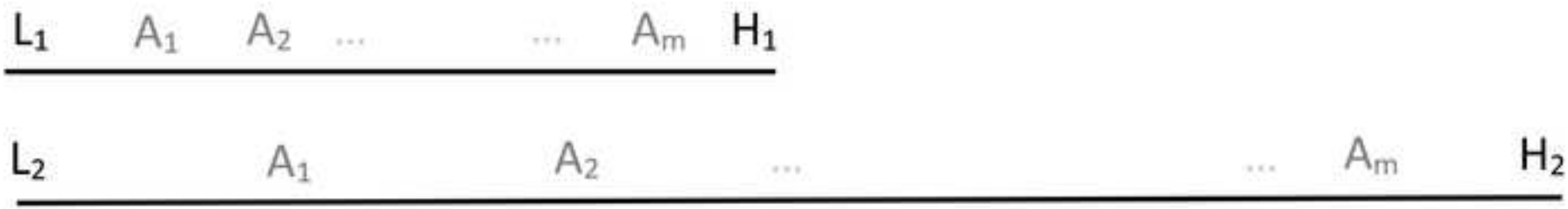


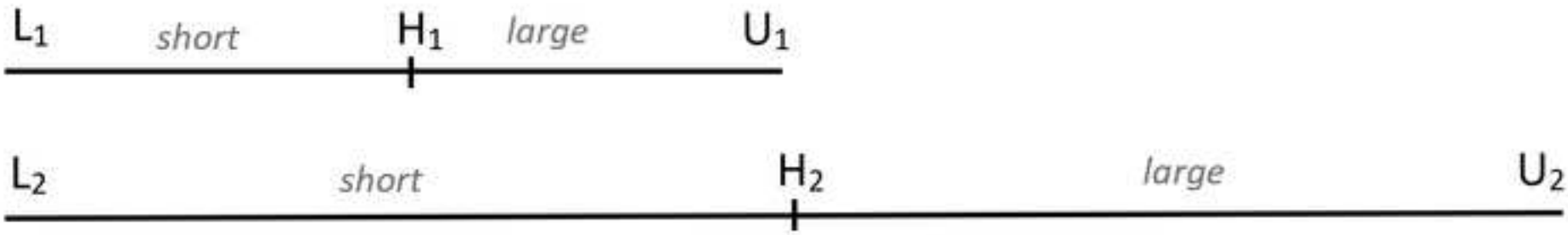












Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Credit Author Statement

I do confirm that all authors have contributed to the paper.

Laureta Hajderanj has proposed the methodology, validation, former analyses, writing and visualization.

Daqing Chen and Sandra Dudley have been part of supervision of the is project and further support on reviewing and editing the drafts.

Guillaume Gilloppe and Baptiste Sivy have contributed to software and providing resources of the project.