

Eight-objective hyperparameter optimization of t-SNE dimensionality reduction using cluster quality and neighborhood preservation criteria

Vadim Romanuke^{1,*†} and Svitlana Merinova^{1,†}

¹ Vinnytsia Institute of Trade and Economics of the State University of Trade and Economics, Soborna 87, 21050, Vinnytsia, Ukraine

Abstract

In nonlinear dimensionality reduction, the quality of a t-SNE embedding strongly depends on the perplexity factor (PF), exaggeration factor (EF), learning rate (LR), and the Barnes – Hut approximation parameter (BHAP). When the dataset is labeled, the hyperparameter quadruple of PF, EF, LR, BHAP, producing the minimum Kullback – Leibler (KL) divergence, does not necessarily provide the most informative embedding for visualization or subsequent data analysis. This induces an eight-objective optimization problem, where KL divergence, Davies – Bouldin index, and runtime are minimized, whereas Trustworthiness, Continuity, Silhouette coefficient, Calinski – Harabasz index, and classification accuracy are maximized. The solution is Pareto-optimal, which is determined by the suggested algorithm and obtained in a few iterations to several tens of iterations with linear time complexity.

Keywords

low-dimensional embedding, perplexity, exaggeration, learning rate, Barnes – Hut approximation parameter, discrete optimization, Pareto-optimal set, Pareto refinement

1. Introduction

The rapid and continuing growth of high-dimensional data in science, engineering, finance, medicine, and information systems has been raising the importance of dimensionality reduction methods so that reducing dimensionality today is a common part in data mining and knowledge engineering [1, 2]. Such methods transform big data or overly-attributed data (having too many features) from a high-dimensional feature space into a low-dimensional representation while preserving the most informative properties of the original dataset [3]. Low-dimensional embeddings facilitate visualization, exploratory data analysis, cluster interpretation, anomaly detection, and the development of machine learning models [4]. Among nonlinear dimensionality reduction techniques, t-distributed Stochastic Neighbor Embedding (t-SNE) remains one of the most widely used approaches for constructing two- and three-dimensional visualizations of complex datasets [5, 6]. Unlike linear methods such as Principal Component Analysis (PCA) [7], t-SNE is designed to preserve local neighborhood relationships and therefore often produces considerably better visual representations of datasets containing nonlinear structures [8]. Recent reviews and application studies confirm that t-SNE remains an important tool for visualization and exploratory analysis across many application domains [7, 9]. The quality of a t-SNE embedding strongly depends on several hyperparameters, including the perplexity factor (PF), exaggeration factor (EF), learning rate (LR), and the Barnes – Hut approximation parameter (BHAP) [10]. Although practical recommendations for selecting these hyperparameters do exist, their values are still commonly chosen empirically or by trial and error. Moreover, recent studies have demonstrated that while automatic hyperparameter selection can improve embedding quality, this does not perform optimally

¹ADT-ISMDDC 2026: *Advanced Digital Technologies: Intelligent Systems, Modeling, and Data-Driven Control*, September 22–24, 2026, Odesa, Ukraine

* Corresponding author.

† These authors contributed equally.

✉ romanukevadimv@gmail.com (V. V. Romanuke); s.merinova@vtei.edu.ua (S. V. Merinova)

id 0000-0001-9638-9572 (V. V. Romanuke); 0000-0001-6563-5320 (S. V. Merinova)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

for every dataset [11]. Most existing approaches evaluate the quality of a t-SNE embedding primarily through minimizing the Kullback — Leibler (KL) divergence between neighborhood probability distributions in the original and embedded spaces [1, 8]. However, when class labels are available within the dataset (i. e., the dataset objects or points are labeled), additional characteristics become important, including cluster compactness [12], cluster separation [13], neighborhood preservation [14, 15], and classification performance [16]. Consequently, the hyperparameter vector (consisting of PF, EF, LR, BHAP) producing the minimum KL divergence does not necessarily provide the most informative embedding for visualization or subsequent data analysis.

2. Relevance of the Research

It is clear that the selection of t-SNE hyperparameters (PF, EF, LR, BHAP) is an important practical problem since the embedding produced by t-SNE depends quite differently on those user-defined hyperparameters. When the dataset is preliminarily labeled, the quality of a low-dimensional embedding can be assessed from the abovementioned complementary perspectives. Cluster compactness and separation may be evaluated using the Silhouette coefficient [12, 17], the Davies — Bouldin index [11, 13], and the Calinski — Harabasz index [18], which assess the general geometric quality of the resulting clusters. Preservation of local neighborhood relationships can be quantified by the Trustworthiness and Continuity measures [11, 15], whereas the practical usefulness of the embedding for supervised learning may be estimated through the classification accuracy of a k -nearest neighbors (k -NN) classifier trained on the embedded data [16]. The k -NN classifier reflects the practical usefulness of the embedding for supervised learning [19]. Since these criteria characterize different properties of the embedding, they along with the KL divergence may lead to conflicting evaluations of the same hyperparameter configuration [10, 20, 21].

The above considerations indicate that selecting t-SNE hyperparameters should be formulated as a multi-objective optimization problem rather than as the minimization of a single objective function (the KL divergence in this case) [22, 23]. Instead of searching for one universally optimal hyperparameter vector [20, 24], it is more reasonable to identify a set of Pareto-optimal solutions [25, 26] representing different trade-offs among neighborhood preservation, cluster quality, classification performance, and the internal optimization criterion of the t-SNE algorithm. Such an approach must provide a more informative and richer dimensionality reduction as for labeled datasets, as well as for non-labeled datasets, whose subsequent application (via their embeddings) may relate to classification or regression analysis.

3. Statement of the Problem

This paper investigates the problem of selecting t-SNE hyperparameters from a multi-objective perspective. Instead of minimizing solely the final KL divergence, the eight complementary embedding quality criteria are considered simultaneously: the KL divergence (denoted by d_{KL} for further consideration), Trustworthiness (u_{trust}), Continuity (u_{cont}), the Silhouette coefficient (b_{SC}), the Davies — Bouldin index (v_{DB}), the Calinski — Harabasz index (v_{CH}), the classification accuracy of a k -NN classifier ($a_{k\text{-NN}}$), and the runtime (t_{run}) evaluating computational cost. Namely this hierarchy of the eight criteria creates a logical progression (Figure 1), which is much easier to understand than a mixed list. However, no hierarchic weighting of the criteria will be applied further because no additional information is assumed to be available except for the dataset and potential labeling of its objects.

The KL divergence should come first because it is the native optimization objective of t-SNE. The algorithm itself minimizes the KL divergence during embedding construction. Everything else evaluates the resulting embedding from an external viewpoint.

Trustworthiness and Continuity are the most closely related to the purpose of dimensionality reduction [11, 15]. They assess how qualitatively the embedding preserves the original neighborhoods and how many false neighbors are introduced. Importantly, they do not require class labels,

so they are applicable to both supervised and unsupervised datasets. Conceptually, they are the first layer of external evaluation after the internal KL objective.

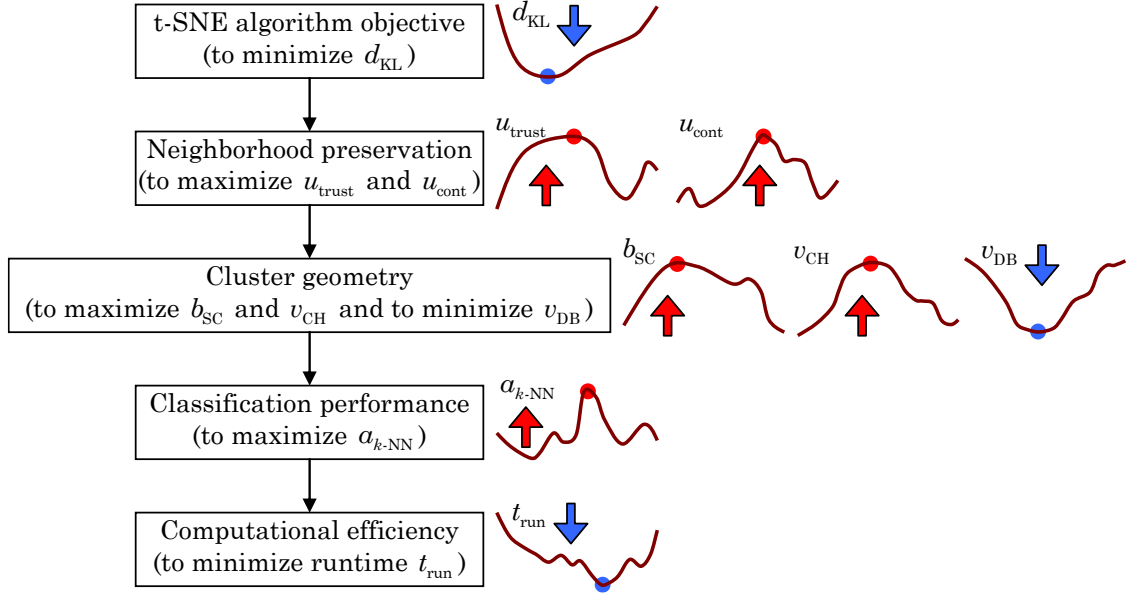


Figure 1: The progression and connection of the eight criteria in optimally selecting t-SNE hyperparameters PF, EF, LR, BHAP (the curves are shown schematically, not from real examples).

The next three cluster-quality metrics all evaluate how well the classes are geometrically separated in the embedding, naturally forming a group: the Silhouette coefficient combines compactness and separation [12, 17], the Davies – Bouldin index penalizes compact clusters that are too close together [13], and the Calinski – Harabasz index compares between-cluster and within-cluster dispersion [18]. Since they all describe cluster geometry, using the class labels, keeping them together makes the methodology easier to follow.

Although the k -NN classifier accuracy also uses the class labels, it measures something different. The previous three metrics assess geometry, and the k -NN classifier accuracy assesses utility of using the embedding for classification. So it is reasonable to discuss it after the geometric criteria.

The runtime should come last because it is fundamentally different from the others. The first seven objectives evaluate quality. The runtime evaluates computational cost, but it does not say whether an embedding is good or bad; it says how expensive it was to obtain. Keeping the runtime last clearly separates effectiveness from efficiency, which is a common distinction in optimization studies. Hence, the goal is to suggest a method of determining the best hyperparameter vector as a Pareto-optimal solution to the eight-objective optimization problem of t-SNE dimensionality reduction. To achieve the goal, the mathematical strict formulation of the problem will be given first, including plausible constraints. Then Pareto-optimal selection of t-SNE hyperparameters will be formalized along with an algorithm to implement the selection. The refinement of Pareto-optimal hyperparameters to a single quadruple of the best hyperparameters PF, EF, LR, BHAP is to be given next. Finally, the results of t-SNE examples on big data set instances should be briefly discussed, whereupon conclusions on the findings are to be made.

4. Research Results

4.1. Eight-objective optimization problem

Denote the quadruple of t-SNE hyperparameters PF, EF, LR, BHAP by α , x , λ , θ , respectively. Denote their respective domains by Λ , X , Λ , and Θ . Hence, formally the eight-objective optimization problem of t-SNE dimensionality reduction is to be solved on Cartesian product

$$\mathbf{A} \times \mathbf{X} \times \mathbf{\Lambda} \times \mathbf{\Theta}, \quad (1)$$

where eight optimization problems

$$\begin{aligned} \min_{\mathbf{A} \times \mathbf{X} \times \mathbf{\Lambda} \times \mathbf{\Theta}} d_{\text{KL}}(\alpha, x, \lambda, \theta), \quad \max_{\mathbf{A} \times \mathbf{X} \times \mathbf{\Lambda} \times \mathbf{\Theta}} u_{\text{trust}}(\alpha, x, \lambda, \theta), \quad \max_{\mathbf{A} \times \mathbf{X} \times \mathbf{\Lambda} \times \mathbf{\Theta}} u_{\text{cont}}(\alpha, x, \lambda, \theta), \quad \max_{\mathbf{A} \times \mathbf{X} \times \mathbf{\Lambda} \times \mathbf{\Theta}} b_{\text{SC}}(\alpha, x, \lambda, \theta), \\ \min_{\mathbf{A} \times \mathbf{X} \times \mathbf{\Lambda} \times \mathbf{\Theta}} v_{\text{DB}}(\alpha, x, \lambda, \theta), \quad \max_{\mathbf{A} \times \mathbf{X} \times \mathbf{\Lambda} \times \mathbf{\Theta}} v_{\text{CH}}(\alpha, x, \lambda, \theta), \quad \max_{\mathbf{A} \times \mathbf{X} \times \mathbf{\Lambda} \times \mathbf{\Theta}} a_{k\text{-NN}}(\alpha, x, \lambda, \theta), \quad \min_{\mathbf{A} \times \mathbf{X} \times \mathbf{\Lambda} \times \mathbf{\Theta}} t_{\text{run}}(\alpha, x, \lambda, \theta) \end{aligned} \quad (2)$$

are to be simultaneously solved. For practical computation, however, four-dimensional domain (1) will be substituted by a four-dimensional lattice

$$R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}}) = \{\alpha_i\}_{i=1}^{N_{\mathbf{A}}} \times \{x_j\}_{j=1}^{N_{\mathbf{X}}} \times \{\lambda_k\}_{k=1}^{N_{\mathbf{\Lambda}}} \times \{\theta_l\}_{l=1}^{N_{\mathbf{\Theta}}}, \quad (3)$$

where integers $N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}}$ are taken such that lattice (3) would cover effectively initial domain (1); the total number of lattice (3) points (quadruples) is $N = N_{\mathbf{A}} \cdot N_{\mathbf{X}} \cdot N_{\mathbf{\Lambda}} \cdot N_{\mathbf{\Theta}}$ and lattice (3) is $R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}}) = \{H_n\}_{n=1}^N$. Thereupon the eight-objective optimization problem by (2), (3) becomes a combinatorial (discrete) multi-objective hyperparameter optimization, where eight discrete optimization problems

$$\begin{aligned} \min_{R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}})} d_{\text{KL}}(\alpha, x, \lambda, \theta), \quad \max_{R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}})} u_{\text{trust}}(\alpha, x, \lambda, \theta), \quad \max_{R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}})} u_{\text{cont}}(\alpha, x, \lambda, \theta), \\ \max_{R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}})} b_{\text{SC}}(\alpha, x, \lambda, \theta), \quad \min_{R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}})} v_{\text{DB}}(\alpha, x, \lambda, \theta), \quad \max_{R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}})} v_{\text{CH}}(\alpha, x, \lambda, \theta), \\ \max_{R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}})} a_{k\text{-NN}}(\alpha, x, \lambda, \theta), \quad \min_{R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}})} t_{\text{run}}(\alpha, x, \lambda, \theta) \end{aligned} \quad (4)$$

are to be simultaneously solved. It is worth noting that four-dimensional lattice (3) is not necessarily to be regular, i. e. one or more one-dimensional domains ($\mathbf{A}$, $\mathbf{X}$, $\mathbf{\Lambda}$, or $\mathbf{\Theta}$) may be discretized (sampled) non-uniformly.

4.2. Pareto-optimal selection of t-SNE hyperparameters

As it is well known, multi-objective optimization problems commonly do not have exact solutions. And it is clear that eight-objective optimization problem (4) over lattice (3) is very likely to not have exact solutions either. Henceforward, it is solved only as a Pareto-optimal quadruple

$$H^{**} = \{\alpha^{**}, x^{**}, \lambda^{**}, \theta^{**}\} \in R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}}) \quad (5)$$

of t-SNE hyperparameters PF, EF, LR, BHAP, which is actually a four-dimensional point of lattice (3). Define a Pareto-optimal set

$$P^* = \{H_m^*\}_{m=1}^M = \{\{\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*\}\}_{m=1}^M \subset R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}}) \quad (6)$$

by $H^{**} \in P^*$ as follows. Quadruple

$$H^* = \{\alpha^*, x^*, \lambda^*, \theta^*\} \in R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}}) \quad (7)$$

is Pareto-optimal if no quadruple

$$H^{(1)} = \{\alpha^{(1)}, x^{(1)}, \lambda^{(1)}, \theta^{(1)}\} \in R(N_{\mathbf{A}}, N_{\mathbf{X}}, N_{\mathbf{\Lambda}}, N_{\mathbf{\Theta}}) \quad (8)$$

exists such that inequalities

$$\begin{aligned} d_{\text{KL}}(\alpha^*, x^*, \lambda^*, \theta^*) & d_{\text{KL}}(\alpha^{(1)}, x^{(1)}, \lambda^{(1)}, \theta^{(1)}), \quad u_{\text{trust}}(\alpha^*, x^*, \lambda^*, \theta^*) & u_{\text{trust}}(\alpha^{(1)}, x^{(1)}, \lambda^{(1)}, \theta^{(1)}), \\ u_{\text{cont}}(\alpha^*, x^*, \lambda^*, \theta^*) & u_{\text{cont}}(\alpha^{(1)}, x^{(1)}, \lambda^{(1)}, \theta^{(1)}), \quad b_{\text{SC}}(\alpha^*, x^*, \lambda^*, \theta^*) & b_{\text{SC}}(\alpha^{(1)}, x^{(1)}, \lambda^{(1)}, \theta^{(1)}), \\ v_{\text{DB}}(\alpha^*, x^*, \lambda^*, \theta^*) & v_{\text{DB}}(\alpha^{(1)}, x^{(1)}, \lambda^{(1)}, \theta^{(1)}), \quad v_{\text{CH}}(\alpha^*, x^*, \lambda^*, \theta^*) & v_{\text{CH}}(\alpha^{(1)}, x^{(1)}, \lambda^{(1)}, \theta^{(1)}), \\ a_{k\text{-NN}}(\alpha^*, x^*, \lambda^*, \theta^*) & a_{k\text{-NN}}(\alpha^{(1)}, x^{(1)}, \lambda^{(1)}, \theta^{(1)}), \quad t_{\text{run}}(\alpha^*, x^*, \lambda^*, \theta^*) & t_{\text{run}}(\alpha^{(1)}, x^{(1)}, \lambda^{(1)}, \theta^{(1)}) \end{aligned} \quad (9)$$

are true and one of inequalities (9) holds strictly.

An algorithm of implementing the Pareto-optimal selection of t-SNE hyperparameters in accordance with (7) – (9) is presented in Figure 2. The algorithm iteratively forms Pareto-optimal set (6) by eliminating dominated quadruples from consideration. In particular, quadruple (7) is dominated by quadruple (8) if inequalities (9) hold and one of them holds strictly.

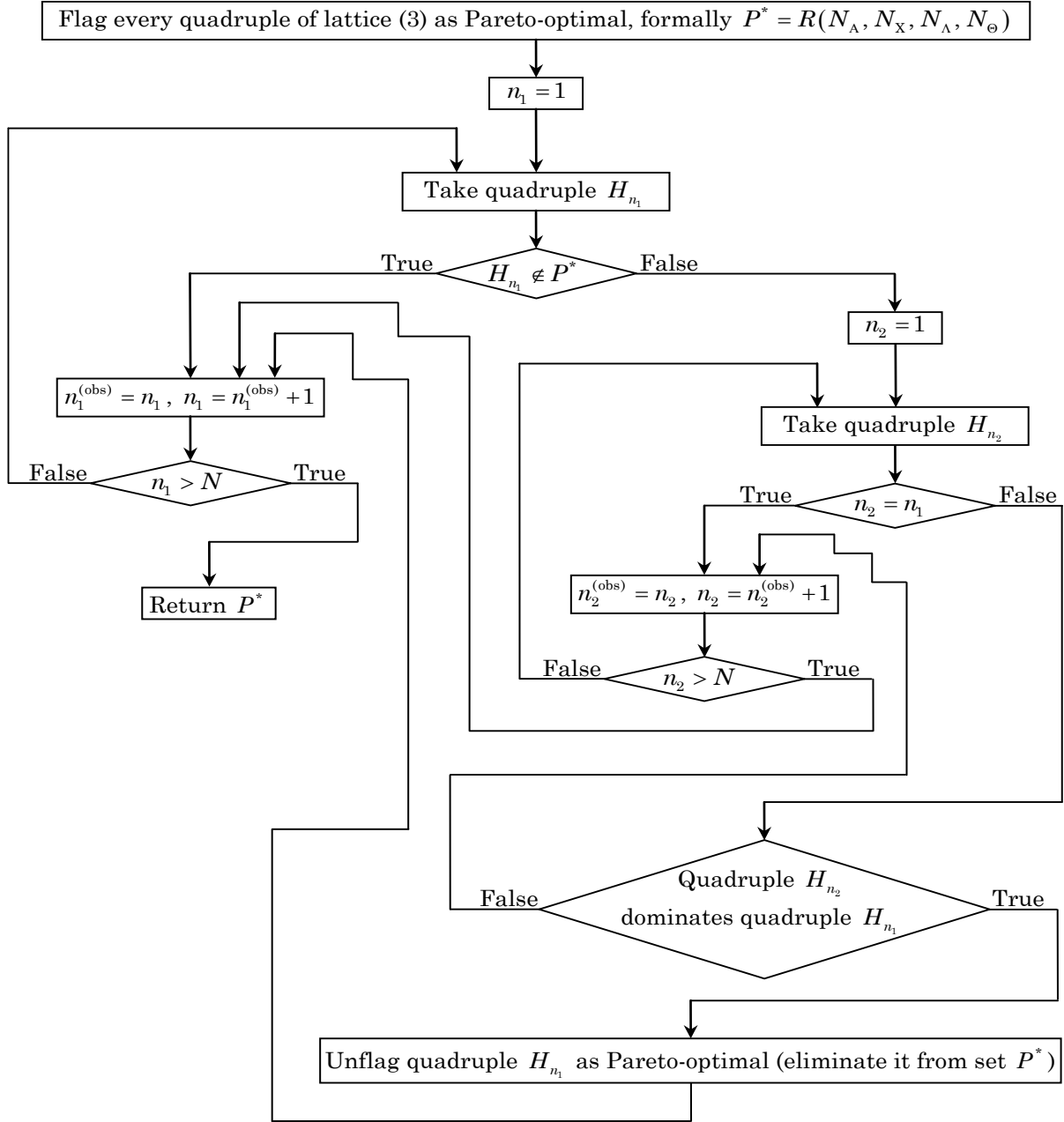


Figure 2: The algorithm of determining Pareto-optimal set (6).

Computational simulations on various datasets reveal that Pareto-optimal set (6) is never of a few quadruples, let alone a single quadruple (5). Typically, Pareto-optimal set (6) constitutes at least about 10 % or so of the total number of lattice (3) points. For example, if

$$A = \{\alpha_i\}_{i=1}^{10} = \{5j\}_{j=1}^{10}, \quad X = \{x_j\}_{j=1}^{11} = \{2j-1\}_{j=1}^{11}, \quad \Lambda = \{\lambda_k\}_{k=1}^5 = \{50 \cdot 2^{k-1}\}_{k=1}^5, \quad \Theta = \{\theta_l\}_{l=1}^8 = \{0.1\}_{l=1}^8 \quad (10)$$

for the well-known benchmarking dataset “Iris flower” consisting of 150 objects (50 samples from each of three species of Iris — Iris setosa, Iris virginica and Iris versicolor) having four attributes each, then the application of the Pareto-optimal set algorithm renders the initial number of 4400 quadruples into 550 to 620 Pareto-optimal quadruples. While this is a good step toward a solution of the

respective eight-objective optimization problem, it is insufficient to stop there as a single quadruple (5) of the best t-SNE hyperparameters must be selected somehow. And this is just a toy dataset, whereas a t-SNE embedding of a real-world big data set instance is typically much complicated and voluminous, and so optimizing the t-SNE algorithm by just determining a bunch of Pareto-optimal quadruples would be quite ineffective. Another problem is that the t-SNE embedding is a random object [1, 5, 8, 27]. This is the consequence of that the optimization problem solved by t-SNE is highly non-convex [1, 28]. The t-SNE algorithm minimizes the KL divergence, but this objective has many local minima. The embedding starts from a random initialization (or sometimes from PCA, depending on the settings), and gradient descent converges to one of these local minima [8, 21, 27]. This means that the t-SNE algorithm, run under the same quadruple of its hyperparameters, every time returns a slightly different embedding, whose quality criteria are factually random as well, but they vary in relatively narrow ranges. Visual comparison of embeddings is hardly consistent or effective [7, 20, 22, 23, 28].

4.3. Refinement of Pareto-optimal set

To select a single (and the best) Pareto-optimal quadruple (5) from Pareto-optimal set (6), some additional criteria should be applied. In particular, a set of eight threshold values

$$d_{\text{KL}}^{(\max)}, u_{\text{trust}}^{(\min)}, u_{\text{cont}}^{(\min)}, b_{\text{SC}}^{(\min)}, v_{\text{DB}}^{(\max)}, v_{\text{CH}}^{(\min)}, a_{k\text{-NN}}^{(\min)}, t_{\text{run}}^{(\max)} \quad (11)$$

for the eight objectives is introduced, and a refined Pareto-optimal set

$$P^{**} = \{H_m^{**}\}_{m=1}^{M^*} = \{(\alpha_m^{**}, x_m^{**}, \lambda_m^{**}, \theta_m^{**})\}_{m=1}^{M^*} \subset P^* \subset R(N_A, N_X, N_\Lambda, N_\Theta) \quad (12)$$

is formed. Pareto-optimal quadruple $H_m^* = \{\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*\}$ becomes an element of subset (12) if the eight inequalities

$$\begin{aligned} d_{\text{KL}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*) &< d_{\text{KL}}^{(\max)}, u_{\text{trust}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*) > u_{\text{trust}}^{(\min)}, u_{\text{cont}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*) > u_{\text{cont}}^{(\min)}, \\ b_{\text{SC}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*) &> b_{\text{SC}}^{(\min)}, v_{\text{DB}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*) < v_{\text{DB}}^{(\max)}, v_{\text{CH}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*) > v_{\text{CH}}^{(\min)}, \\ a_{k\text{-NN}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*) &> a_{k\text{-NN}}^{(\min)}, t_{\text{run}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*) < t_{\text{run}}^{(\max)} \end{aligned} \quad (13)$$

hold simultaneously. Primarily, threshold values (11) can be set to the average values:

$$\begin{aligned} d_{\text{KL}}^{(\max)} &= \frac{1}{M} \cdot \sum_{m=1}^M d_{\text{KL}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*), u_{\text{trust}}^{(\min)} = \frac{1}{M} \cdot \sum_{m=1}^M u_{\text{trust}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*), \\ u_{\text{cont}}^{(\min)} &= \frac{1}{M} \cdot \sum_{m=1}^M u_{\text{cont}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*), b_{\text{SC}}^{(\min)} = \frac{1}{M} \cdot \sum_{m=1}^M b_{\text{SC}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*), \\ v_{\text{DB}}^{(\max)} &= \frac{1}{M} \cdot \sum_{m=1}^M v_{\text{DB}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*), v_{\text{CH}}^{(\min)} = \frac{1}{M} \cdot \sum_{m=1}^M v_{\text{CH}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*), \\ a_{k\text{-NN}}^{(\min)} &= \frac{1}{M} \cdot \sum_{m=1}^M a_{k\text{-NN}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*), t_{\text{run}}^{(\max)} = \frac{1}{M} \cdot \sum_{m=1}^M t_{\text{run}}(\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*). \end{aligned} \quad (14)$$

Then threshold values (11) are iteratively varied in their appropriate directions until set (12) becomes a singleton. Thus, thresholds $d_{\text{KL}}^{(\max)}, v_{\text{DB}}^{(\max)}, t_{\text{run}}^{(\max)}$ are iteratively decreased as

$$d_{\text{KL}}^{(\max)(\text{obs})} = d_{\text{KL}}^{(\max)}, d_{\text{KL}}^{(\max)} = d_{\text{KL}}^{(\max)(\text{obs})} - \delta d_{\text{KL}}^{(\max)(\text{obs})}, \quad (15)$$

$$v_{\text{DB}}^{(\max)(\text{obs})} = v_{\text{DB}}^{(\max)}, v_{\text{DB}}^{(\max)} = v_{\text{DB}}^{(\max)(\text{obs})} - \delta v_{\text{DB}}^{(\max)(\text{obs})}, \quad (16)$$

$$t_{\text{run}}^{(\max)(\text{obs})} = t_{\text{run}}^{(\max)}, t_{\text{run}}^{(\max)} = t_{\text{run}}^{(\max)(\text{obs})} - \delta t_{\text{run}}^{(\max)(\text{obs})}, \quad (17)$$

and thresholds $u_{\text{trust}}^{(\min)}, u_{\text{cont}}^{(\min)}, b_{\text{SC}}^{(\min)}, v_{\text{CH}}^{(\min)}, a_{k\text{-NN}}^{(\min)}$ are iteratively increased as

$$u_{\text{trust}}^{(\min)(\text{obs})} = u_{\text{trust}}^{(\min)}, u_{\text{trust}}^{(\min)} = u_{\text{trust}}^{(\min)(\text{obs})} + \delta u_{\text{trust}}^{(\min)(\text{obs})}, \quad (18)$$

$$u_{\text{cont}}^{(\min)(\text{obs})} = u_{\text{cont}}^{(\min)}, u_{\text{cont}}^{(\min)} = u_{\text{cont}}^{(\min)(\text{obs})} + \delta u_{\text{cont}}^{(\min)(\text{obs})}, \quad (19)$$

$$b_{\text{SC}}^{(\min)(\text{obs})} = b_{\text{SC}}^{(\min)}, b_{\text{SC}}^{(\min)} = b_{\text{SC}}^{(\min)(\text{obs})} + \delta b_{\text{SC}}^{(\min)(\text{obs})}, \quad (20)$$

$$v_{\text{CH}}^{(\min)(\text{obs})} = v_{\text{CH}}^{(\min)}, v_{\text{CH}}^{(\min)} = v_{\text{CH}}^{(\min)(\text{obs})} + \delta v_{\text{CH}}^{(\min)(\text{obs})}, \quad (21)$$

$$a_{k\text{-NN}}^{(\min)(\text{obs})} = a_{k\text{-NN}}^{(\min)}, a_{k\text{-NN}}^{(\min)} = a_{k\text{-NN}}^{(\min)(\text{obs})} + \delta a_{k\text{-NN}}^{(\min)(\text{obs})}, \quad (22)$$

for some $\delta > 0$. If at least one of inequalities (13) turns false by starting with (14) and proceeding with (15) – (22), Pareto-optimal quadruple $H_m^* = \{\alpha_m^*, x_m^*, \lambda_m^*, \theta_m^*\}$ does not go into refined Pareto-optimal set (12). To get rid of uncertainty completely, this set is required to be a singleton. The selection of δ , obviously, influences the speed of obtaining singleton

$$P^* = \{H^*\} = \{\{\alpha^*, x^*, \lambda^*, \theta^*\}\} \quad (23)$$

at which the t-SNE embedding receives the eight quality values

$$d_{\text{KL}}^{**}, u_{\text{trust}}^{**}, u_{\text{cont}}^{**}, b_{\text{SC}}^{**}, v_{\text{DB}}^{**}, v_{\text{CH}}^{**}, a_{k\text{-NN}}^{**}, t_{\text{run}}^{**}. \quad (24)$$

Henceforward, the optimization workflow presented in Figure 3 consists of successively defining four-dimensional lattice (3), determining Pareto-optimal set (6), and refining it to a single hyperparameter quadruple in Pareto singleton (23). The time complexity of the refinement is linear.

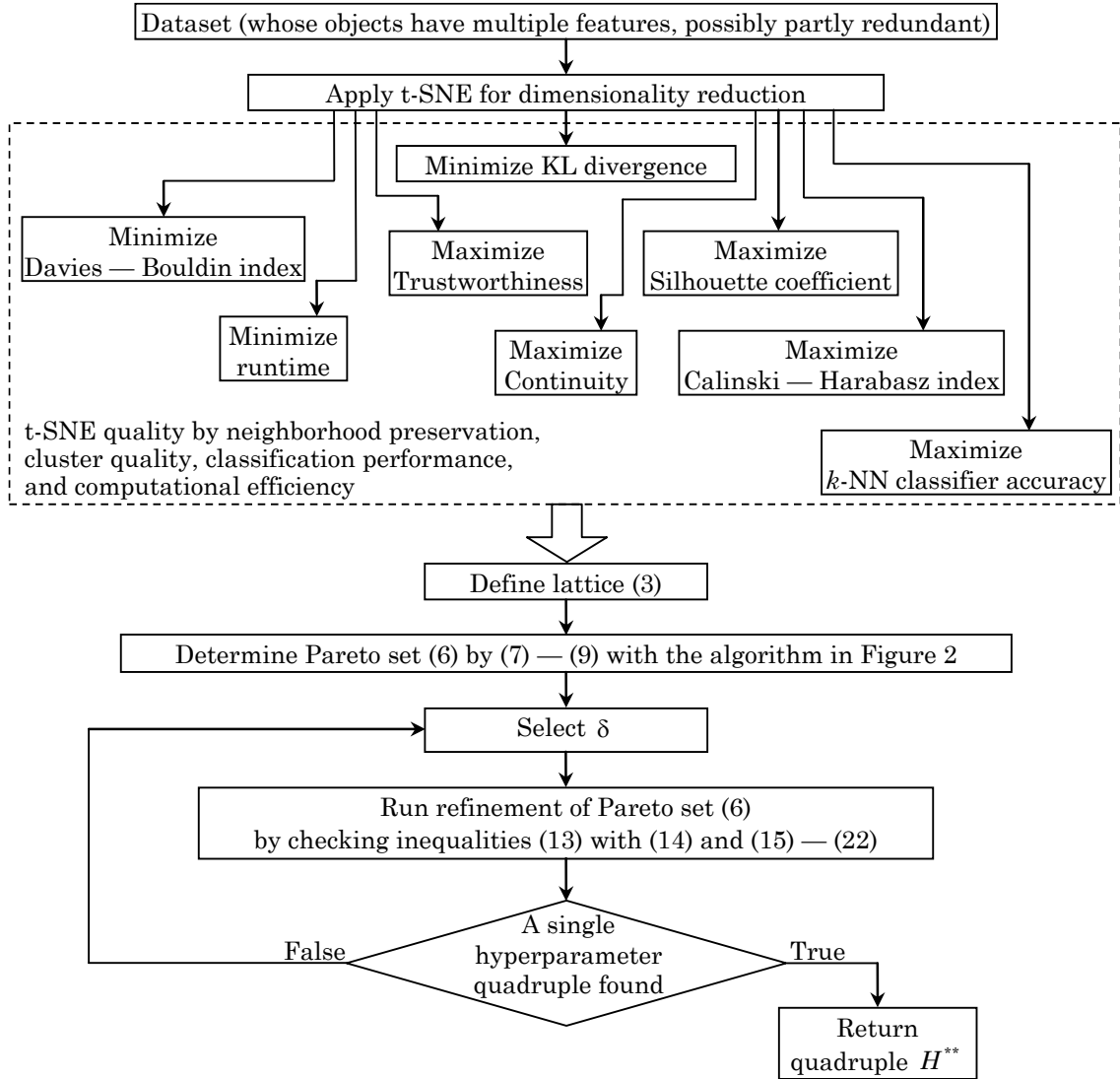


Figure 3: The eight-objective optimization workflow.

Singleton (23) is obtained slower (in a greater number of iterations) by a sufficiently small δ , but setting δ too high may return the empty set. Thus, Figure 4 shows four polylines at

$$\delta \in \{0.001, 0.0005, 0.0002, 0.0001\} \quad (25)$$

of how number M^* of the refined Pareto-optimal quadruples in set (12) changes through iterations of (15) – (22) for dataset “Iris flower” and t-SNE hyperparameters (10). The best quadruple (5) therein is

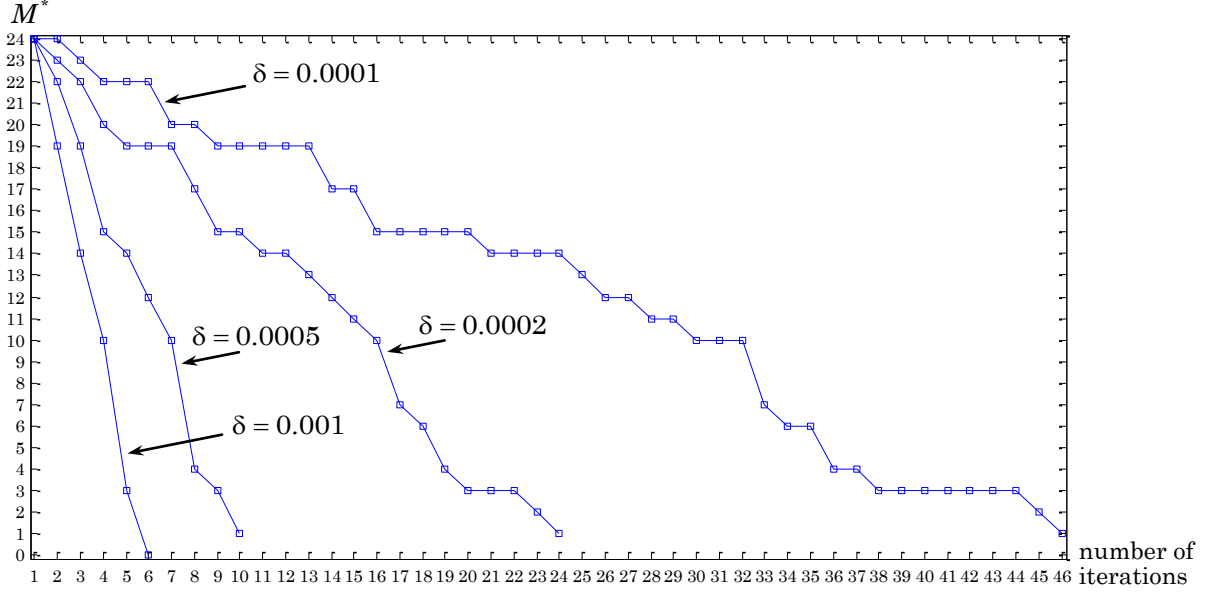


Figure 4: Four iterative processes of refining the Pareto-optimal quadruples at parameter (25), where it is seen that fewer iterations are typical after setting value δ higher.

$$H^{**} = \{\alpha^{**}, x^{**}, \lambda^{**}, \theta^{**}\} = \{35, 3, 200, 0.7\}. \quad (26)$$

The t-SNE algorithm with hyperparameters (26) applied to dataset “Iris flower” returns embeddings significantly differing in visual representation (Figure 5), but their quality criteria values (24) fluctuate much less:

$$d_{KL}^{**} \in (0.1; 0.15), u_{\text{trust}}^{**} \in (0.965; 0.975), u_{\text{cont}}^{**} \in (0.97; 0.977), b_{SC}^{**} \in (0.6; 0.65), v_{DB}^{**} \in (0.71; 0.8), \\ v_{CH}^{**} \in (450; 4000), a_{k-NN}^{**} \in (0.92; 0.98), t_{\text{run}}^{**} \in (0.7; 0.8).$$

Figure 5 shows that the Setosa cluster appears in different quadrants (corners) of the embedding plane. Repeated executions of the t-SNE algorithm with the same Pareto-optimal hyperparameter quadruple (26) produce noticeably different two-dimensional embeddings because the optimization of the KL divergence is a stochastic, non-convex process [27, 28]. Depending on the random initialization and the subsequent gradient descent trajectory, the t-SNE algorithm may converge to different local minima that preserve local neighborhood relationships almost equally well while differing in the global arrangement of the embedded points [20, 22].

This stochasticity explains the relatively wide variation of the Calinski – Harabasz index for dataset “Iris flower”. Since the index is defined as the ratio of the between-cluster dispersion to the within-cluster dispersion, even moderate changes in the compactness of individual clusters or in the distances between their centroids may result in substantial changes in its numerical value. In contrast, the remaining embedding quality criteria, particularly the Trustworthiness and Continuity measures u_{trust}^{**} and u_{cont}^{**} , are considerably more stable because they primarily evaluate the preservation of local neighborhood relationships rather than the global geometry of the embedding.

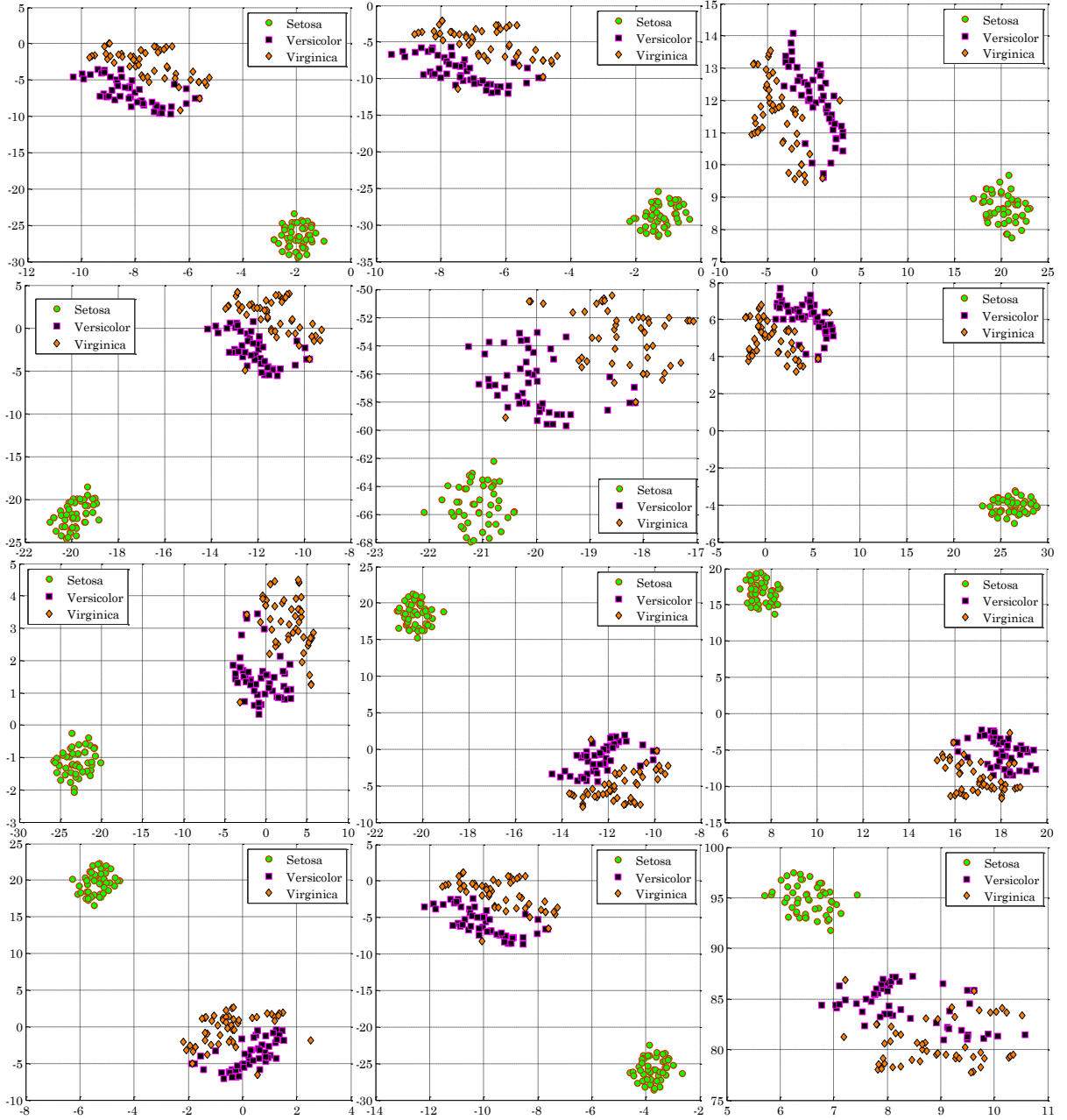


Figure 5: The t-SNE embeddings of dataset “Iris flower” obtained under hyperparameters (26), where PF is 35, EF is 3, LR is 200, and BHAP is 0.7 (only a small fraction of Iris virginica is mixed with Iris versicolor).

The visual appearance of the embeddings in Figure 5 also exhibits substantial variability along the axes. Although the Setosa class consistently forms a well-separated compact cluster, this cluster may appear in any of the four corners of the embedding plane, while the partially overlapping Versicolor and Virginica clusters occupy the diagonally opposite region. Such changes are expected because the absolute position, orientation, and reflection of a t-SNE embedding are not uniquely determined and have no intrinsic meaning (unless the t-SNE embedding is obtained at the fixed pseudorandom number generator state or seed [27]). Furthermore, the numerical ranges of both coordinate axes may vary considerably between different runs, indicating that the embedding is also subject to arbitrary translation and scaling. Consequently, only the relative neighborhood structure and cluster configuration should be interpreted, whereas the absolute coordinates, orientation, and axis ranges (of any dataset embeddings, not only those in Figure 5) should not be regarded as meaningful characteristics of the embedding. Meanwhile, averaging over a group of embeddings cannot be a way to obtain a single embedding with reliable and stable global geometry.

4.4. Examples on big data set instances

Bigger datasets are handled slower by the t-SNE. When the number of features is a few hundred and more, the PCA preprocessing is applied to reduce the number of features. For instance, for a dataset of a few million objects, each having 100 features, the PCA preprocessing is completed within minutes by sufficient RAM (no less than 32 GB or so), so PCA is not the bottleneck. The t-SNE algorithm must construct local neighborhoods (approximately, when using the Barnes – Hut t-SNE implementation). This is computationally intensive and typically dominates the PCA preprocessing stage. The Barnes – Hut approximation reduces the computational complexity from approximately $O(n^2)$ to $O(n \log n)$ [8], giving an enormous workload for computational machines from big data set instances (datasets of 10^5 to a few million objects having at least a few hundred features each). Therefore, the t-SNE algorithm takes several hours to produce a two-dimensional embedding of a dataset of 10^6 objects having 100 features each. Such datasets (and other big data sets) are synthetically generated for a given number of classes. Each class is generated independently, where for every class the informative features are sampled from a multivariate normal distribution whose mean vector differs from the mean vectors of the other classes. Consequently, the classes become distinguishable in the informative feature subspace. At the same time, the remaining features are generated as zero-mean Gaussian noise with a prescribed standard deviation. These noisy features do not carry discriminative information and imitate redundant or irrelevant variables that are commonly encountered in high-dimensional real-world datasets.

In this way, bigger datasets resembling the “Iris flower” dataset are generated, and the t-SNE algorithm is applied to them. For instance, a synthetic dataset of 600 objects, divided into 200 objects per class, where each object has 500 features and the number of informative features is 250, is handled by the t-SNE within 2 seconds (processor Intel Core i7, RAM is 64 GB). The best hyperparameter quadruple for the t-SNE dimensionality reduction here is

$$H^{**} = \{\alpha^{**}, x^{**}, \lambda^{**}, \theta^{**}\} = \{40, 3, 50, 0.6\}$$

being close to hyperparameter quadruple (26). It is also obtained by determining a Pareto-optimal set (6) of the eight-objective optimization problem (4) over lattice (3) by (10), and subsequently refining the Pareto-optimal set by checking inequalities (13) with (14) and (15) – (22) for $\delta = 0.0001$. Nine t-SNE embeddings of this synthetic dataset are shown in Figure 6, where an interesting property can be spotted outright. Every embedding has specific two points of the square-marked class: one point is always located very close to the circled-marked cluster, whereas the other point is always located in-between the circled-marked and the square-marked clusters. This demonstrates once again that, despite the same-hyperparameter embedding “fluctuates”, the t-SNE excellently preserves local neighborhood relationships from the original dataset.

Those local neighborhood relationships may become very complicated for bigger datasets consisting of more classes. For instance, another synthetic dataset with seven classes, with a million objects per class, where each object has 200 features (50 of which are informative), is generated by the same Gaussian cloud approach used above, but with adding “light” sinusoidal waves to the data. The best hyperparameter quadruple for the t-SNE dimensionality reduction of this dataset is

$$H^{**} = \{\alpha^{**}, x^{**}, \lambda^{**}, \theta^{**}\} = \{25, 11, 800, 0.3\}.$$

Using a very powerful workstation, it takes 5 to 8 hours to produce t-SNE embeddings of this dataset. Four embeddings are shown in Figure 7, where some local neighborhood relationships are highlighted as well. PF value $\alpha^{**} = 25$ suggests that the best embeddings are obtained on preserving the neighborhoods of roughly a few dozen nearest points rather than trying to preserve a larger-scale structure. Since the classes are modified Gaussian clouds, this is reasonable: each point has many close neighbors within its own class; preserving these local neighborhoods naturally produces compact clusters. Had the optimal perplexity been around 80 to 100, it would have implied that more global relationships between the classes were important. Instead, the Pareto optimization indicates that emphasizing local structure yields better overall performance. EF value $x^{**} = 11$ suggests that

the optimization benefits from pulling neighboring points together quite aggressively before the repulsive forces dominate. This usually leads to more compact clusters, clearer visual separation, improved values of cluster quality indices such as the Silhouette coefficient and the Davies — Bouldin index, which here are $b_{SC}^{**} \in (0.69; 0.78)$, $v_{DB}^{**} \in (0.56; 0.64)$. A relative high LR value ($\lambda^{**} = 800$) is generally expected as many t-SNE implementations actually recommend an LR that increases with the dataset size. BHAP value $\theta^{**} = 0.3$ is the easiest to interpret for such a large dataset. This relatively small BHAP means more accurate force computations and slower execution, but slightly better embeddings. So, for this dataset, the improvement in embedding quality outweighed the increase in runtime. If the optimization had favored, say, BHAP $\theta^{**} = 0.7$, it would have implied that computational efficiency was more important than the small loss in embedding quality.

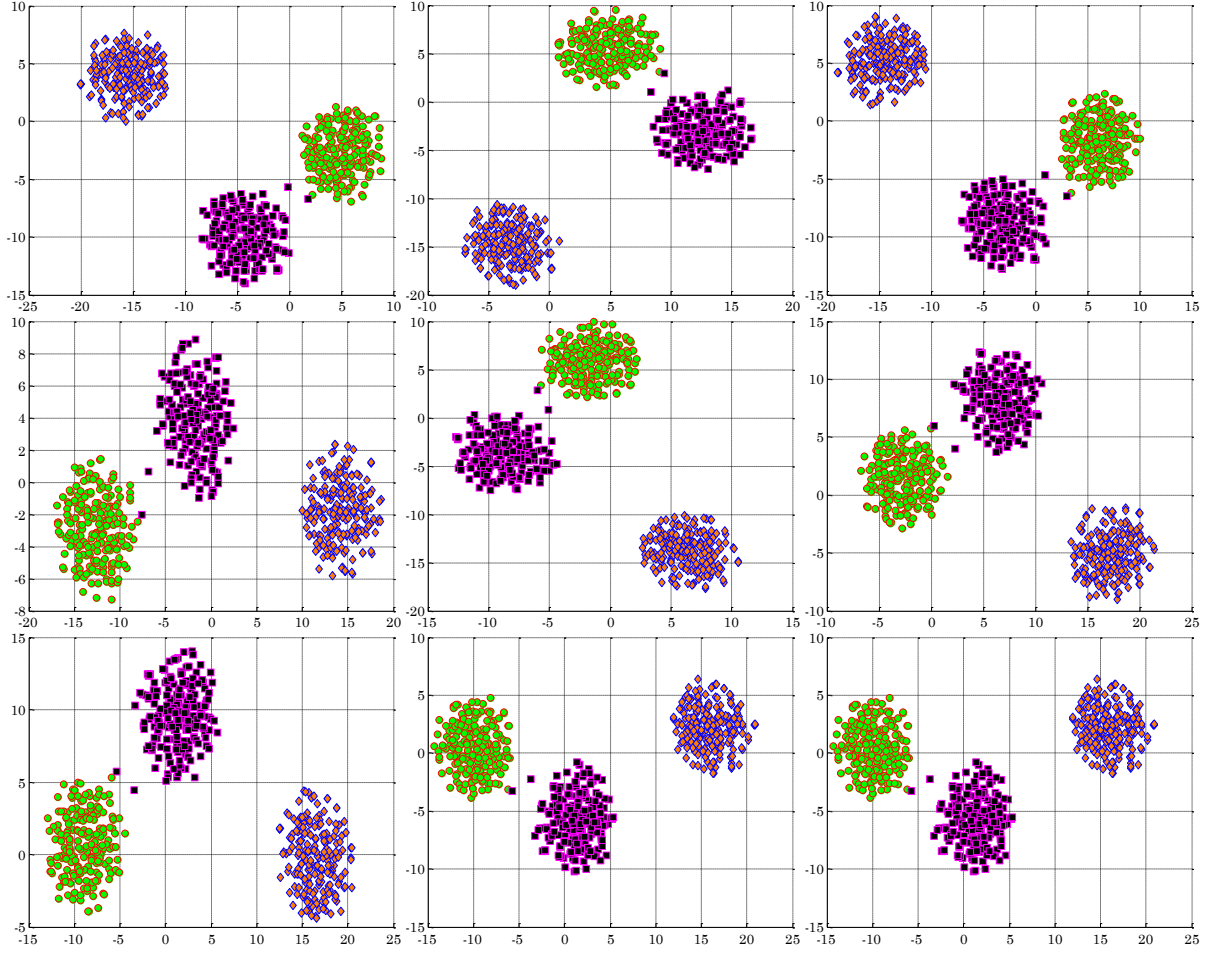


Figure 6: Nine t-SNE embeddings of a synthetic dataset (600 objects, three classes, 200 objects per class, each object has 500 features) obtained under the best Pareto-optimal hyperparameters, where PF is 40, EF is 3, LR is 50, and BHAP is 0.6.

Interval estimates of runtime and percentage of quality criteria values fluctuation, apart from the Calinski — Harabasz index and runtime, for typical big data set instances are given in Table 1. The intervals encompass 95 % of the 100 simulated instances per dataset size and dimensionality.

Table 1

Interval estimates of runtime and percentage of quality criteria fluctuation (without v_{CH}^{**} , t_{run}^{**})

Dataset and dimensionality reduction	1	2	3	4	5
Size (in millions of objects)	1	5	7	10	20
Number of features (dimensionality)	100	150	200	400	750
Runtime (in hours)	(0.4; 0.75)	(2.8; 4.6)	(4.8; 7.9)	(9.2; 15)	(23; 44)
Percentage of quality criteria fluctuation	(11; 17)	(10; 14.6)	(9.5; 13.5)	(9.1; 13.2)	(8.7; 12.8)

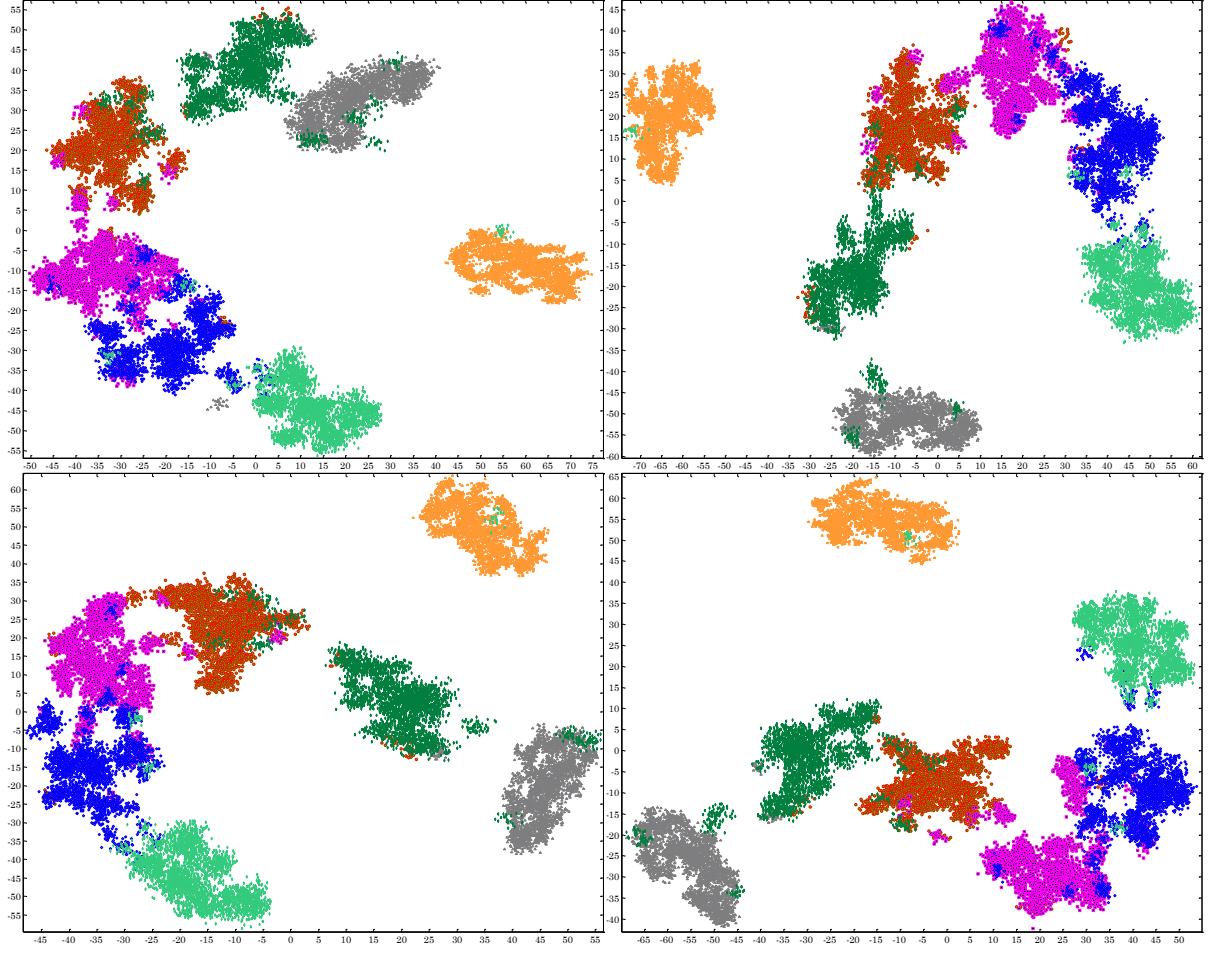


Figure 7: Four t-SNE embeddings of a synthetic big data set (7 million objects, seven classes, a million objects per class, each object has 200 features) obtained under the best Pareto-optimal hyperparameters, where PF is 25, EF is 11, LR is 800, and BHAP is 0.3.

5. Discussion

The obtained Pareto-optimal hyperparameter quadruples indicate that different quality criteria may favor different embedding characteristics, making the search for a single universally optimal hyperparameter vector really inappropriate, when the dataset subject to the t-SNE dimensionality reduction is labeled. Instead, Pareto optimization by (7) – (9) with the algorithm in Figure 2 identifies a set of mutually non-dominated hyperparameter quadruples that represent different compromises between embedding quality and computational cost. The subsequent refinement of the Pareto set by checking inequalities (13) with (14) and (15) – (22) into a single recommended hyperparameter quadruple enables the practical application of the proposed methodology, preserving the advantages of multi-objective optimization and not thinking about selecting among Pareto-optimal quadruples, which often constitute at least about 10 % or so of the total number of lattice (3) points. The only parameter to select is a sufficiently small δ for decrements (15) – (17) and increments (18) – (22).

Repeated executions of the t-SNE algorithm with identical hyperparameter values confirm that the absolute coordinates of embedded points have no intrinsic meaning and that only the relative neighborhood relationships and cluster configuration should be interpreted. The coordinate ranges of both embedding axes (for two-dimensional embeddings) may vary substantially between runs because the absolute position, orientation, reflection, and scaling of a t-SNE embedding are not uniquely determined. The stochastic nature of t-SNE also influences the values of the embedding quality criteria. While the KL divergence, Trustworthiness, Continuity, the Silhouette coefficient, the Davies – Bouldin index, the classification accuracy, and the runtime exhibit comparatively moderate variability between repeated runs (Table 1), the Calinski – Harabasz index varies over a much wider

range. This is explained by the mathematical definition of the index, which represents the ratio of the between-cluster dispersion to the within-cluster dispersion. Relatively small changes in cluster compactness or centroid locations caused by convergence to different local minima may therefore produce substantial changes in the Calinski – Harabasz value even when the overall visual appearance of the embedding remains similar.

The experimental results also confirm that the computational requirements of t-SNE increase rapidly with the dataset size. For datasets containing several hundred or several thousand observations, the runtime remains sufficiently small, and this enables exhaustive hyperparameter optimization using a grid search over four-dimensional lattice (3). However, for datasets containing hundreds of thousands or millions of observations, the computational cost of repeated t-SNE executions becomes the principal limitation rather than the optimization procedure itself, despite the Barnes – Hut approximation significantly accelerates the original algorithm [8, 10]. For genuinely big data applications involving millions of observations, scalable implementations such as Fit-SNE (Fast Interpolation-based t-SNE) [29, 30], openTSNE [31, 32], or GPU-accelerated variants like t-SNE-CUDA [33, 34] are more appropriate.

6. Conclusions

Although the KL divergence represents the optimization objective minimized internally by t-SNE, it does not explicitly account for cluster compactness, class separability, classification performance, or computational efficiency. Therefore, the proposed eight-objective optimization, which simultaneously considers the four label-independent criteria (KL divergence, Trustworthiness, Continuity, runtime) and the four label-based criteria (Silhouette coefficient, Davies – Bouldin index, Calinski – Harabasz index, k -NN classifier accuracy), provides a more comprehensive basis for t-SNE hyperparameter selection, regardless of whether the dataset is labeled or not. In this way, the eight-objective hyperparameter optimization of t-SNE dimensionality reduction uses cluster quality and neighborhood preservation criteria so that the t-SNE embedding efficiency is expanded from the single internal objective (KL divergence) to a hierarchy of complementary external quality measures. The optimization is realized in three successive stages (Figure 3): creation of four-dimensional lattice (3), formation of Pareto-optimal set (6), and its refinement to a single hyperparameter quadruple in Pareto singleton (23). The singleton is obtained in a few iterations to several tens of iterations with linear time complexity.

Practical approbation of the suggested methodology demonstrates that the eight-objective optimization provides a considerably richer evaluation of t-SNE embeddings than relying exclusively on the final KL divergence. Simultaneously accounting for neighborhood preservation, cluster quality, classification performance, and computational efficiency (especially for big data instances) enables the selection of hyperparameter quadruples that produce more balanced and practically useful embeddings, thereby improving the reliability and readability of dimensionality reduction for both visualization and subsequent machine learning tasks. Nevertheless, the Barnes – Hut approximation of t-SNE has computational complexity $O(n \log n)$ and therefore fits better shallow datasets like “Iris flower”; embeddings of big data sets are optimized within reasonable time only with implementations such as Fit-SNE whose approximate computational complexity is $O(n)$ [29, 30]. Then for a few million observations, the Fit-SNE implementation is expected to become 14 to 18 times faster than the Barnes – Hut approximation.

Further research can be directed toward investigating peculiarities of parallelization of determining Pareto-optimal set (6) for large datasets. Despite running through four-dimensional lattice (3) is easily parallelizable (theoretically), nodes (t-SNE hyperparameter quadruples) of the lattice are usually estimated in widely varying runtime, which may make parallelization inefficient for shallower datasets. Another open question is the lattice density, which strongly affects the optimization as sparser lattices may just omit some promising hyperparameter quadruples.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

- [1] L. van der Maaten, G. Hinton, Visualizing Data using t-SNE, *Journal of Machine Learning Research* 9 (2008) 2579–2605.
- [2] L. Rokach, O. Maimon, E. Shmueli, *Machine Learning for Data Science Handbook. Data Mining and Knowledge Discovery Handbook*, 3rd ed., Springer, 2023. doi:10.1007/978-3-031-24628-9.
- [3] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics*, MIT Press, 2023.
- [4] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, D. A. Keim, Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis, *IEEE Transactions on Visualization and Computer Graphics* 23(1) (2017) 241–250. doi:10.1109/TVCG.2016.2598495.
- [5] D. Kobak, P. Berens, The art of using t-SNE for single-cell transcriptomics, *Nature Communications* 10 (2019) 5416. doi:10.1038/s41467-019-13056-x.
- [6] J. Balderas, L. Wang, A. Korzeniowski, R.-C. Li, Transductive t-SNE (tt-SNE) for classification and data visualization, *Pattern Recognition Letters* 206 (2026) 42–48. doi:10.1016/j.patrec.2026.05.007.
- [7] F. Anowar, S. Sadaoui, B. Selim, Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE), *Computer Science Review* 40 (2021) 100378. doi:10.1016/j.cosrev.2021.100378.
- [8] L. van der Maaten, Accelerating t-SNE using Tree-Based Algorithms, *Journal of Machine Learning Research* 15(93) (2014) 3221–3245.
- [9] J. A. Lee, E. Couplet, P. H. Lambert, P. Merveille, L. Journaux, D. Mulders, C. de Bodt, M. Verleysen, Improving on early exaggeration in t-SNE: Early hierarchization better preserves global structure, *Neurocomputing* 660 (2026) 131882. doi:10.1016/j.neucom.2025.131882.
- [10] A. C. Belkina, C. O. Ciccolella, R. Anno, R. Halpert, J. Spidlen, J. E. Snyder-Cappione, Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets, *Nature Communications* 10 (2019) 5415. doi:10.1038/s41467-019-13055-y.
- [11] A. A. Wani, Comprehensive review of dimensionality reduction algorithms: challenges, limitations, and innovative solutions, *PeerJ Computer Science* 11 (2025) e3025. doi:10.7717/peerj-cs.3025.
- [12] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics* 20 (1987) 53–65. doi:10.1016/0377-0427(87)90125-7.
- [13] D. L. Davies, D. W. Bouldin, A Cluster Separation Measure, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1(2) (1979) 224–227. doi:10.1109/TPAMI.1979.4766909.
- [14] J. Venna, S. Kaski, Neighborhood Preservation in Nonlinear Projection Methods: An Experimental Study, in: G. Dorffner, H. Bischof, K. Hornik (Eds.), *Artificial Neural Networks — ICANN 2001*, volume 2130 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2001, pp. 485–491. doi:10.1007/3-540-44668-0_68.
- [15] J. Venna, J. Peltonen, K. Nybo, H. Aidos, S. Kaski, Information retrieval perspective to nonlinear dimensionality reduction for data visualization, *Journal of Machine Learning Research* 11(13) (2010) 451–490.
- [16] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*, 2nd ed., Springer New York, 2009. doi:10.1007/978-0-387-84858-7.
- [17] V. Romanuke, S. Merinova, H. Yehoshyna, Optimized centroid-based clustering of dense nearly-square point clouds by the hexagonal pattern, *Electrical, Control and Communication Engineering* 19(1) (2023) 29–39. doi:10.2478/ecce-2023-0005.

- [18] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, *Communications in Statistics* 3(1) (1974) 1–27. doi:10.1080/03610927408827101.
- [19] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, J. R. Foulds, Chapter 12 – Beyond supervised and unsupervised learning, in: *Data Mining: Practical Machine Learning Tools and Techniques* (Fifth Edition), Morgan Kaufmann, 2026, pp. 473–484. doi:10.1016/B978-0-443-15888-9.00013-3.
- [20] L. Xia, C. Lee, J. J. Li, Statistical method scDEED for detecting dubious 2D single-cell embeddings and optimizing t-SNE and UMAP hyperparameters, *Nature Communications* 15 (2024) 1753. doi:10.1038/s41467-024-45891-y.
- [21] M. Allaoui, S. B. Belhaouari, R. Hedjam, K. Bouanane, M. L. Kherfi, t-SNE-PSO: Optimizing t-SNE using particle swarm optimization, *Expert Systems with Applications* 269 (2025) 126398. doi:10.1016/j.eswa.2025.126398.
- [22] Z. Yang, Y. Chen, J. Corander, T-SNE Is Not Optimized to Reveal Clusters in Data, arXiv:2110.02573 [cs.LG] (2021). doi:10.48550/arXiv.2110.02573.
- [23] V. Marx, Seeing data as t-SNE and UMAP do, *Nature Methods* 21(6) (2024) 930–933. doi:10.1038/s41592-024-02301-x.
- [24] Y.-T. Liao, H. Luo, A. Ma, Efficient and Robust Bayesian Selection of Hyperparameters in Dimension Reduction for Visualization, arXiv:2306.00357 [stat.ML] (2023). doi:10.48550/arXiv.2306.00357.
- [25] S. Bechikh, R. Datta, A. Gupta, *Recent Advances in Evolutionary Multi-objective Optimization*, Springer Cham, 2017. doi:10.1007/978-3-319-42978-6.
- [26] A. Jahan, K. L. Edwards, M. Bahraminasab, *Multi-criteria Decision Analysis for Supporting the Selection of Engineering Materials in Product Design*, 2nd ed, Butterworth-Heinemann, 2016. doi:10.1016/C2014-0-03347-3.
- [27] D. Kobak, G. C. Linderman, Initialization is critical for preserving global data structure in both t-SNE and UMAP, *Nature Biotechnology* 39 (2021) 156–157. doi:10.1038/s41587-020-00809-z.
- [28] S. Arora, W. Hu, P. K. Kothari, An Analysis of the t-SNE Algorithm for Data Visualization, *Proceedings of the 31st Conference On Learning Theory*, in: *Proceedings of Machine Learning Research*, 75 (2018) 1455–1462.
- [29] G. C. Linderman, M. Rachh, J. G. Hoskins, S. Steinerberger, Y. Kluger, Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data, *Nature Methods* 16 (2019) 243–245. doi:10.1038/s41592-018-0308-4.
- [30] G. C. Linderman, S. Steinerberger, Clustering with t-SNE, provably, *SIAM Journal on Mathematics of Data Science* 1(2) (2019) 313–332. doi:10.1137/18M1216134.
- [31] P. G. Poličar, M. Stražar, B. Zupan, openTSNE: A modular python library for t-SNE dimensionality reduction and embedding, *Journal of Statistical Software* 109(3) (2024) 1–30. doi:10.18637/jss.v109.i03.
- [32] P. G. Poličar, M. Stražar, B. Zupan, Embedding to reference t-SNE space addresses batch effects in single-cell classification, *Machine Learning* 112 (2023) 721–740. doi:10.1007/s10994-021-06043-1.
- [33] D. M. Chan, R. Rao, F. Huang, J. F. Canny, T-SNE-CUDA: GPU-Accelerated T-SNE and its Applications to Modern Data, in: *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, Lyon, France, 2018, pp. 330–338. doi:10.1109/CAHPC.2018.8645912.
- [34] D. M. Chan, R. Rao, F. Huang, J. F. Canny, GPU accelerated t-distributed stochastic neighbor embedding, *Journal of Parallel and Distributed Computing* 131 (2019) 1–13. doi:10.1016/j.jpdc.2019.04.008.