# L'extension hors échantillon de l'algorithme t-SNE

Paul HONEINE[1]

[1]Univ Rouen Normandie, INSA Rouen Normandie, Université Le Havre Normandie,
Normandie Univ, LITIS UR 4108, F-76000 Rouen, France

**Résumé –** L'algorithme *t-distributed stochastic neighbor embedding* (t-SNE) est une technique de réduction de dimension communément explorée en science des données et apprentissage statistique. Il vise à représenter un ensemble d'échantillons dans un espace de plus faible dimension, en minimisant la divergence de Kullback-Leibler entre les distributions dans les deux espaces. Le présent article s'attaque à la résolution du problème de prolongement hors échantillon d'entraînement, connu en anglais par *out-of-sample extension* (OoS). Nous démontrons que t-SNE s'apprête naturellement à une extension OoS. De plus, nous proposons une résolution par un algorithme itératif basé sur le point fixe, tout en mettant en évidence plusieurs propriétés intéressantes. Les résultats expérimentaux sur des chiffres manuscrits démontrent la pertinence de l'approche proposée.

**Abstract –** The t-distributed stochastic neighbor embedding (t-SNE) is central in data science for data visualization and dimensionality reduction. It seeks to represent a set of samples in a lower-dimensional space according to a Student distribution, by minimizing the Kullback-Leibler divergence of the distributions in both spaces. This paper introduces an out-of-sample extension of t-SNE, by extending the embedding to samples not considered in the training phase. While this problem was seldom tackled in the literature, we show that the internal mechanisms of t-SNE allow a straightforward out-of-sample extension. We provide an efficient resolution by a fixed-point iteration scheme with interesting properties. Experiments on the MNIST dataset demonstrate the relevance of this approach.

## 1 Introduction

L'algorithme *t-distributed stochastic neighbor embedding* (t-SNE) est une technique de réduction de dimension très connue pour la visualisation en science des données. Introduit en 2008 par Geoffrey Hinton et Laurens van der Maaten dans [16], il est sans doute l'un des plus populaires grâce à ses performances exceptionnelles dans un large éventail d'applications [3]. Malgré son grand intérêt, t-SNE et ses différentes variantes reposent exclusivement sur la représentation d'un jeu de données prédéfini, dit échantillon d'entraînement. Toutefois, les algorithmes de t-SNE ne permettent pas de représenter de nouveaux échantillons qui ne font pas partie de l'ensemble d'échantillons d'entraînement.

Le problème de prolongement hors échantillon d'entraînement est connu en anglais par *out-of-sample extension* (OoS). L'analyse en composantes principales (ACP) et ses différentes variantes sont dotées d'un mécanisme simple pour une extension OoS, par projections linéaires ou non linéaires [5]. Toutefois, l'ACP semble être l'exception dans le paysage de la littérature. Les méthodes de réduction de dimension, notamment *local linear embedding*, *isomap* et *Laplacian eigenmap*, reposent sur une décomposition en vecteurs-valeurs propres d'une matrice de similarité/voisinage, ce qui permet de proposer des extensions OoS par approximation de Nyström [1]. Les résolutions proposées pour l'extension OoS sont souvent génériques et indépendantes des algorithmes de réduction de dimension, comme par exemple avec un réseau de neurones profond [15]. Voir [14] pour une revue récente.

Le présent article s'attaque au problème du prolongement hors échantillon d'entraînement de l'algorithme t-SNE. À notre connaissance, il n'y a pas d'extension OoS de t-SNE, mais seulement l'utilisation de techniques sur étagère ou la modification de t-SNE (*e.g.* projection à noyau) [4]. Nous démontrons qu'il est possible de profiter pleinement des mécanismes internes de t-SNE pour proposer un algorithme naturel pour le prolongement OoS. Après avoir présenté une résolution par descente de gradient, nous introduisons un algorithme itératif basé sur le point fixe. De plus, nous mettons en évidence plusieurs propriétés intéressantes pour une telle résolution, dont l'adaptation du pas à chaque itération et l'optimisation convexe. Les résultats expérimentaux sur le jeu de données MNIST d'images de chiffres manuscrits démontrent la pertinence de la méthode proposée.

## 2 t-SNE

Soient $x_1, x_2, \ldots, x_n$ les échantillons dans un espace de grande dimension $\mathbb{X} \subset \mathbb{R}^D$. L'algorithme t-SNE les projette en $y_1, y_2, \ldots, y_n$ dans un espace de faible dimension $\mathbb{Y} \subset \mathbb{R}^d$, en préservant les affinités entre the échantillons dans les deux espaces. Cette affinité est mesurée entre 2 échantillons à partir de leur probabilité d'être voisins. Dans l'espace d'origine, la probabilité jointe est considérée

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}, \tag{1}$$

où $p_{i|j}$ désigne la probabilité que $x_i$ soit voisin de $x_j$ selon une distribution gaussienne

$$p_{i|j} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_j^2)}{\sum_{k\neq i} \exp(-\|x_i - x_k\|^2/2\sigma_j^2)}. \tag{2}$$

Dans l'espace de représentation $\mathbb{Y}$, la similarité est mesurée selon la loi de Student avec un degré de liberté, afin de surmonter la problématique de l'encombrement et compenser l'écart

entre les dimensions, c'est à dire

$$q_{ij} = \frac{(1 + \| y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \| y_k - y_l\|^2)^{-1}},$$

avec $q_{ii} = q_{jj} = 0$. L'algorithme t-SNE minimise la divergence de Kullback-Leibler (KL) entre les deux distributions,

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{3}$$

L'algorithme opère une descente de gradient conventionnelle

$$y_i^{t+1} = y_i^t - \eta^t \nabla_{y_i^t} KL(y_i^t), \tag{4}$$

où $\eta^t$ est le pas et le gradient est évalué à $y_i$ selon

$$\nabla_{y_i} KL(y_i) = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \| y_i - y_j\|^2)^{-1}. \tag{5}$$

Plusieurs techniques pour accroître la convergence sont préconisées dans [11, 16], notamment l'exagération précoce (multiplier les probabilités d'origine par $\alpha > 1$).

# 3 Prolongement de t-SNE

À la lumière des dérivations de t-SNE, nous proposons dans cette section une extension OoS avec un schéma d'optimisation par descente de gradient. Soit $x_0 \in \mathbb{X}$ un nouvel échantillon à intégrer dans la représentation obtenue par t-SNE à partir des échantillons $x_1, x_2, \ldots, x_n$. Notre objectif est de déterminer $y_0$, sa représentation dans l'espace $\mathbb{Y}$. Soit $p_{0i}$ la mesure d'affinité entre $x_0$ et $x_i$, comme défini par la probabilité jointe (1). La mesure d'affinité $q_{0i}$ entre $y_0$ et $y_i$ dans $\mathbb{Y}$ est donnée par

$$q_{0i} = \frac{(1 + \| y_0 - y_i\|^2)^{-1}}{\sum_{k \neq l}(1 + \| y_k - y_l\|^2)^{-1}}.$$

En posant $d_{ij} = \| y_i - y_j\|$, nous obtenons

$$q_{0i} = \frac{(1 + d_{0i}^2)^{-1}}{Z},$$

où le dénominateur est estimé sur l'ensemble d'entraînement, avec

$$Z = \sum_{\substack{k,l=1 \\ k \neq l}}^{n} (1 + \| y_k - y_l\|^2)^{-1}.$$

Pour représenter l'échantillon $x_0$ par $y_0 \in \mathbb{Y}$, nous minimisons la fonction objectif

$$\Xi(y_0) = \sum_{i=1}^{n} p_{0i} \log \frac{p_{0i}}{q_{0i}}, \tag{6}$$

ce qui peut être vu comme la minimisation d'une divergence de KL augmentée en incluant $y_0$

$$\begin{aligned} KL_{\text{augm}} &= \sum_{\substack{i,j=0 \\ i \neq j}}^{n} p_{ij} \log \frac{p_{ij}}{q_{ij}} \\ &= \sum_{\substack{i,j=1 \\ i \neq j}}^{n} p_{ij} \log \frac{p_{ij}}{q_{ij}} + 2 \sum_{i=1}^{n} p_{0i} \log \frac{p_{0i}}{q_{0i}}. \end{aligned}$$

Puisque

$$\Xi(y_0) = \sum_{i=1}^{n} p_{0i} \log p_{0i} - \sum_{i=1}^{n} p_{0i} \log q_{0i}, \tag{7}$$

le premier terme ne dépend pas de $y_0$. De plus, les modifications de $y_0$ affectent seulement les distances $d_{i0}$ et $d_{0i}$ pour tout $i = 1, 2, \ldots, n$. Par conséquent, le gradient de $\Xi(y_0)$ est donné par la règle de dérivation en chaîne

$$\begin{aligned} \nabla_{y_0} \Xi(y_0) &= \sum_{j=1}^{n} \Big( \frac{\partial \Xi}{\partial d_{j0}} + \frac{\partial \Xi}{\partial d_{0j}} \Big)(y_0 - y_j) \\ &= 2 \sum_{j=1}^{n} \frac{\partial \Xi}{\partial d_{0j}} (y_0 - y_j). \end{aligned} \tag{8}$$

La dérivée $\frac{\partial \Xi}{\partial d_{0j}}$ est calculée de (7) selon

$$\begin{aligned} \frac{\partial \Xi}{\partial d_{0j}} &= -\sum_{i=1}^{n} p_{0i} \frac{\partial \log q_{0i}}{\partial d_{0j}} \\ &= -\sum_{i=1}^{n} p_{0i} \frac{\partial \log \frac{(1+d_{0i}^2)^{-1}}{Z}}{\partial d_{0j}} \\ &= -\sum_{i=1}^{n} p_{0i} \frac{-1}{(1 + d_{0i}^2)} \frac{\partial (1 + d_{0i}^2)}{\partial d_{0j}} + \sum_{i=1}^{n} p_{0i} \frac{1}{Z} \frac{\partial Z}{\partial d_{0j}}. \end{aligned}$$

Pour le premier terme du membre droit, $\frac{\partial (1+d_{0i}^2)}{\partial d_{0j}}$ est non nul si et seulement si $i = j$, alors que le second terme est nul car $Z$ est indépendant de $d_{0j}$. Donc,

$$\frac{\partial \Xi}{\partial d_{0j}} = -p_{0j} \frac{-1}{(1 + d_{0j}^2)} \frac{\partial (1 + d_{0j}^2)}{\partial d_{0j}} = 2 p_{0j} (1 + d_{0j}^2)^{-1}.$$

Enfin, en substituant cette expression dans (8), nous obtenons

$$\nabla_{y_0} \Xi = 4 \sum_{j=1}^{n} w_{0j} p_{0j} (y_0 - y_j), \tag{9}$$

où $w_{0j} = (1 + d_{0j}^2)^{-1}$. Nous pouvons donc utiliser cette expression avec un schéma de descente de gradient de la forme

$$y_0^{t+1} = y_0^t - \eta^t \nabla_{y_0^t} \Xi(y_0^t), \tag{10}$$

pour un pas $\eta^t$ à optimiser à chaque itération $t$.

# 4 Algorithme itératif de point fixe

Nous proposons dans cette section un schéma d'optimisation itérative de point fixe, c'est à dire une règle de mise à jour de la forme $y^{t+1} = f(y^t)$ qui vise à estimer $y_0$. Pour cela, il suffit d'avoir une expression de la forme $y = f(y)$ à l'optimum. La fonction $f(\cdot)$ n'étant pas unique, différents algorithmes itératifs de point fixe peuvent être dérivés. Nous présentons dans la suite une stratégie simple, en annulant le gradient $\nabla_{y_0} \Xi$ dans (9), ce qui permet d'écrire :

$$y_0 \sum_{j=1}^{n} w_{0j} p_{0j} = \sum_{j=1}^{n} w_{0j} p_{0j}\, y_j.$$

De cette expression, nous proposons l'itération de point fixe

$$y_0^{t+1} = \frac{\sum_j w_{0j}^t p_{0j}\, y_j}{\sum_j w_{0j}^t p_{0j}}. \tag{11}$$

Malgré sa simplicité, cette résolution itérative possède plusieurs propriétés intéressantes.

## 4.1 Propriétés sur le schéma d'optimisation

Comparée à une descente de gradient qui nécessite un réglage du pas $\eta^t$ dans (10), l'itération (11) correspond à une descente de gradient avec un pas ajusté à chaque itération selon $\eta_*^t = 1/\big(4\sum_j w_{0j}^t p_{0j}\big)$, qui est positif car $w_{0j}^t, p_{0j} > 0$).

L'itération (11) est de la forme

$$y_0^{t+1} = \sum_{j=1}^{n} \beta_j^t \, y_j,$$

où $\beta_j^t = w_{0j}^t p_{0j} / \sum_j w_{0j}^t p_{0j}$. Puisque $\sum_{j=1}^n \beta_j^t = 1$ et $\beta_j^t > 0$, chaque nouvelle estimation $y_0^{t+1}$ se trouve à l'intérieur de l'enveloppe convexe de l'ensemble $\{y_1, y_2, \ldots, y_n\}$.

Alors que t-SNE nécessite une attention particulière pour la convergence, en particulier avec l'exagération et une bonne initialisation, ceci n'est pas le cas de l'algorithme itératif de point fixe. L'exagération n'a aucun effet sur la règle de mise à jour, dû à la normalisation. De plus, aucune initialisation particulière est nécessaire (nous avons utilisé l'initialisation $y_0^0 = \mathbf{0}$ dans les expérimentations). Ceci n'est pas le cas de t-SNE et d'autres méthodes connexes comme UMAP, où les résultats de l'optimisation dépendent fortement de l'initialisation, comme corroboré dans [8, 9, 17].

## 4.2 Complexité calculatoire et multiples OoS

La règle (11) peut s'écrire sous la forme matricielle

$$y_0^{t+1} = \boldsymbol{Y}\left(\tfrac{1}{\boldsymbol{p}_0^\top \boldsymbol{w}_0^t} \boldsymbol{p}_0 \odot \boldsymbol{w}_0^t\right), \tag{12}$$

où $\boldsymbol{Y}$ désigne la matrice de taille $d \times n$ des coordonnées estimées $y_1, y_2, \ldots, y_n$ par t-SNE ; $\boldsymbol{p}_0$ et $\boldsymbol{w}_0^t$ sont les vecteurs colonnes comprenant $p_{01}, p_{02}, \ldots, p_{0n}$ et $w_{01}^t, w_{02}^t, \ldots, w_{0n}^t$, respectivement. En termes de complexité calculatoire pour un seul échantillon OoS, (12) nécessite $\mathcal{O}(2n)$ multiplications et $\mathcal{O}(n)$ divisions pour calculer $\boldsymbol{w}_0^t$ à chaque itération, alors que $\boldsymbol{p}_0$ est calculée une seule fois et reste inchangée à chaque itération. La multiplication matricielle avec $\boldsymbol{Y}$ est en $\mathcal{O}(2n)$.

## 4.3 Estimation de multiples échantillons OoS

On peut étendre facilement cette règle pour adresser plusieurs échantillons, comme suit. Pour $m$ échantillons OoS, nous écrivons (12) selon l'expression matricielle

$$\boldsymbol{Y}_0^{t+1} = \boldsymbol{Y}\left(\big(\boldsymbol{P}_0 \odot \boldsymbol{W}_0^t\big)\,\mathrm{diag}\left(\boldsymbol{P}_0^\top \boldsymbol{W}_0^t\right)^{-1}\right), \tag{13}$$

où $\boldsymbol{Y}_0^{t+1}$ désigne la matrice des $m$ estimations à l'itération $t+1$, $\boldsymbol{P}_0$ et $\boldsymbol{W}_0^t$ sont les matrices formées par les vecteurs colonnes $\boldsymbol{p}_0$ et $\boldsymbol{w}_0^t$ pour tous les $m$ échantillons. L'opérateur $\mathrm{diag}(\cdot)$ produit une matrice diagonale à partir de la diagonale de sa matrice d'entrée.

## 4.4 Connexion à l'algorithme mean shift

L'algorithme itératif de point fixe (11) peut être vue comme une approximation de l'algorithme de clustering *mean shift*. Pour rappel, ce dernier vise à rechercher les modes d'une densité estimée à partir de la méthode de Parzen-Rosenblatt. Cette dernière consiste à estimer la densité d'un ensemble d'échantillons $\{y_1, y_2, \ldots, y_n\}$ selon

$$\widehat{p}(y) = \sum_{j=1}^{n} k(\|y - y_j\|^2), \tag{14}$$

pour un profil de noyau $k(\cdot)$. En annulant le gradient de l'estimation de la densité, c'est à dire $\nabla_y \widehat{p}(y) = \sum_{j=1}^n \nabla_y k(\|y - y_j\|^2)$, on obtient l'algorithme de *mean shift*

$$y^{t+1} = \frac{\sum_j k'\left(\|y^t - y_j\|^2\right) y_j}{\sum_j k'\left(\|y^t - y_j\|^2\right)}, \tag{15}$$

où $k'(\cdot)$ est la dérivée du profil du noyau $k(\cdot)$.

L'itération de point fixe (11) peut être vue comme une variante du *mean shift*, où les poids $w_{0j}$ définis par la $w(u) = 1/(1+u)$ pour $u = \|y_0^t - y_j\|^2$ jouent le même rôle que $k'(\cdot)$ dans (15). Cette analogie permet de donner une perspective de clustering à la méthode proposée, ce qui corrobore les résultats théoriques récents sur t-SNE comme méthode de clustering [2, 12].

## 4.5 Connexion à la résolution de la pré-image

Nous pouvons également dresser une analogie avec la résolution du problème de pré-image par un algorithme de point fixe. Le problème de pré-image en apprentissage statistique peut être résumé comme suit. Soit $\phi(\cdot)$ un plongement non linéaire d'un espace d'observation $\mathbb{Y}$ à un espace latent $\mathbb{X}$. Les méthodes d'apprentissage reposent sur un modèle linéaire dans ce dernier espace, de la forme

$$\psi = \sum_{j=1}^{n} \alpha_j \, \phi(y_j),$$

pour des paramètres $\alpha_1, \alpha_2, \ldots \alpha_n$ à optimiser selon un critère donné (*e.g.* maximisation de la marge pour les séparateurs à vaste marge). La pré-image $y \in \mathbb{Y}$ de $\psi$ peut être obtenue par minimisation de la norme du résiduel entre $\psi$ et $\phi(y)$ dans l'espace latent $\mathbb{X}$. En utilisant un noyau radial pour induire $\phi(\cdot)$, la fonction objectif devient

$$\Gamma(y) = -\sum_{j=1}^{n} \alpha_j \, k(\|y - y_j\|^2).$$

En annulant son gradient, nous obtenons la règle itérative de point fixe

$$y^{t+1} = \frac{\sum_{j=1}^n \alpha_j k'(\|y^t - y_j\|^2)\, y_j}{\sum_{j=1}^n \alpha_j k'(\|y^t - y_j\|^2)}, \tag{16}$$

où $k'(\cdot)$ est la dérivée du profil $k(\cdot)$, qui est de la forme $k'(u) = -ck(u)$ pour le noyau gaussien, où $c$ est une constante de normalisation. Pour plus de détails, voir [7] pour la définition du problème de pré-image et sa résolution.

Même si les perspectives des espaces d'observation et de représentation (latent) sont inversées entre t-SNE et méthodes d'apprentissage à noyaux, nous pouvons mettre en évidence plusieurs analogies à l'instar de ce qui a été présenté pour le *mean shift* (pas détaillé ici par manque de place). Ceci permet de profiter des récentes avancées théoriques sur le problème de la pré-image et sa résolution, comme étudié en profondeur pour le point fixe dans [6].

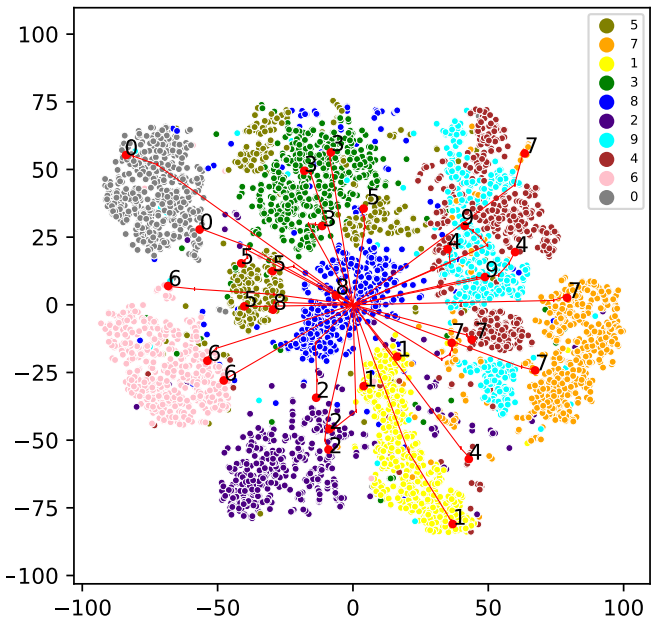


FIGURE 1 : La représentation de t-SNE pour 6000 images de MNIST, illustrées par les points colorés selon les classes des 10 chiffres. Le prolongement à 30 échantillons OoS est donné par les points rouges (•, avec leurs étiquettes) après seulement 5 itérations, et le chemin de solution est illustré par les lignes rouges (—).

# 5 Expérimentations

Pour illustrer la méthode itérative de point fixe proposée pour l'extension hors échantillon de l'algorithme t-SNE, nous avons considéré le jeu de données MNIST d'images manuscrites des 10 chiffres (de "0" à "9") [10]. De taille $28 \times 28$ pixels, chaque image a été représentée sous la forme d'un vecteur de taille 784. Pour être comparable aux résultats de la littérature, nous avons utilisé la même configuration que celle proposée dans [16]. En particulier, l'ensemble d'entraînement comprend $n = 6000$ images et l'algorithme t-SNE repose sur l'implémentation de `scikit-learn` avec ses paramètres par défaut [13], à savoir une perplexité de 30 pour calculer les largeurs de bande $\sigma_j$, une exagération précoce de $\alpha = 4$ et un taux d'apprentissage de $\eta = 1000$.

Nous avons choisi aléatoirement 30 échantillons OoS, c'est à dire qui ne font pas partie des $n = 6000$ échantillons d'entraînement de t-SNE. Pour l'initialisation de chaque estimation, nous avons utilisé $y_0^0 = \mathbf{0}$. La Figure 1 illustre les résultats de t-SNE des $n$ échantillons. L'évolution des estimations des 30 échantillons OoS est représentée en lignes rouges, où l'estimation finale est donnée après seulement 5 itérations, illustrée par des points rouges et les étiquettes correspondantes. Nous pouvons voir que la méthode proposée permet de bien représenter les échantillons OoS dans les bons clusters. Non présentés dans cet article par manque de place, les résultats obtenus par un schéma de descente de gradient montrent qu'une telle résolution souffre d'une faible convergence avec la nécessité de choisir un pas adapté.

# 6 Conclusion

Cet article a montré que nous pouvons profiter des mécanismes internes de t-SNE pour proposer une extension OoS. De plus, nous avons dérivé une résolution itérative par point fixe avec plusieurs propriétés intéressantes. Pas introduits dans cet article par manque de place, d'autres résultats théoriques sont présentés au colloque du GRETSI 2025, notamment démontant que l'algorithme itératif de point fixe est un algorithme MM, ainsi que des résultats expérimentaux plus approfondis.

# Références


[1] Yoshua BENGIO, Jean-françcois PAIEMENT, Pascal VINCENT, Olivier DELALLEAU, Nicolas ROUX et Marie OUIMET : Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems*, 16, 2003.

[2] T. Tony CAI et Rong MA : Theoretical foundations of t-SNE for visualizing high-dimensional clustered data. *Journal of Machine Learning Research*, 23(301):1–54, 2022.

[3] Benyamin GHOJOGH, Mark CROWLEY, Fakhri KARRAY et Ali GHODSI : *Stochastic Neighbour Embedding*, pages 455–477. Springer International Publishing, Cham, 2023.

[4] Andrej GISBRECHT, Alexander SCHULZ et Barbara HAMMER : Parametric nonlinear dimensionality reduction using kernel t-SNE. *Neurocomputing*, 147:71–82, 2015.

[5] Paul HONEINE : Online kernel principal component analysis : a reduced-order model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1814 – 1826, septembre 2012.

[6] Paul HONEINE : Theoretical insights on the pre-image resolution in machine learning. *Pattern Recognition*, 156:110800, décembre 2024.

[7] Paul HONEINE et Cédric RICHARD : Preimage problem in kernel-based machine learning. *IEEE Signal Processing Magazine*, 28(2):77 – 88, mars 2011.

[8] Dmitry KOBAK et George C. LINDERMAN : UMAP does not preserve global structure any better than t-SNE when using the same initialization. *bioRxiv*, 2019.

[9] Dmitry KOBAK et George C. LINDERMAN : Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature biotechnology*, 39(2):156–157, 2021.

[10] Yann LECUN et Corinna CORTES : MNIST handwritten digit database. 2010.

[11] George C. LINDERMAN, Manas RACHH, Jeremy G HOSKINS, Stefan STEINERBERGER et Yuval KLUGER : Efficient algorithms for t-distributed stochastic neighborhood embedding. *arXiv preprint arXiv :1712.09005*, 2017.

[12] George C LINDERMAN et Stefan STEINERBERGER : Clustering with t-SNE, provably. *SIAM journal on mathematics of data science*, 1(2):313–332, 2019.

[13] F. PEDREGOSA, G. VAROQUAUX, A. GRAMFORT, V. MICHEL, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU, M. BRUCHER, M. PERROT et E. DUCHESNAY : Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] Luca REICHMANN, David HÄGELE et Daniel WEISKOPF : Out-of-core dimensionality reduction for large data via out-of-sample extensions. *arXiv preprint arXiv :2408.04129*, 2024.

[15] Edgar ROMAN-RANGEL et Stephane MARCHAND-MAILLET : Inductive t-SNE via deep learning to visualize multi-label images. *Engineering Applications of Artificial Intelligence*, 81:336–345, 2019.

[16] Laurens van der MAATEN et Geoffrey HINTON : Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[17] Zhirong YANG, Yuwei CHEN et Jukka CORANDER : t-SNE is not optimized to reveal clusters in data. *arXiv preprint arXiv :2110.02573*, 2021.