**Advances in Applied Mathematics** 应用数学进展, 2025, 14(4), 584-600
Published Online April 2025 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2025.144189

Hans汉斯

# 基于三元组约束的神经网络降维方法

**翁德海**

广东工业大学数学与统计学院，广东 广州

收稿日期：2025年3月18日；录用日期：2025年4月11日；发布日期：2025年4月21日

## 摘 要

**与传统的降维和可视化方法(如t-SNE)相比，基于三元组约束的降维方法在保持数据全局结构方面表现优异。三元组约束是指三个数据点之间的约束，能够描述该三点的相对相似性。本文提出了一种基于三元组约束的降维模型(Trivis)，该模型能充分利用Siamese神经网络(SNNs)来增强高维数据降维和可视化的局部结构保持能力。与现有的基于三元组约束的降维方法相比，Trivis模型有两方面的改进：一方面，采用了一种新的目标损失函数，涉及除法运算，有助于保持数据的局部结构；另一方面，对SNNs的输入模块进行了增强，允许每个锚点关联多个三元组，从而能更好地捕捉数据中的局部结构。实验结果表明，Trivis模型在保持局部和全局结构方面能取得良好的平衡，提供了一个稳健的高维数据降维可视化解决方案。**

## 关键词

**数据降维，数据可视化，深度学习，度量学习，损失函数**

# A Neural Network Dimensionality Reduction Method Based on Triplets Constraints

**Dehai Weng**

School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou Guangdong

Received: Mar. 18th, 2025; accepted: Apr. 11th, 2025; published: Apr. 21st, 2025

## Abstract

**Compared with traditional dimensionality reduction and visualization methods (e.g., t-SNE), triplet constraint-based dimensionality reduction methods excel in maintaining the global structure of the data. A triplet constraint is a constraint between three data points that describes the relative similarity of those three points. In this paper, we propose a triplet constraint-based dimensionality reduction model (Trivis) that can make full use of Siamese Neural Networks (SNNs) to enhance the**

文章引用: 翁德海. 基于三元组约束的神经网络降维方法[J]. 应用数学进展, 2025, 14(4): 584-600.
DOI: 10.12677/aam.2025.144189

**local structure preservation of high-dimensional data reduction and visualization. Compared with existing triplet-constraint-based dimensionality reduction methods, the Trivis model has two improvements: on one hand, a new objective loss function involving a division operation is adopted, which helps to preserve the local structure of the data; on the other hand, enhancements are made to the input module of the SNNs, which allows multiple triplets to be associated with each anchor point, thus enabling better capture of the local structure in the data. The experimental results show that the Trivis model can strike a good balance between preserving local and global structure, providing a robust solution for downscaling visualization of high-dimensional data.**

## Keywords

**Data Dimensionality Reduction, Data Visualization, Deep Learning, Metric Learning, Loss Function**



## 1. 引言

随着数据维度的不断增加，降维(Dimensionality Reduction, DR)技术在数据预处理、探索性数据分析及可视化等方面的重要性日益凸显。由于人类通常通过二维或三维的方式感知世界，DR 方法始终是数据科学领域的重要研究内容[1] [2]。传统的线性降维方法，如主成分分析(PCA) [3]和线性判别分析(LDA) [4]，已被广泛应用。然而，这类方法基于线性假设，在处理非线性数据时性能往往受到限制，从而催生了更为先进的非线性降维技术。

在非线性降维方法中，t-分布随机邻域嵌入(t-SNE) [5]是一种典型方法，旨在降维的同时保持数据的局部和全局结构。t-SNE 通过最小化原始空间与降维空间之间的 KL (Kullback-Leibler)散度，对数据点之间的相似性进行有效建模。然而，由于其依赖迭代优化且缺乏参数化形式，t-SNE 在扩展性与泛化能力方面仍存在一定局限[6]。

为了解决这些问题，研究者们相继提出了若干改进算法，包括 t-SNE 的优化版本[7]、统一流形近似投影(UMAP) [8] [9]和 TriMap [10]。这些方法在不同应用场景中均表现出色，其中类 t-SNE 和 UMAP 在保持数据的局部结构方面尤为有效，但在全局结构的保留上仍存在不足[11] [12]。作为新兴方法，TriMap 在全局结构保持方面具有一定优势，但其性能较大程度依赖于初始化，而非算法本身的特性[6] [13]。然而，目前尚无单一方法能够同时平衡局部与全局结构的保持。由于各类算法的损失函数设计和参数敏感性存在差异，比较这些方法依然充满挑战[14]。即使是密切相关的算法，如 t-SNE 和 UMAP，它们实现点间排斥力的方式也各不相同，因此直接比较并不容易[11] [15]-[17]。

近年来，参数化降维方法取得了显著进展，如参数化 t-SNE [18]和参数化 UMAP [19]。这类方法利用神经网络实现高效且灵活的数据嵌入，能够在无需重新训练的情况下处理新数据。类似地，ivis [20]通过引入孪生神经网络(Siamese Neural Networks, SNNs) [21] [22]进一步提升了降维效果。此外，Adam 优化器[23]和 Leaky ReLU 激活函数[24] [25]等优化技术在解决局部极小值等问题上也发挥了重要作用，推动了这些方法在多种数据类型与结构上的广泛应用[18]。

## 2. 相关工作

DR 技术是数据可视化与分析中的重要工具，能够将高维数据映射到低维空间，便于直观解读与分

析。经典的 DR 方法，如 PCA 和 LDA，主要通过最大化数据的方差或增强类别分离性来实现降维。然而，这类方法在处理复杂的非线性数据时表现不足。因此，非线性降维方法[26] [27]应运而生，旨在更好地保持数据点之间的局部与全局关系。例如，LargeVis [28] [29]在 t-SNE 基础上优化，去除了嵌入空间的规范化约束，提升了计算效率和扩展性。统一流形近似投影(UMAP) [8]结合代数拓扑理论，能够在降低计算成本的同时保持局部结构，其性能与 t-SNE 和 LargeVis 相当。

随机邻域嵌入(SNE) [30]及其改进版 t-SNE 通过最小化高维空间与低维空间之间的 Kullback-Leibler 散度，有效地保持了数据的局部结构。然而，t-SNE 等非参数方法存在一个显著局限：缺乏显式的映射函数，导致无法有效泛化到新数据。为了解决这一问题，研究者提出了参数化改进版本。

例如，参数化 t-SNE [31]通过引入受限玻尔兹曼机(RBM) [18]预训练的神经网络，进一步扩展了 t-SNE 的功能，使其能够有效泛化到新数据集。同样，UMAP 也发展出参数化版本[19]，利用深度学习框架提高了算法的可扩展性，并有效缓解了局部极小值问题。这些参数化方法将非参数方法的结构保持能力与参数模型的灵活性相结合，标志着 DR 技术取得了重要的突破性进展。

值得注意的是，ivis [20]采用了带三元组损失函数的有监督神经网络架构，在模拟数据与单细胞数据分析中展现出较强的全局结构保持能力。然而，ivis 在局部结构的保持方面仍存在不足。尽管 ivis 引入了显式的映射函数，使得新数据点能够被无缝添加，但局部结构的保持性能仍有提升空间。

针对这一问题，本文提出了一种新的参数化降维框架——Trivis。该框架旨在增强数据局部结构的保持能力，同时继承现有方法在全局结构保持方面的优势，从而在局部与全局结构保持之间实现更好的平衡。

## 2.1. 降维方法

假设数据集为 $X=\left[\boldsymbol{x}_1^{\mathrm{T}},\boldsymbol{x}_2^{\mathrm{T}},\cdots,\boldsymbol{x}_n^{\mathrm{T}}\right]^{\mathrm{T}}\in R^{n\times D}$，其中行向量 $\boldsymbol{x}_i$ 表示 $D$ 维数据点，T 表示向量和矩阵的转置。在实践中，特征维度 $D$ 通常较高。降维的目标是将矩阵 $X\in R^{n\times D}$ 转换为低维矩阵 $Y\in R^{n\times D}$ $(d<D)$，通过映射 $f$ 在 $Y$ 中保持 $X$ 中样本点之间的关键相对关系，可形式地表示为：

$$\boldsymbol{Y}^{n\times d}=f\left(\boldsymbol{X}^{n\times D}\right) \tag{1}$$

公式(1)表示数据集中每个高维数据点 $\boldsymbol{x}_i\in X$ 被映射为低维空间中的一个 $d$ 维向量 $\boldsymbol{y}_i\in Y$。通常，映射 $f$ 是未知的，需要根据高维样本点的位置关系通过学习进行确定。为了定义或学习该映射 $f$，需要准确地描述并捕捉样本点之间的关系，以便在降维过程中尽可能保留这些关系。一般而言，矩阵 $X$ 中的所有条目为数值型数据，而非类别型变量。

## 2.2. 非参数降维方法

一种有效学习映射 $f$ 的方法是 t-SNE 模型。t-SNE 是一种非参数技术，在降维的同时保留数据的局部结构。t-SNE 的映射 $f$ 是通过最小化高维空间样本点概率矩阵与低维空间基于 t-分布的联合概率矩阵之间的 KL 散度来确定的。KL 散度越小，高维与低维空间之间的样本点概率矩阵越相似。在高维空间中，样本点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的概率定义如下：

$$P_{ij}=\frac{\exp\left(-\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\left\|\boldsymbol{x}_i-\boldsymbol{x}_k\right\|^2/2\sigma_i^2\right)} \tag{2}$$

其中，$\sigma_i$ 是点 $\boldsymbol{x}_i$ 的高斯核带宽，符号 $\|\bullet\|$ 表示欧几里得距离。设置 $P_{ii}=0$，此时概率矩阵的行和为 1。从公式(2)可知，当两点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 之间的距离越小，概率 $P_{ij}$ 越大，说明两点之间的相似性越强。此外，$\sigma_i$ 的

值可以用来调整点 $\boldsymbol{x}_i$ 周围的邻域尺度，并影响其局部相似性结构的构建。较大的 $\sigma_i$ 值扩大了 $\boldsymbol{x}_i$ 的邻域范围，使更远的点也会对相似性产生贡献；较小的 $\sigma_i$ 值则收紧邻域范围，更加强调近邻点的影响。合理选择或调整 $\sigma_i$ 有助于捕捉数据的局部结构特征。

类似地，在低维空间中，基于 t-分布的样本点 $\boldsymbol{y}_i$ 和 $\boldsymbol{y}_j$ 的联合概率定义如下：

$$Q_{ij} = \frac{\left(1+\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|^2\right)^{-1}}{\sum_{k\neq i}\left(1+\left\|\boldsymbol{y}_k - \boldsymbol{y}_i\right\|^2\right)^{-1}} \tag{3}$$

同样设置 $Q_{ii} = 0$，此时概率矩阵的行和为 1。与 $P_{ij}$ 类似，$Q_{ij}$ 反映了低维空间中样本点之间的相似性结构。

更具体地，t-SNE 中的映射 $f$ 通过最小化 KL 散度得到；两个概率矩阵之间的 KL 散度，定义如下：

$$C = \sum_i \sum_j P_{ij} \log \frac{P_{ij}}{Q_{ij}} \tag{4}$$

在优化过程中，公式(4)的值不断减小，高维和低维空间概率矩阵元素 $P_{ij}$ 和 $Q_{ij}$ 之间的不一致性逐渐降低。最终，低维空间中的数据点 $\boldsymbol{y}_i$ 的位置关系尽可能准确地反映高维空间中数据点 $\boldsymbol{x}_i$ 的位置关系。

综上，从公式(2)到(4)可以看出，t-SNE 中的映射 $f$ 是一种从高维数据点到低维数据点的非参数映射，能有效捕捉高维空间中的局部结构。

### 2.3. 参数化降维方法

t-SNE 和 UMAP 是典型的非参数方法。而其参数化版本(即参数化 t-SNE 和参数化 UMAP)通过引入基于神经网络的方法学习未知的映射函数 $f$，从而提供了更大的灵活性和可扩展性。一旦神经网络完成了对 $f$ 的学习，参数化方法便能够高效处理新数据点，而无需重新训练模型。

参数化降维方法旨在通过学习一个显式的映射函数来扩展非参数方法的灵活性，使其能够直接且轻松地应用于新数据点。例如，参数化 t-SNE 使用受限玻尔兹曼机(RBMs)来近似映射 $f$ [32]。RBMs 通常采用分层堆叠的方式进行预训练，然后通过优化 t-SNE 的目标函数进行微调。类似地，参数化 UMAP 使用典型的随机梯度下降深度学习框架来学习映射 $f$。这种神经网络通过保留数据的拓扑结构来训练，类似于非参数 UMAP 的方法。神经网络的架构通常由多层神经元组成，并配有激活函数，使网络能够学习从高维到低维的复杂映射关系。

参数化 t-SNE 和参数化 UMAP 都利用深度学习优化器和激活函数来克服局部极小值问题，并增强降维过程的可扩展性。引入这些参数化映射函数标志着降维领域的显著进步，使降维技术的应用更加多样化且具有更强的可扩展性。

使用神经网络的参数化 DR 方法有以下几点优势：

- 可扩展性：神经网络能够高效处理大规模数据集，非常适合处理涉及海量数据的实际应用场景[18]。
- 泛化能力：训练完成后，神经网络能够很好地泛化到新数据点，在不同数据集上表现出较强的鲁棒性[19]。
- 灵活性：神经网络通过其分层架构可以学习数据中的复杂转换关系，从而支持多种数据变换和交互方式[31]。
- 高效性：参数化模型可以使用基于梯度的优化方法，确保即使在复杂任务中也能够实现高效的训练[8]。

## 3. 基于三元组的 DR 方法

### 3.1. 基于三元组的非参数化方法

最初，嵌入技术(如 t-SNE、UMAP 和 LargeVis)主要关注点对点的关系。这些方法利用点对点距离度量(如欧几里得距离)来保留数据点之间的空间关系。然而，点对点方法仅关注相邻点，往往忽略远距离点，从而在处理复杂数据结构时可能无法捕捉更复杂的关系。

为了解决这一局限，DR 技术逐渐发展为反映三元组的相对(非)相似性，而不局限于点对点关系。形式上，三元组是由三个数据点 $a$、$p$、$n$ 构成的约束，简记成 $(a,p,n)$，其含义是“点 $a$ 与点 $p$ 的相似性大于点 $a$ 与点 $n$ 的相似性”。与仅关注点对点距离或相似性的度量不同，t-分布随机三元组嵌入(t-STE) [33] 利用三元组约束来更好地捕捉数据点之间的复杂关系。三元组约束框架的设计灵感来自人类对于物品相似性判断的方式[34]，通常被认为更加可靠。在本文中，定义三元组集合为：

$$T=\left\{(a,p,n)\mid \boldsymbol{x}_a \text{与} \boldsymbol{x}_p \text{的相似性大于} \boldsymbol{x}_n \text{的相似性}\right\} \tag{5}$$

在 t-STE 中，三元组的概率 $P_{t_{apn}}$ 定义如下：

$$P_{t_{apn}}=\frac{\left(1+\frac{\left\|\boldsymbol{y}_a-\boldsymbol{y}_p\right\|^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}}{\left(1+\frac{\left\|\boldsymbol{y}_a-\boldsymbol{y}_p\right\|^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}+\left(1+\frac{\left\|\boldsymbol{y}_a-\boldsymbol{y}_n\right\|^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}} \tag{6}$$

其中，$\alpha$ 是 t-分布的自由度。当点 $a$ 和 $p$ 之间的距离减小时，概率 $P_{t_{apn}}$ 增大，意味着 $a$ 和 $p$ 更相似；反之，当 $a$ 和 $n$ 的距离增大时，概率 $P_{t_{apn}}$ 也会增大，意味着 $a$ 和 $n$ 的相似性更低。因此，概率 $P_{t_{apn}}$ 可以用来描述三元组 $(a,p,n)$ 的相对(非)相似性。

求解 DR 转换映射 $f$ 的方法是最大化所有三元组约束的对数似然函数：

$$L=\sum_{(a,p,n)\in T}\log P_{t_{apn}} \tag{7}$$

在现有文献中，点 $a$ 被称为锚点，点 $p$ 被称为正样本，点 $n$ 被称为负样本。最大化目标函数 $L$ 的过程旨在学习嵌入 $f$ 函数，使得在低维空间中，锚点与正样本之间的距离更近，而与负样本的距离更远。这种方法确保嵌入后的点在低维空间中能够更准确地反映数据点之间的相对关系。

在文献[13]中，受 t-STE 启发，提出了一种新的降维可视化方法 TriMap，其重点在于更准确地保留全局结构。TriMap 中，三元组 $(a,p,n)$ 的满足概率(satisfaction probability)定义为：

$$P_{newt_{apn}}=\frac{E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)}{E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)+E\left(\boldsymbol{y}_a,\boldsymbol{y}_n\right)}=\frac{1}{1+\frac{E\left(\boldsymbol{y}_a,\boldsymbol{y}_n\right)}{E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)}} \tag{8}$$

当 $E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)\gg E\left(\boldsymbol{y}_a,\boldsymbol{y}_n\right)$ 时，满足概率 $\left(P_{newt_{apn}}\to 1\right)$。目标函数定义为：

$$L_{\text{TriMap}}=\sum_{(a,p,n)\in T}\omega_{apn}\frac{E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)}{E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)+E\left(\boldsymbol{y}_a,\boldsymbol{y}_n\right)} \tag{9}$$

其中 $E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)=\left(1+\left\|\boldsymbol{y}_a-\boldsymbol{y}_p\right\|^2\right)^{-1}$，$\omega_{apn}$ 是分配给三元组 $(a,p,n)$ 的权重。当 $\left\|\boldsymbol{y}_a-\boldsymbol{y}_p\right\|$ 减小时，$P_{newt_{apn}}$ 增大；

而当$\|\boldsymbol{y}_a-\boldsymbol{y}_n\|$增大时，$P_{newt_{apn}}$同样增大。

在公式(9)中，TriMap 首次引入了权重$\omega_{apn}$，以反映高维空间中的相对相似性。三元组$(a,p,n)$的初始权重定义为：

$$\tilde{\omega}_{apn}=d_{an}^2-d_{ap}^2\geq 0 \tag{10}$$

其中$d_{ap}$是高维空间中$\boldsymbol{x}_a$和$\boldsymbol{x}_p$之间的距离度量。文献[35]中引入了一种缩放距离度量：

$$d_{ap}^2=\frac{\|\boldsymbol{x}_a-\boldsymbol{x}_p\|^2}{\sigma_{ap}} \tag{11}$$

其中$(\sigma_{ap}=\sigma_a\sigma_p)$，$\sigma_a$是点$\boldsymbol{x}_a$与其第 4 至第 6 邻居之间的平均欧几里得距离。这种$\sigma_{ap}$的选择根据数据密度自适应调整缩放。

在 TriMap 中，最终权重$\omega_{apn}$通过减去最小初始权重值后再进行对数变换得到，定义为：

$$\omega_{apn}=\log_t\left(1+\tilde{\omega}_{apn}-\omega_{\min}\right) \tag{12}$$

其中：

$$\omega_{\min}=\min_{(a',p',n')\in T}\tilde{\omega}_{a'p'n'}$$

并且：

$$\log_t(u)=\frac{1}{1-t}\left(u^{1-t}-1\right),\quad t\neq 1$$

### 3.2. 基于三元组的参数化方法：ivis

通常，结构保持型 DR 需要复杂的神经网络架构设计。显然，三元组关系与点对点关系有着本质的区别。因此，相较于点对点关系，三元组降维方法的参数化需要一种不同于点对点的网络架构。为此，一种高效的三元组网络架构[36]被提出，该架构的开发基于孪生神经网络(Siamese Neural Networks, SNNs) [37]。

该架构通过独特的设计在评估输入相似性上表现出色，包含三个完全相同的基础网络。每个网络由三层密集层(每层包含 128 个神经元)组成，最终连接到一个嵌入层，该层的维度由目标输出维度决定。例如，在降维到二维空间时，嵌入层输出就选取为两个神经元。

SNNs 的密集层采用缩放指数线性单元(SELU)激活函数，其数学定义为：

$$selu(x)=\lambda\begin{cases}x, & \text{if } x>0\\ \alpha e^x-\alpha, & \text{if } x\leq 0\end{cases} \tag{13}$$

其中，$\lambda$和$\alpha$分别取值为 1.05 和 1.67。这种配置赋予了网络自正则化的特性，有助于将激活值保持在最佳范围内。隐藏层的权重参数采用 LeCun 正态分布初始化，而嵌入层采用线性激活函数，并使用 Glorot 均匀分布初始化权重参数。LeCun 正态分布和 Glorot 均匀分布的数学定义为：

$$N\left(0,\sqrt{\frac{1}{n}}\right),u\left(-\sqrt{\frac{6}{n_{in}+n_{out}}},\sqrt{\frac{6}{n_{in}+n_{out}}}\right), \tag{14}$$

其中，$n$表示输入到隐藏层的神经元数量，$n_{in}$和$n_{out}$分别为嵌入层的输入和输出神经元数量。

为避免过拟合并增强模型鲁棒性，SNNs 的密集层之间引入了 Alpha Dropout 层，选择 0.1 的丢弃率。这种策略不仅降低了过拟合风险，还通过保持输入的均值和方差支持 SELU 的自正则化特性。

Hoffer 和 Nir [36]指出，SNNs 非常适合用于三元组类型的降维任务。在此启发下，文献[20]提出了基

于 SNNs 的降维框架 ivis，其损失函数定义为

$$L_{\text{ivis}} = \max\left[\sum_{(a,p,n)\in T} E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right) - \min\left(E\left(\boldsymbol{y}_a,\boldsymbol{y}_n\right),E\left(\boldsymbol{y}_p,\boldsymbol{y}_n\right)\right)+m,0\right] \tag{15}$$

其中 $E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)=\left\|\boldsymbol{y}_a-\boldsymbol{y}_p\right\|$，$m$ 为边界(一个小的正数)，其作用是控制降维过程中参与求和的三元组数量。损失函数 $L_{\text{ivis}}$ 是标准三元组损失函数的一种变体[38] [39]，能有效替代传统三元组损失函数和 softmax-ratio 损失函数[36]。最小化 $L_{\text{ivis}}$ 的目标是确保锚点和正样本之间的距离比锚点和负样本之间的距离至少小值 $m$。

SNNs 的训练依赖于三元组目标函数 $L_{\text{ivis}}$，目标是通过遵循每个三元组的相对距离约束来优化嵌入空间。虽然公式中未显式包含高维空间中数据点的信息，但 SNNs 网络架构中的输入为高维数据点，这意味着高维空间中的信息在 ivis 的训练过程中被逐步学习。

### 3.3. 新三元组参数降维方法：Trivis

针对在第 1 节提到的 ivis 模型局部结构保持方面存在不足的这一问题，本文设计了一种针对三元组局部结构优化的新的损失函数和采样方法。

ivis 方法在全局结构保持方面表现出色，但在局部结构保持方面仍存在不足。这种不足的原因可能是公式(15)中的目标函数 $L_{\text{ivis}}$ 仅涉及三元组中的减法运算，而减法运算难以准确描述数据点在三元组约束情况下低维空间中数据点的局部相对结构。受[10] [33]的启发，本文提出了一种新的目标损失函数，定义如下：

$$L_{\text{Trivis}} = \sum_{(a,p,n)\in T} \frac{E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)}{E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)+E\left(\boldsymbol{y}_a,\boldsymbol{y}_n\right)} \tag{16}$$

其中，$E\left(\boldsymbol{y}_a,\boldsymbol{y}_p\right)=1+\left\|\boldsymbol{y}_a-\boldsymbol{y}_p\right\|^2$。一般来说，可以任意采用其他距离度量方式。基于 SNNs，我们提出了一个新的降维框架(Trivis)，其标准三元组目标函数为 $L_{\text{Trivis}}$。图 1 展示了 Trivis 的技术流程图。最终的损失函数 $L_{\text{Trivis}}$ 在整个降维框架中使用带动量的全批量梯度下降法进行最小化。

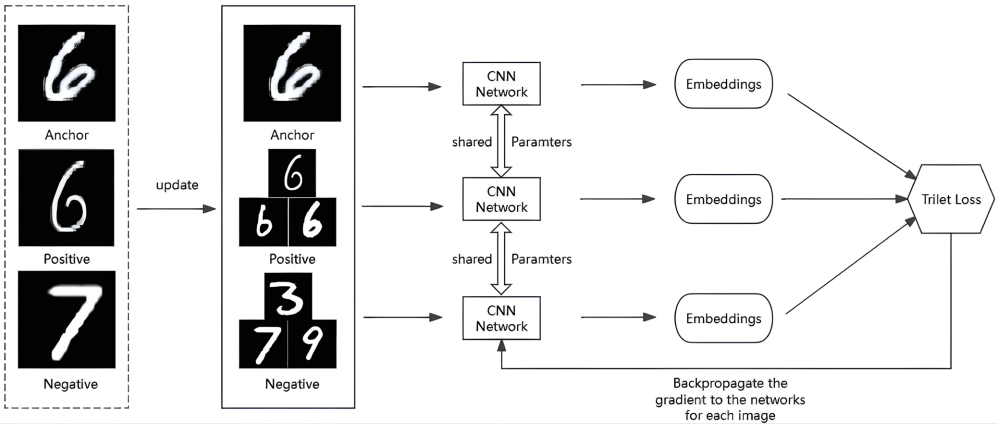



**Figure 1.** Sampling method optimization
**图 1.** 采样方法优化

相比于公式(15)，Trivis 的目标函数(16)引入了除法来计算损失函数，使其值域落在了区间(0, 1)之中。这一修改有助于在低维空间中更好地保持数据点的局部结构。这种损失函数的改变使得三元组之间相似

性比例的评估更加平衡，从而能够准确反映高维空间中点的相对位置关系，进一步增强了局部结构的保持能力。

本文也对三元组的采样进行了改进，细节如下：在原始 SNNs 中，一个锚点仅对应一个三元组；在新的 SNNs 中，一个锚点可以对应多个三元组(参见图 1)。这一改进，让一个锚点在降维过程中有更多的特征进行位置选取，在保持全局结构的同时，显著增强了局部结构的保持能力。有关 Trivis 训练的更多细节和改进，请参见算法 1。

通过以上设计，Trivis 在降维任务中表现出色，能够有效捕捉和平衡数据的局部和全局结构。

**算法 1** Trivis 算法

**输入**：数据集 $X$，嵌入维度 $d$

**输出**：低维映射 $Y$

**初始化**：构建或加载 Annoy KNN 矩阵
基于 Annoy KNN 矩阵得到三元组邻接矩阵
定义三元组损失函数 $L_{\text{Trivis}}$，指定距离度量
使用优化器和 $L_{\text{Trivis}}$ 编译 SNNs 模型

**for** 1: TotalEpochs
  根据邻接矩阵从数据集中生成多个三元组
  对这些三元组进行前向传播和反向传播训练模型
    **if** 验证损失满足提前停止条件 **then**
    终止训练
    **end if**
**end for**

**返回**：训练好的模型，可将高维数据 $X$ 映射到低维嵌入 $Y$

算法 1 呈现了 Trivis 的伪代码。显然，三元组的采样在 Trivis 框架中起着关键的作用。在训练过程中动态生成三元组对于捕捉数据的局部和全局特性非常重要。一般地，在训练过程中每一个点都会被作为锚点输入；初始构建的 Annoy KNN 矩阵涵盖了 K 近邻的全部信息，三元组可以基于此矩阵进行样本点选择；Trivis 从 Annoy KNN 矩阵中选择输入锚点的 K 近邻作为正样本，选择非 K 近邻作为负样本。

与原始 ivis 算法相比，Trivis 在输入模块上做了增强改进。在原始 ivis 算法中，每个锚点仅对应一个三元组；而在 Trivis 中，每个锚点可以对应多个三元组。具体而言，Trivis 为每个锚点允许选择 $nm$ 个三元组，其中 $n$ 是正样本数，$m$ 是负样本数。显然，$n$ 和 $m$ 可以作为可调节的超参数。通过增加 $n$ 和 $m$，就能提供更多的相似性和非相似性约束，从而改进 Trivis 的局部收敛能力。这种灵活性使得 Trivis 能够更好地捕捉数据的全局和局部结构。

然而，需要注意的是，选取过大的 $n$ 和 $m$ 可能会给 SNNs 带来显著的计算负担，从而降低降维过程的效率。对于参数的选择，在我们的实验中，选定 $n=3$，$m=3$，以平衡局部收敛优化与计算效率。这种选择使得模型能够在提升局部收敛性的同时，保持降维过程的整体效率。事实上，如果使用大量三元组会带来很大的计算开销，并且需要大量迭代才能收敛。

此外，本文中的实验训练还设置了提前中止训练机制：SNNs 在 1000 个训练轮中以 128 个样本的小批量进行训练，使用 Adam 优化器进行高效的反向传播；当验证集上的损失在 50 个连续轮次中不再下降时，将提前停止训练。这一机制确保了训练效率，同时不会降低训练质量，从而有效降低了三元组参数化降维方法的计算负担。

## 4. 实验

### 4.1. 基准方法

为了评估全局结构的保持能力，我们使用了原始 ivis 和 PCA 作为基准方法，通过直接降低维度并对比从三个 3D 数据集的可视化结果来比较它们的全局结构保持效果。为了评估局部结构的保持能力，我们将经典的t-SNE方法及其参数化版本p-tSNE作为基准方法，利用K近邻(KNN)指标和支持向量机(SVM)指标对改进模型的局部结构保持能力进行评估。

### 4.2. 质量指标

与其他文献类似，我们使用带标签的数据集来评估不同降维方法的质量。在算法完成降维后，利用标签进行评估；本文考虑了一个全局结构指标和二个局部结构指标。

一般地，全局结构指标通过质心三元组准确率(Centroid Triplet Accuracy, CTA)来衡量。另外，我们还采用了对三维数据降维后的嵌入结果进行直观地可视化，即将其降至二维来观察，以此评估全局结构是否得以保持。这种方法能让我们直观地辨别出高维空间中数据点的总体排列和关系在低维表示里是否得到了保持。

局部结构保持的质量主要通过两种指标来评估。首先，应用留一法交叉验证，并使用 K 近邻(KNN)分类器进行评估。这已成为降维方法评估的标准方法。其背后的原理是，标签在小的邻域内通常具有相似性，因此基于邻域的分类准确性应与高维空间中的分类准确性保持接近。此外，如果降维方法未能保持邻域关系，我们预期 KNN 的预测准确度会下降。定义 KNN 准确率为：

$$\text{KNN Accuracy} = \frac{1}{n}\sum_{i=1}^{n} I\left(\boldsymbol{y}_i = \hat{\boldsymbol{y}}_i\right) \tag{17}$$

其中 $I(\bullet)$ 是指示函数， $\boldsymbol{y}_i$ 是第 $i$ 个样本的真实标签，而 $\hat{\boldsymbol{y}}_i$ 是 KNN 预测的标签。

此外，我们还借助非线性 SVM 模型来评估局部结构的准确性。采用径向基函数 RBF 核，并通过交叉验证的方式进行评估。与 KNN 相似，SVM 准确率衡量的是邻域的紧凑程度，不过它在处理密度不均的数据时更具灵活性。具体而言，对于每种 DR 方法，我们将嵌入数据划分为 5 个部分，每次利用其中 4 个部分训练 SVM 模型，再用剩余的部分进行准确性评估。定义 SVM 准确率为：

$$\text{SVM Accuracy} = \frac{1}{n}\sum_{i=1}^{n} I\left(\boldsymbol{y}_i = \hat{\boldsymbol{y}}_i^{\text{SVM}}\right) \tag{18}$$

其中 $\hat{\boldsymbol{y}}_i^{\text{SVM}}$ 是由训练好的支持向量机(SVM)模型预测的标签。

### 4.3. 实验过程与结果

在本节中，将 Trivis 应用于一组真实的数据集，并与 t-SNE、参数化 t-SNE、ivis、PCA、UMAP 和 PaCMAP 方法进行比较。在实验中，使用的数据集包括 3D 数据集 S-curve、Swiss roll、Mammoths 和高维数据集 20NG、COIL-20、USPS、MNIST 和 FMNIST，其中所有参数分析将用模拟数据集来进行实验。所有实验均在一台配备 3.00 GHz Intel Core i5 CPU 和 16 GB 内存的计算机上运行。在程序实现上，使用了 t-SNE 和 PCA 的默认 sklearn 实现，而参数化 t-SNE 和 ivis 则基于 GitHub 上的项目实现。表 1~3 中的指标值均为重复 50 次实验取均值得到的结果。

图 2~4 展示了几个 3D 数据集的二维可视化结果。从展示结果可以看出：Trivis 和 ivis 在保持全局结构方面表现良好；PCA 模型也能够有效的保持全局结构；而 t-SNE 和 p-tSNE 模型则在保持全局结构方面表现较差。这表明，Trivis 成功地继承了 ivis 在全局结构保持方面的优点。

**Table 1.** CTA values of ivis and Trivis
**表 1.** ivis 和 Trivis 的 CTA 指标值

| Dataset | ivis | Trivis |
|---|---|---|
| COIL-20 (1.4 K) | 0.7655 | 0.8021 |
| USPS (9 K) | 0.8515 | 0.8521 |
| 20NEWSGROUPS (18 K) | 0.7876 | 0.7905 |
| MNIST (70 K) | 0.7757 | 0.7858 |
| F-MNIST (70 K) | 0.8505 | 0.8424 |

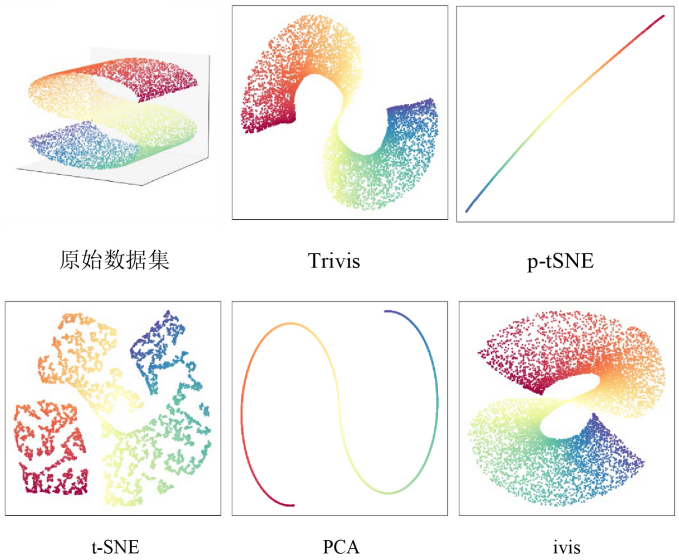



**Figure 2.** Results of DR for each model of the 3D dataset S-curve
**图 2.** 三维数据集 S-curve 各模型降维结果

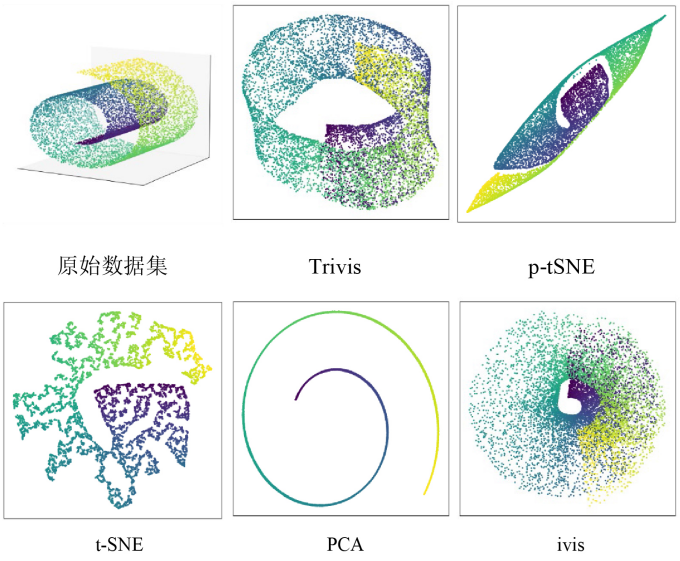



**Figure 3.** Results of DR for each model of the 3D dataset Swiss roll
**图 3.** 三维数据集 Swiss roll 各模型降维结果

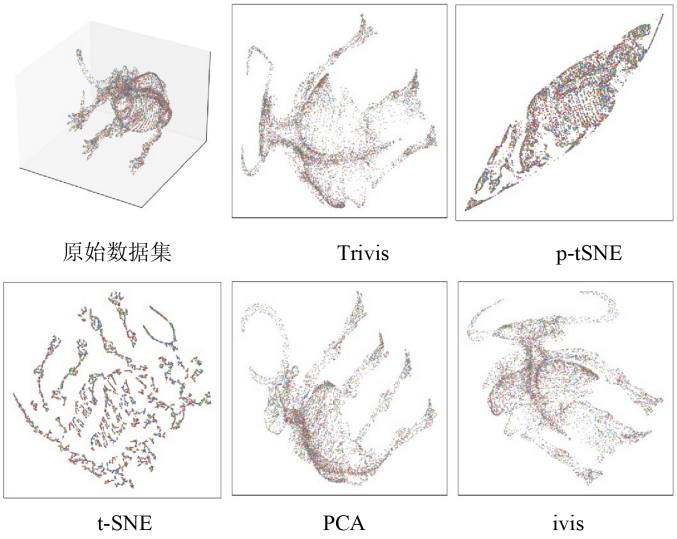



**Figure 4.** Results of DR for each model of the 3D dataset Mammoths
**图 4.** 三维数据集 Mammoths 各模型降维结果

具体地说，从图 2~4 都可以看出 ivis，Trivis 和 PCA 都保留了嵌入空间中每个簇的形状和相对位置，而 t-SNE 和 p-tSNE 的嵌入都产生了额外的虚假簇，并且完全丢失了所有簇间关系。表 1 列出了几个高维数据集的降维全局结构指标 CTA 的值；从该指标值可以看出，Trivis 比 ivis 更能保持全局结构。

表 2，表 3 分别列出了几个高维数据集的降维局部结构 KNN 指标值和 SVM 指标值。从这两个指标值可以看出，尽管 Trivis 模型比不上 t-SNE 模型，但 Trivis 模型的局部结构能力相较于 ivis 模型得到了明显的改善。

**Table 2.** KNN metrics for each dataset in each model
**表 2.** 各模型中各数据集的 KNN 指标

(a)

| Dateset (size) | t-SNE | p-tSNE | Trivis | ivis | PCA |
|---|---|---|---|---|---|
| COIL-20 (1.4 K) | 0.993 ± 0.005 | 0.602 ± 0.020 | 0.799 ± 0.004 | 0.744 ± 0.001 | 0.667 ± 0.001 |
| USPS (9 K) | 0.963 ± 0.002 | 0.926 ± 0.006 | 0.899 ± 0.008 | 0.710 ± 0.010 | 0.475 ± 0.001 |
| 20NEWSGROUPS (18 K) | 0.575 ± 0.014 | 0.135 ± 0.031 | 0.362 ± 0.006 | 0.188 ± 0.013 | 0.091 ± 0.001 |
| MNIST (70 K) | 0.960 ± 0.012 | 0.943 ± 0.160 | 0.967 ± 0.010 | 0.613 ± 0.007 | 0.379 ± 0.001 |
| F-MNIST (70 K) | 0.805 ± 0.008 | 0.703 ± 0.010 | 0.795 ± 0.010 | 0.625 ± 0.006 | 0.452 ± 0.001 |

(b)

| Dateset (size) | UMAP | PaCMAP |
|---|---|---|
| COIL-20 (1.4 K) | 0.915 ± 0.015 | 0.964 ± 0.021 |
| USPS (9 K) | 0.963 ± 0.010 | 0.962 ± 0.013 |
| 20NEWSGROUPS (18 K) | 0.555 ± 0.014 | 0.547 ± 0.016 |
| MNIST (70 K) | 0.963 ± 0.006 | 0.966 ± 0.009 |
| F-MNIST (70 K) | 0.771 ± 0.011 | 0.752 ± 0.010 |

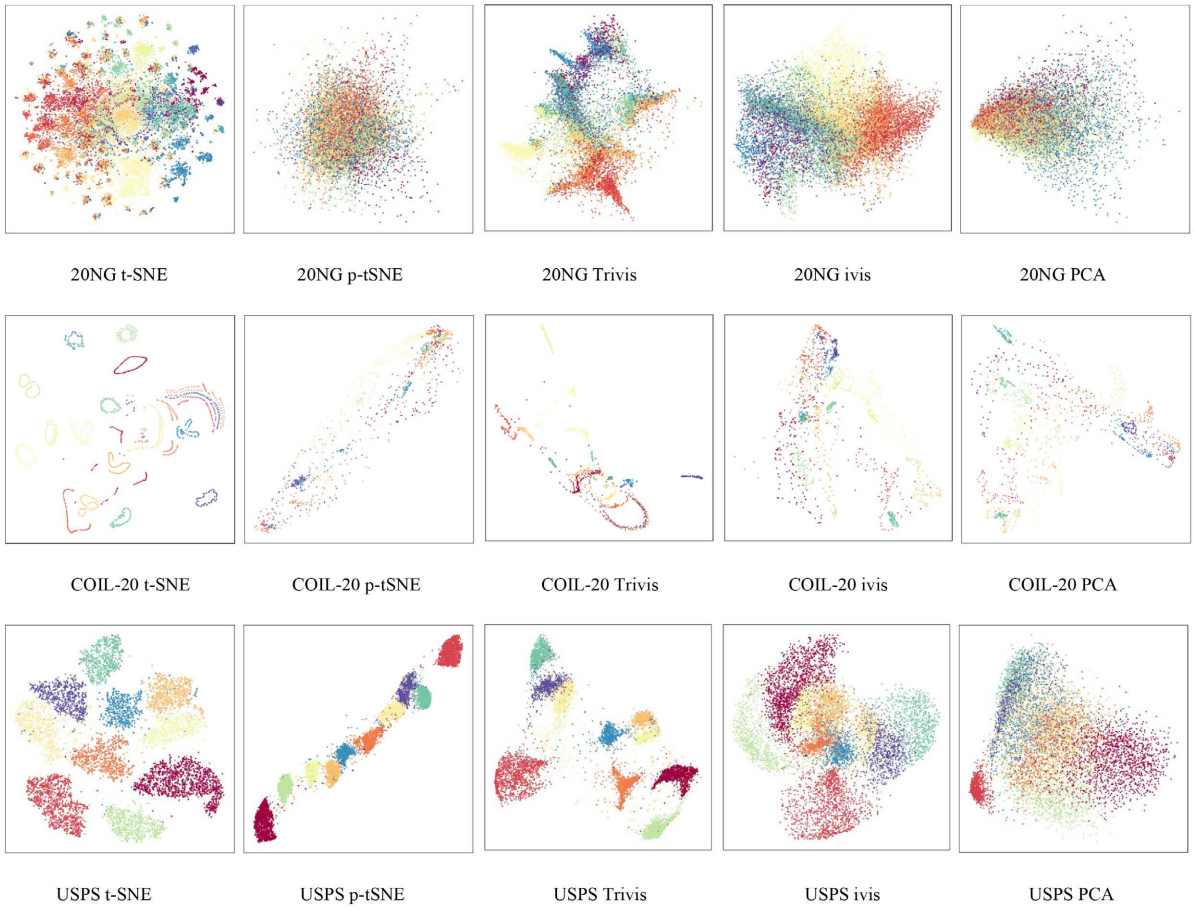
**Table 3.** SVM metrics for each dataset in each model
**表 3.** 各模型中各数据集的 SVM 指标

(a)

| Dateset (size) | t-SNE | p-tSNE | Trivis | ivis | PCA |
|---|---|---|---|---|---|
| COIL-20 (1.4 K) | 0.831 ± 0.015 | 0.441 ± 0.021 | 0.799 ± 0.004 | 0.744 ± 0.001 | 0.649 ± 0.001 |
| USPS (9 K) | 0.959 ± 0.002 | 0.940 ± 0.016 | 0.928 ± 0.011 | 0.710 ± 0.010 | 0.564 ± 0.001 |
| 20NEWSGROUPS (18 K) | 0.435 ± 0.014 | 0.127 ± 0.046 | 0.374 ± 0.006 | 0.188 ± 0.013 | 0.146 ± 0.001 |
| MNIST (70 K) | 0.967 ± 0.002 | 0.964 ± 0.121 | 0.957 ± 0.010 | 0.714 ± 0.007 | 0.468 ± 0.001 |
| F-MNIST (70 K) | 0.748 ± 0.003 | 0.739 ± 0.010 | 0.762 ± 0.010 | 0.685 ± 0.006 | 0.556 ± 0.001 |

(b)

| Dateset (size) | UMAP | PaCMAP |
|---|---|---|
| COIL-20 (1.4 K) | 0.814 ± 0.004 | 0.939 ± 0.009 |
| USPS (9 K) | 0.928 ± 0.002 | 0.956 ± 0.001 |
| 20NEWSGROUPS (18 K) | 0.416 ± 0.014 | 0.446 ± 0.006 |
| MNIST (70 K) | 0.959 ± 0.001 | 0.967 ± 0.031 |
| F-MNIST (70 K) | 0.729 ± 0.003 | 0.734 ± 0.004 |



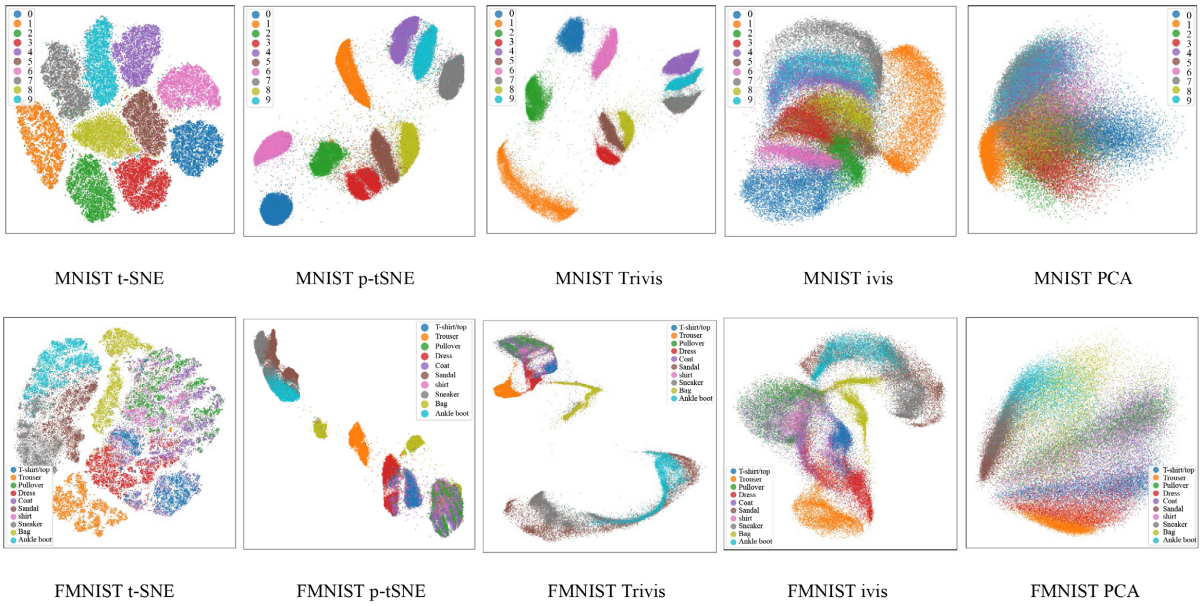



**Figure 5.** Visualisation results of DR of high-dimensional datasets in each model
**图 5.** 各模型中高维数据集的降维可视化结果

图 5 展示了几个高维数据集的二维可视化结果。从图 5 可看出，ivis 在保持全局结构方面的效果与 PCA 模型相当；Trivis 在保持局部结构方面显著优于原始 ivis；t-SNE 和 p-tSNE 模型在保持局部结构方面与表现相当；Trivis 的类簇比 t-SNE 和 p-tSNE 的类簇分得更开，区分更明显。特别地，在 Trivis 模型下，MNIST 和 FMNIST 数据集中的全局结构保持效果得到了显著提升。具体地说，针对 MNIST 数据集，可看出手写数字 4，7 和 9 之间具有较强的相似性；手写数字 3，5 和 8 之间的相似性也比较显著；而手写数字 1 和 2 与其他数字的相似性较弱；手写数字 0 和 6 之间存在一定的相似性。这与我们对手写体数字 0~9 的目力识别感觉一致，突显了 Trivis 方法在保持全局结构方面的优势；而通过 t-SNE 模型降维得到的类簇之间区分不明显，没有得到类似信息。针对 FMNIST 数据集，Trivis 模型降维得到的鞋类簇、包类簇和服装类簇之间区分明显；而其他的降维方法没能得到如此强度的类簇区分信息。

总的来说，以上实验结果展示了 Trivis 模型保持降维局部结构和全局结构方面的能力；全局和局部结构的双重保持能力，突显了 Trivis 在数据可视化中的综合有效性。

## 4.4. 参数敏感性分析

### 4.4.1. 采样参数分析

为了深入探究多三元组采样策略中参数 $n$ 和 $m$ 的选择对模型性能的影响，我们进行了一系列实验。利用 Python 的 numpy 和 scikit-learn 库模拟生成了多个不同规模和特征的数据集。例如，生成一个具有 500 个样本、10 维特征的数据集 simulated_data_1，数据分布近似高斯分布；再生成一个包含 800 个样本、20 维特征的数据集 simulated_data_2，数据分布具有一定的聚类结构。

针对每个模拟数据集，设定不同的 $n$ 和 $m$ 值组合，包括$(n = 1, m = 1)$、$(n = 2, m = 2)$、$(n = 3, m = 3)$、$(n = 5, m = 5)$以及$(n = 7, m = 7)$。对每个组合，使用 Trivis 模型进行 10 次降维实验，降维后的维度设定为 2 维。

从表 4 可以看出，随着 $n$ 和 $m$ 值的增加，模型在保持数据局部和全局结构方面的性能总体呈上升趋势，但当 $n$ 和 $m$ 过大(如 $n = 7, m = 7$)时，性能提升幅度变缓，且可能会增加计算负担。在不同数据集上，

合适的 $n$ 和 $m$ 值有所差异。对于具有近似高斯分布的 simulated_data_1 数据集，($n$ = 5, $m$ = 5)时模型性能相对较好；而对于具有聚类结构的 simulated_data_2 数据集，($n$ = 5, $m$ = 5)同样表现出较好的性能，但在实际应用中，还需结合具体数据特点和计算资源来选择合适的参数。

**Table 4.** Simulation dataset sampling parameter analysis indicators
**表 4.** 模拟数据集采样参数分析指标

| Dataset | ($n$, $m$) | CTA | KNN 准确率 | SVM 准确率 |
|---|---|---|---|---|
| simulated_data_1 | (1, 1) | 0.653 ± 0.021 | 0.725 ± 0.018 | 0.684 ± 0.023 |
| simulated_data_1 | (2, 2) | 0.702 ± 0.019 | 0.786 ± 0.015 | 0.745 ± 0.020 |
| simulated_data_1 | (3, 3) | 0.751 ± 0.016 | 0.823 ± 0.012 | 0.782 ± 0.018 |
| simulated_data_1 | (5, 5) | 0.780 ± 0.014 | 0.851 ± 0.010 | 0.810 ± 0.015 |
| simulated_data_1 | (7, 7) | 0.775 ± 0.015 | 0.845 ± 0.011 | 0.802 ± 0.016 |
| simulated_data_2 | (1, 1) | 0.582 ± 0.025 | 0.654 ± 0.020 | 0.613 ± 0.025 |
| simulated_data_2 | (2, 2) | 0.640 ± 0.022 | 0.720 ± 0.018 | 0.675 ± 0.022 |
| simulated_data_2 | (3, 3) | 0.705 ± 0.019 | 0.785 ± 0.015 | 0.730 ± 0.020 |
| simulated_data_2 | (5, 5) | 0.750 ± 0.016 | 0.830 ± 0.012 | 0.775 ± 0.018 |
| simulated_data_2 | (7, 7) | 0.740 ± 0.017 | 0.820 ± 0.013 | 0.760 ± 0.019 |

### 4.4.2. 模型参数分析

为了探究不同参数设置对 Trivis 模型性能的影响，指导实际应用中的参数选择，我们对网络结构和优化器参数进行了敏感性分析。利用模拟生成的包含 800 个样本、25 维特征的数据集 simulated_data_3，数据分布具有一定的层次结构。

网络结构参数：调整 Siamese 神经网络(SNNs)的网络结构。分别设置隐藏层的层数为 2 层、3 层和 4 层，每层神经元的数量分别为 64、128 和 256。对不同网络结构设置下的 Trivis 模型在数据集 simulated_data_3 上进行降维实验，降维后的维度为 2 维，实验重复多次，评估指标采用 CTA、KNN 准确率和 SVM 准确率。实验结果如下表所示：

**Table 5.** Simulation dataset network structure parameter analysis indicators
**表 5.** 模拟数据集网络结构参数分析指标

| 隐藏层层数 | 每层神经元数量 | CTA | KNN 准确率 | SVM 准确率 |
|---|---|---|---|---|
| 2 | 64 | 0.700 ± 0.018 | 0.760 ± 0.012 | 0.720 ± 0.015 |
| 2 | 128 | 0.730 ± 0.016 | 0.790 ± 0.010 | 0.750 ± 0.013 |
| 2 | 256 | 0.740 ± 0.015 | 0.800 ± 0.009 | 0.760 ± 0.012 |
| 3 | 64 | 0.720 ± 0.017 | 0.770 ± 0.011 | 0.730 ± 0.014 |
| 3 | 128 | 0.760 ± 0.014 | 0.820 ± 0.008 | 0.780 ± 0.011 |
| 3 | 256 | 0.770 ± 0.013 | 0.830 ± 0.007 | 0.790 ± 0.010 |
| 4 | 64 | 0.710 ± 0.017 | 0.760 ± 0.011 | 0.720 ± 0.014 |
| 4 | 128 | 0.750 ± 0.015 | 0.810 ± 0.008 | 0.770 ± 0.012 |
| 4 | 256 | 0.760 ± 0.014 | 0.820 ± 0.007 | 0.780 ± 0.011 |

从表 5 可以看出，随着隐藏层层数和每层神经元数量的增加，模型性能总体有所提升，但当增加到一定程度后，性能提升幅度变小，且计算时间明显增加。在本实验中，3 层隐藏层、每层 128 个神经元的网络结构在性能和计算效率之间取得了较好的平衡。

优化器参数：针对使用的 Adam 优化器，改变其学习率(分别设置为 0.001、0.0001 和 0.00001)和动量(分别设置为 0.8、0.9 和 0.95)。在相同的数据集 simulated_data_3 上进行降维实验，降维后的维度为 2 维，实验重复 10 次，使用 CTA、KNN 准确率和 SVM 准确率评估模型性能。实验结果如下表所示：

**Table 6.** Simulated dataset optimizer parameter analysis metrics
**表 6.** 模拟数据集优化器参数分析指标

| 学习率 | 动量 | CTA | KNN 准确率 | SVM 准确率 |
|---|---|---|---|---|
| 0.001 | 0.8 | 0.700 ± 0.019 | 0.740 ± 0.014 | 0.700 ± 0.019 |
| 0.001 | 0.9 | 0.720 ± 0.018 | 0.760 ± 0.013 | 0.720 ± 0.018 |
| 0.001 | 0.95 | 0.730 ± 0.017 | 0.770 ± 0.012 | 0.730 ± 0.017 |
| 0.0001 | 0.8 | 0.710 ± 0.018 | 0.750 ± 0.013 | 0.710 ± 0.018 |
| 0.0001 | 0.9 | 0.740 ± 0.016 | 0.780 ± 0.012 | 0.740 ± 0.016 |
| 0.0001 | 0.95 | 0.750 ± 0.015 | 0.790 ± 0.011 | 0.750 ± 0.015 |
| 0.00001 | 0.8 | 0.680 ± 0.020 | 0.720 ± 0.015 | 0.670 ± 0.020 |
| 0.00001 | 0.9 | 0.700 ± 0.019 | 0.740 ± 0.014 | 0.700 ± 0.019 |
| 0.00001 | 0.95 | 0.710 ± 0.018 | 0.750 ± 0.013 | 0.710 ± 0.018 |

从表 6 可以看出，学习率和动量对模型性能有一定影响。学习率过大时，模型收敛速度快但容易错过最优解，导致性能下降；学习率过小时，模型收敛速度慢，训练时间长。动量在一定范围内增大，有助于加快模型收敛速度和提升性能。在本实验中，学习率为 0.0001、动量为 0.95 时，模型在保持数据结构方面表现较好。

综上所述，在实际应用中，应根据数据的规模、特征和计算资源等因素，合理选择 Trivis 模型的网络结构和优化器参数，以获得最佳的降维效果。

## 5. 结论

在数据挖掘中，经常需要将复杂的高维数据可视化到二维或三维空间中，因此降维和可视化仍然是数据处理和机器学习中的核心问题之一。基于三元组约束的参数化降维方法为实现高质量、可扩展和高效的数据嵌入开辟了新的路径。本文提出了一种新的基于三元组约束的参数化降维框架(Trivis)，旨在增强现有三元组约束降维方法在局部结构保持方面的能力。实验结果表明，Trivis 能够有效地在数据中捕捉全局和局部结构，提供了一种稳健的高维数据可视化解决方案。未来的工作将进一步改进该框架，并扩大其在不同数据环境中的应用和普适性。

## 参考文献

[1] Tenenbaum, J.B., Silva, V.d. and Langford, J.C. (2000) A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, **290**, 2319-2323. https://doi.org/10.1126/science.290.5500.2319

[2] Roweis, S.T. and Saul, L.K. (2000) Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, **290**, 2323-2326. https://doi.org/10.1126/science.290.5500.2323

[3] Shlens, J. (2014) A Tutorial on Principal Component Analysis.

[4] Balakrishnama, S. and Ganapathiraju, A. (1998) Linear Discriminant Analysis—A Brief Tutorial. Institute for Signal and Information Processing, 1-8.

[5] Van der Maaten, L. and Hinton, G. (2008) Visualizing Data Using t-SNE. *Journal of Machine Learning Research*, **9**, 2579-2605.

[6] Wang, Y., Huang, H., Rudin, C., Shaposhnik, Y., *et al*. (2021) Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMAP, and PaCMAP for Data Visualization. *Journal of Machine Learning Research*, **22**, 1-73.

[7] Belkina, A.C., Ciccolella, C.O., Anno, R., Halpert, R., Spidlen, J. and Snyder-Cappione, J.E. (2019) Automated Optimized Parameters for T-Distributed Stochastic Neighbor Embedding Improve Visualization and Analysis of Large Datasets. *Nature Communications*, **10**, Article No. 5415. https://doi.org/10.1038/s41467-019-13055-y

[8] McInnes, L., Healy, J., Melville, J., *et al*. (2018) UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.

[9] Ghojogh, B., Ghodsi, A., Karray, F., Crowley, M., *et al*. (2021) Uniform Manifold Approximation and Projection (UMAP) and Its Variants: Tutorial and Survey.

[10] Amid, E. and Warmuth, M.K. (2019) TriMap: Large-Scale Dimensionality Reduction Using Triplets.

[11] Wattenberg, M., Viégas, F. and Johnson, I. (2016) How to Use T-Sne Effectively. Distill. https://doi.org/10.23915/distill.00002

[12] Dorrity, M.W., Saunders, L.M., Queitsch, C., *et al.* (2020) Dimensionality Reduction by UMAP to Visualize Physical and Genetic Interactions. *Nature Communications*, **11**, Article No. 1537. https://doi.org/10.1038/s41467-020-15351-4

[13] Amid, E. and Warmuth, M.K. (2018) A More Globally Accurate Dimensionality Reduction Method Using Triplets.

[14] Kobak, D. and Linderman, G.C. (2021) Initialization Is Critical for Preserving Global Data Structure in Both t-SNE and UMAP. *Nature Biotechnology*, **39**, 156-157. https://doi.org/10.1038/s41587-020-00809-z

[15] Cao, H. and Wang, L. (2017) Advancing t-SNE's Efficiency with Distance Metric Learning. *Pattern Recognition Letters*, **94**, 62-67.

[16] Nguyen, L.H. and Holmes, S. (2019) Ten Quick Tips for Effective Dimensionality Reduction. *PLOS Computational Biology*, **15**, e1006907. https://doi.org/10.1371/journal.pcbi.100690 7

[17] Belkina, A.C. (2019) Automated Optimization of t-SNE. Bioinformatics.

[18] Van der Maaten, L. (2009) Learning a Parametric Embedding by Preserving Local Structure. *The* 12*th International Conference on. Artificial Intelligence and Statistics*, Clearwater Beach, 16-18 April 2009, 384-391.

[19] Sainburg, T., McInnes, L. and Gentner, T.Q. (2021) Parametric UMAP Embeddings for Representation and Semisupervised Learning. *Neural Computation*, **33**, 2881-2907. https://doi.org/10.1162/neco_a_01434

[20] Szubert, B., Cole, J.E., Monaco, C. and Drozdov, I. (2019) Structure-Preserving Visualisation of High Dimensional Single-Cell Datasets. *Scientific Reports*, **9**, Article No. 8914. https://doi.org/10.1038/s41598-019-45301-0

[21] Koch, G., Zemel, R. and Salakhutdinov, R. (2015) Siamese Neural Networks for One-Shot Image Recognition. In: *ICML Deep Learning Workshop*, Volume 2, 1-30.

[22] Chicco, D. (2020) Siamese Neural Networks: An Overview. In: Cartwright, H., Ed., *Artificial Neural Networks*, Springer, 73-94. https://doi.org/10.1007/978-1-0716-0826-5_3

[23] Zhang, Z.J. (2018) Improved Adam Optimizer for Deep Neural Networks. 2018 *IEEE/ACM* 26*th International Symposium on Quality of Service* (*IWQoS*), Banff, 4-6 June 2018, 1-2. https://doi.org/10.1109/IWQoS.2018.8624183

[24] Maas, A.L., Hannun, A.Y. and Ng, A.Y. (2013) Rectifier Nonlinearities Improve Neural Network Acoustic Models. In: *Proc. Icml*, Vol. 30, p. 3.

[25] Xu, B., Wang, N., Chen, T., Li, M., *et al*. (2015) Empirical Evaluation of Rectified Activations in Convolutional Network.

[26] Borg, I. and Groenen, P.J.F. (2007) Modern Multidimensional Scaling: Theory and Applications. Springer Science & Business Media.

[27] Belkin, M. and Niyogi, P. (2002) Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In: Dietterich, T.G., Becker, S. and Ghahramani, Z., Eds., *Advances in Neural Information Processing Systems* 14, The MIT Press, 585-592. https://doi.org/10.7551/mitpress/1120.003.0080

[28] Tang, J., Liu, J., Zhang, M. and Mei, Q. (2016) Visualizing Large-Scale and High-Dimensional Data. *Proceedings of the* 25*th International Conference on World Wide Web*, Montréal, 11-15 April 2016, 287-297. https://doi.org/10.1145/2872427.2883041

[29] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J. and Mei, Q. (2015) LINE: Large-Scale Information Network Embedding. *Proceedings of the* 24*th International Conference on World Wide Web*, Florence, 18-22 May 2015, 1067-1077. https://doi.org/10.1145/2736277.2741093

[30] Hinton, G.E. and Roweis, S. (2002) Stochastic Neighbor Embedding. *Proceedings of the* 16*th International Conference on Neural Information Processing Systems*, 1 January 2002, 857-864.

[31] Lai, C., Kuo, M., Lien, Y., Su, K. and Wang, Y. (2022) Parametric Dimension Reduction by Preserving Local Structure. 2022 *IEEE Visualization and Visual Analytics* (*VIS*), Oklahoma City, 16-21 October 2022, 75-79. https://doi.org/10.1109/vis54862.2022.00024

[32] Fischer, A. and Igel, C. (2014) Training Restricted Boltzmann Machines: An Introduction. *Pattern Recognition*, **47**, 25-39. https://doi.org/10.1016/j.patcog.2013.05.025

[33] van der Maaten, L. and Weinberger, K. (2012) Stochastic Triplet Embedding. 2012 *IEEE International Workshop on Machine Learning for Signal Processing*, Santander, 23-26 September 2012, 1-6.

[34] Wilber, M.J., Kwak, I.S., Kriegman, D. and Belongie, S. (2015) Learning Concept Embeddings with Combined Human-Machine Expertise. 2015 *IEEE International Conference on Computer Vision* (*ICCV*), Santiago, 7-13 December 2015, 981-989. https://doi.org/10.1109/iccv.2015.118

[35] Zelnik-Manor, L. and Perona, P. (2004) Self-Tuning Spectral Clustering. *Proceedings of the* 18*th International Conference on Neural Information Processing Systems*, Vancouver, 1 December 2004, 1601-1608.

[36] Hoffer, E. and Ailon, N. (2015) Deep Metric Learning Using Triplet Network. In: Feragen, A., Pelillo, M. and Loog, M., Eds., *Similarity-Based Pattern Recognition*, Springer International Publishing, 84-92. https://doi.org/10.1007/978-3-319-24261-3_7

[37] Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., Lecun, Y., Moore, C., *et al*. (1994) Signature Verification Using a "Siamese" Time Delay Neural Network. In: Guyon, I. and Wang, P.S.P., Eds., *Advances in Pattern Recognition Systems Using Neural Network Technologies*, World Scientific, 25-44. https://doi.org/10.1142/9789812797926_0003

[38] Hermans, A., Beyer, L. and Leibe, B. (2017) In Defense of the Triplet Loss for Person Re-Identification.

[39] Wang, B., Zhu, J., Pierson, E., Ramazzotti, D. and Batzoglou, S. (2017) Visualization and Analysis of Single-Cell RNA-seq Data by Kernel-Based Similarity Learning. *Nature Methods*, **14**, 414-416. https://doi.org/10.1038/nmeth.4207