


Summarizing Data Structures with Gaussian Process and Robust Neighborhood Preservation

Koshi Watanabe¹, Keisuke Maeda², Takahiro Ogawa², Miki Haseyama ²

¹ Graduate School of Information Science and Technology, Hokkaido University,
Japan

² Faculty of Information Science and Technology, Hokkaido University, Japan
`{koshi,maeda,ogawa,mhaseyama}@lmd.ist.hokudai.ac.jp`

Abstract. Latent variable models summarize high-dimensional data while preserving its many complex properties. This paper proposes a locality-aware and low-rank approximated Gaussian process latent variable model (LolaGP) that can preserve the global relationship and local geometry in the derivation of the latent variables. We realize the global relationship by imitating the sample similarity non-linearly and the local geometry based on our newly constructed neighborhood graph. Formally, we derive LolaGP from GP-LVM and implement a locality-aware regularization to reflect its adjacency relationship. The neighborhood graph is constructed based on the latent variables, making the local preservation more resistant to noise disruption and the curse of dimensionality than the previous methods that directly construct it from the high-dimensional data. Furthermore, we introduce a new lower bound of a log-posterior distribution based on low-rank matrix approximation, which allows LolaGP to handle larger datasets than the conventional GP-LVM extensions. Our contribution is to preserve both the global and local structures in the derivation of the latent variables using the robust neighborhood graph and introduce the scalable lower bound of the log-posterior distribution. We conducted an experimental analysis using synthetic as well as images with and without highly noise disrupted datasets. From both qualitative and quantitative standpoint, our method produced successful results in all experimental settings.

Keywords: Latent Variable Model · Gaussian Processes · Neighborhood Graph · Diffusion Map.

1 Introduction

Real-world data exist in a high-dimensional data space while including a low-dimensional manifold. This statement is consistent with the manifold hypothesis [5], and determining the meaningful structure is a general task in machine learning. *Dimensionality reduction* [42] is one of the basic approaches to explore the meaningful structure and estimate the low dimensional manifold, which preserves several properties of the original high dimensional data. The obtained

low dimensional representation, particularly two or three-dimensional representation, is frequently useful in interpreting and visualizing complex high dimensional data.

There are numerous approaches to dimensionality reduction. Principal component analysis (PCA) [19, 35] is one of the representative approaches to deriving a linear mapping into a low-dimensional subspace. Specifically, PCA selects the dominant components of a sample covariance matrix and treats them as new axes of the low-dimensional subspace. Based on this mapping, we can obtain the low-dimensional representation while preserving the global relationship of the high-dimensional data. However, in general, since this mapping-based approach does not take into account the desired subspace, we cannot obtain an optimal representation for its dimensionality. Probabilistic PCA (PPCA) [37] approaches this problem by explicitly assuming the low-dimensional representation as *latent variables* and learning them linearly by imitating the sample similarity. Gaussian process latent variable model (GP-LVM) [22] incorporates the kernel method into PPCA and non-linearly imitates the similarity. Those conventional latent variable models, on the other hand, still have some limitations, such as the *explainability* [6, 18] of the latent variables, the *interpretability* [2, 25, 33] of the locality within the high-dimensional data, and the *scalability* [27, 34] to the sample size.

Previous latent variable models can overcome the limitations of these methods. β -variational autoencoder (β -VAE) [6, 18, 20] tackles the explainability limitation of the latent variables and extracts them as a factor of variation. Specifically, β -VAE attempts to map one latent feature to one variation factor hidden in the entire dataset (e.g., rotation or shrinking scale). By the *disentangled* representation, β -VAE can generate artificial data where only a single factor has changed [48]. While β -VAE is useful for global analysis of high-dimensional data, it sacrifices *local geometry* within the high-dimensional data. Locally linear embedding (LLE) [7, 33] is one of the representative approaches for incorporating local geometry into latent variables. LLE employs a two-step learning process, constructing a neighborhood graph and embedding it into the latent space. LLE can compress high-dimensional data while maintaining high interpretability of their local geometry by embedding local information based on the neighborhood graph. Uniform manifold approximation and projection (UMAP) [27] is a state-of-the-art method in this graph embedding method that addresses the scalability issue. UMAP can reduce the dimensionality of large-scale data reflecting their local manifold (e.g., class separation) and has received a lot of attention for real-data analysis due to its high scalability and visibility, such as genetic analysis [3], human population analysis [12], and social network analysis [31]. These embedding methods, however, have a limitation in their graph construction. They build the neighborhood graph directly from the high-dimensional data, which may result in an undesirable graph construction due to noise disruption [7] or the curse of dimensionality [41].

In this paper, we learn the local geometry under noise disruption and the curse of dimensionality while considering the global relationship via *locality-*

aware and low-rank approximated GP-LVM (LolaGP). LolaGP constructs the neighborhood graph with the low-dimensional latent variables using an iterative learning strategy. Then, LolaGP introduces a new locality-aware regularization into GP-LVM, forcing the latent variables to move closer if they are adjacent on the neighborhood graph. As a result, we can preserve the local geometry more precisely despite noise disruption and the curse of dimensionality. Furthermore, the latent representation in LolaGP holds the property of GP-LVM, which implies LolaGP can compress the high-dimensional data while considering both the global and local structures.

However, Gaussian process-based methods require a matrix inversion for each training step, and their scalability for even thousands of data points is a concern [17, 32]. Bayesian GP-LVM [11, 39] solves the scalability problem and produces latent variables in a fully Bayesian manner. While Bayesian GP-LVM improves scalability more than previous methods, its scalability and even its closed-form expression collapse with a complex regularization [11]. For this reason, we newly derive a lower bound of a log-posterior distribution based on a combination of the previous research [21, 38], and this bound allows for scalable optimization using complex regularization. In summary, LolaGP has the following contributions:

- We construct the neighborhood graph with the low-dimensional latent variables, making the graph construction more robust to the noise disruption and the curse of dimensionality. Furthermore, We can also reflect the global relationship and local geometry into the latent variables by using the robust adjacency relation with GP-LVM.
- We introduce the lower bound of the log-posterior distribution. This bound enables scalable optimization while considering the complex regularization, which is difficult with the previous GP-LVM extensions.

We conducted an experimental analysis in which we visualized and quantified the derived latent variables using one synthetic dataset and two image datasets with and without noise disruptions. In this experiment, we demonstrated that our proposed method could embed high-dimensional data with high interpretability of their local geometry while maintaining the global relationship.

2 Background

In this section, we introduce the previous related methods to clarify the positioning of our proposed method. In 2.1, we briefly describe the graph embedding methods and discuss their limitations of the learning procedure with a synthetic dataset. In 2.2, we explain GP-LVM and its extensions and discuss why they cannot handle thousands of data points.

2.1 Graph Embedding Methods

LLE is a representative approach to reducing dimensionality while preserving local geometry, and it is closely related to Manifold Learning or Topological

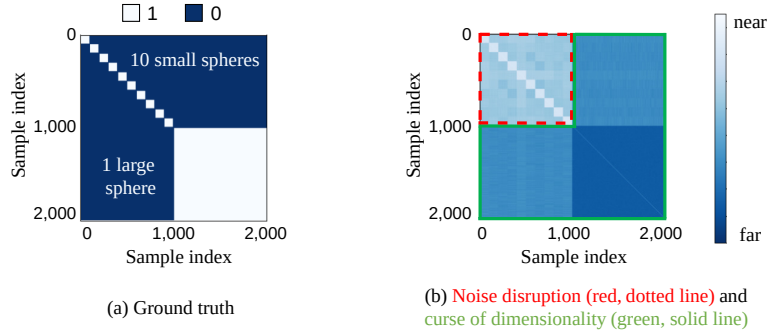


Fig. 1: An example of a neighborhood graph with a synthetic dataset, Spheres. (a) Ground truth that completely separates each sphere in 101-dimensional space (‘1’ means an edge exists, and ‘0’ means an edge does not), and (b) an example of a weighted adjacency matrix computed by the Square Exponential kernel.

Data Analysis [46]. LLE and its related methods [2, 4, 15, 27] usually construct a neighborhood graph of high-dimensional data to preserve the local geometry while calculating the low dimensional representation based on its adjacency relation. In the language of *topology*, shapes of a point set determined by a graph are *Vietoris-Rips complexes*, and they can recover various manifolds on which the point set is actually distributed [1, 14]. The Vietoris-Rips complexes’ properties are supported by a solid theoretical foundation, which motivates the effectiveness of the graph-based approaches. Although they produce successful results even in the application settings [47], they have several concerns during the graph construction phase. We demonstrate them using the Spheres dataset, [28] (Fig. 1), which contains ten small spheres and one large sphere surrounding them, and the spheres exist in the 101-dimensional space. Ten small spheres have 100 data points with white Gaussian noise, and one large sphere has as many data points as all small spheres (i.e., 1,000 data points). Under this condition, the most desirable adjacency matrix should separate each sphere, as shown in Fig. 1 (a). However, in Fig. 1 (b), the non-diagonal blocks of the small spheres’ entries have small values, and the separation of the small spheres is disturbed by the additive Gaussian noise. Furthermore, the distance between the small spheres’ points and the large sphere’s points is smaller than the distance within the large sphere’s points. This implies that we cannot realize the large sphere from the adjacency matrix because of the curse of dimensionality. LolaGP solves these problems by simultaneously learning the low-dimensional latent variables, and the neighborhood graph is constructed based on the low-dimensional representation. This learning strategy can precisely calculate the local geometry-aware representation under the noise disruption and dimensionality problem.

2.2 Gaussian Process Latent Variable Model (GP-LVM)

Let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N]^\top \in \mathbb{R}^{N \times D}$ be D dimensional (i.e., high-dimensional) data containing N data points. GP-LVM aims to compress these observed variables into low-dimensional latent variables $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times Q}$ ($D \gg Q$) and assumes a generative process from the latent space to the observed space as

$$\mathbf{y}_{:,d} = \mathbf{f}(\mathbf{X}) + \boldsymbol{\epsilon}, \mathbf{f}(\mathbf{X}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{NN}^{(\mathbf{f})}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \beta^{-1}\mathbf{I}), \quad (1)$$

where $\mathbf{y}_{:,d}$ is $d (= 1, 2, \dots, D)$ -th column vector of \mathbf{Y} , $\boldsymbol{\epsilon} \in \mathbb{R}^N$ is a Gaussian noise with a precision β , and $\mathbf{f}(\mathbf{X}) \triangleq \mathbf{f} \in \mathbb{R}^N$ is a Gaussian process prior of the generative process with a covariance matrix $\mathbf{K}_{NN}^{(\mathbf{f})} \in \mathbb{R}^{N \times N}$. Each entry of the matrix $\mathbf{K}_{NN}^{(\mathbf{f})}$ is calculated by a positive definite kernel $k^{(\mathbf{f})}(\mathbf{x}, \mathbf{x}')$. Equation (1) can be rewritten as $p(\mathbf{y}_{:,d}|\mathbf{f}) = \mathcal{N}(\mathbf{y}_{:,d}|\mathbf{f}, \beta^{-1}\mathbf{I})$ and $p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_{NN}^{(\mathbf{f})})$, and we can derive a likelihood function by marginalizing the Gaussian process prior \mathbf{f} as

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{d=1}^D \mathcal{N}(\mathbf{y}_{:,d}|\mathbf{0}, \mathbf{K}_{NN}^{(\mathbf{f})} + \beta^{-1}\mathbf{I}). \quad (2)$$

In vanilla GP-LVM, the derivation of the latent variables is performed by maximizing the log-likelihood with respect to \mathbf{X} , and this is the same as imitating the sample similarity matrix $\mathbf{Y}\mathbf{Y}^\top$ into the latent precision matrix $(\mathbf{K}_{NN}^{(\mathbf{f})} + \beta^{-1}\mathbf{I})^{-1}$. By this learning strategy, we can preserve the data structure globally in the derivation of the low dimensional latent variables \mathbf{X} . The extensions of GP-LVM typically introduce a prior distribution $p(\mathbf{X})$ as a regularization to reflect several properties into the latent variables [13, 23, 36, 40, 41, 45]. By this expansion, the maximum likelihood estimation of the latent variables \mathbf{X} is replaced by *maximum a posteriori* (MAP) estimation, and a log-posterior distribution is shown as follows:

$$\log p(\mathbf{X}|\mathbf{Y}) = \log p(\mathbf{Y}|\mathbf{X}) + \log p(\mathbf{X}) + C, \quad (3)$$

where $C = -\log p(\mathbf{Y})$ is a log-normalized constant of the distribution $p(\mathbf{X}|\mathbf{Y})$ and usually ignored. Although those extensions perform well when embedding high-dimensional data, they fail to handle thousands of data points because of the matrix inversion of the $N \times N$ matrix $\mathbf{K}^{(\mathbf{f})} + \beta^{-1}\mathbf{I}$. Bayesian GP-LVM [11, 39] introduces low-rank matrix approximation to perform fully Bayesian optimization and can handle large datasets as side effects. Bayesian GP-LVM is an efficient method for obtaining the low-dimensional representation. However, with a complex prior distribution, such as the GP-LVM extensions, its scalability and even closed-form expression easily collapse [11]. From the above, we newly derive a scalable lower bound for MAP estimation, and this bound enables us to handle large datasets with a complex prior distribution.

3 Locality-aware and Low-rank Approximated GP-LVM

LolaGP employs an iterative learning process to derive the latent variables and construct the neighborhood graph using a low-dimensional representation. In 3.1, we present a locality-aware regularization based on graph Gaussian process [30] and show how to derive the scalable lower bound in 3.2. In 3.3, we describe how to build a neighborhood graph for efficient local preservation using latent variables [25]. Our learning strategy allows for the preservation of both global and local structures and the calculation of latent variables on a scalable basis.

3.1 Locality-aware Regularization

Similar to the previous graph-based methods presented in 2.1, we preserve the local geometry using the neighborhood graph and its weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$. For efficient reflection of the local geometry, we assume that the latent variables are weighted averages of their neighbors based on graph Gaussian process [30] by the following equations:

$$\mathbf{x}_{:,q} = \mathbf{g}(\mathbf{X}) + \boldsymbol{\eta}, \quad (4)$$

$$g_n = \frac{\sum_{i=1}^N W_{ni} h_i}{D_n} \quad (n = 1, 2, \dots, N), \quad (5)$$

where

$$\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \gamma^{-1} \mathbf{I}), \quad \mathbf{h}(\mathbf{X}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{NN}^{(\mathbf{h})}),$$

$\mathbf{x}_{:,q}$ is $q (= 1, 2, \dots, Q)$ -th column vector of \mathbf{X} , W_{ni} is an entry of the weighted adjacency matrix \mathbf{W} , and $\mathbf{D} \in \mathbb{R}^{N \times N}$ with a diagonal entry D_n is a degree matrix of \mathbf{W} . Furthermore, $\mathbf{g}(\mathbf{X}) = [g_1, g_2, \dots, g_N]^\top \in \mathbb{R}^N$ is a graph Gaussian process prior, $\mathbf{h}(\mathbf{X}) = [h_1, h_2, \dots, h_N]^\top \in \mathbb{R}^N$ is a Gaussian process prior with a covariance matrix $\mathbf{K}_{NN}^{(\mathbf{h})}$ with a kernel function $k^{(\mathbf{h})}(\mathbf{x}, \mathbf{x}')$, and $\boldsymbol{\eta}$ is a Gaussian noise with a precision γ . We show how to calculate \mathbf{W} in 3.3. By the property of Gaussian distribution, we can rewrite Eqs. (4) and (5) as the following probability distributions:

$$p(\mathbf{x}_{:,q} | \mathbf{g}) = \mathcal{N}(\mathbf{x}_{:,q} | \mathbf{g}, \gamma^{-1} \mathbf{I}), \quad (6)$$

$$\begin{aligned} p(\mathbf{g}) &= \mathcal{N}(\mathbf{g} | \mathbf{0}, \mathbf{P} \mathbf{K}_{NN}^{(\mathbf{h})} \mathbf{P}^\top) \\ &\triangleq \mathcal{N}(\mathbf{g} | \mathbf{0}, \mathbf{K}_{NN}^{(\mathbf{g})}), \end{aligned} \quad (7)$$

where $\mathbf{P} = \mathbf{D}^{-1} \mathbf{W}$ is a normalized adjacency matrix of \mathbf{W} . Note that we can set up the matrix \mathbf{P} as $\mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$ by simple modification of Eq. (5). By marginalizing \mathbf{g} , we derive the locality-aware prior distribution $p(\mathbf{X})$ as follows:

$$p(\mathbf{X}) = \prod_{q=1}^Q \mathcal{N}(\mathbf{x}_{:,q} | \mathbf{0}, \mathbf{K}_{NN}^{(\mathbf{g})} + \gamma^{-1} \mathbf{I}). \quad (8)$$

By this prior distribution, we can reflect the local geometry represented by the weighted adjacency matrix \mathbf{W} into the latent variables, and this graph Gaussian process-based formulation helps to derive a scalable lower bound. We optimize the latent variables \mathbf{X} by maximizing the posterior distribution $p(\mathbf{X}|\mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X})p(\mathbf{X})$.

3.2 Lower Bound of Log-posterior Distribution

In this subsection, we demonstrate the objective function of LolaGP. We first describe an **exact** log-posterior distribution as follows:

$$\begin{aligned} \log p(\mathbf{X}|\mathbf{Y}) &= -\frac{ND}{2} \log(2\pi) - \frac{D}{2} \log |\mathbf{K}_{NN}^{(\mathbf{f})} + \beta^{-1}\mathbf{I}| - \frac{1}{2} \text{tr} \left[(\mathbf{K}_{NN}^{(\mathbf{f})} + \beta^{-1}\mathbf{I})^{-1} \mathbf{Y}\mathbf{Y}^\top \right] \\ &\quad - \frac{NQ}{2} \log(2\pi) - \frac{Q}{2} \log |\mathbf{K}_{NN}^{(\mathbf{g})} + \gamma^{-1}\mathbf{I}| - \frac{1}{2} \text{tr} \left[(\mathbf{K}_{NN}^{(\mathbf{g})} + \gamma^{-1}\mathbf{I})^{-1} \mathbf{X}\mathbf{X}^\top \right]. \end{aligned} \quad (9)$$

This exact log-posterior is not a scalable objective function because its evaluation requires $O(N^3)$ time complexity for the matrix inversion of the $N \times N$ matrices $\mathbf{K}_{NN}^{(\mathbf{f})} + \beta^{-1}\mathbf{I}$ and $\mathbf{K}_{NN}^{(\mathbf{g})} + \gamma^{-1}\mathbf{I}$. From the above, we introduce a newly scalable lower bound based on low-rank approximation of these matrices. Fortunately, both the likelihood in Eq. (2) and the prior distribution in Eq. (8) are Gaussian distributed with the marginalized Gaussian process priors $\mathbf{f} \in \mathbb{R}^N$ and $\mathbf{g} \in \mathbb{R}^N$, and we can select M inducing points $\mathbf{u} \in \mathbb{R}^M$ and $\mathbf{v} \in \mathbb{R}^M$ from the Gaussian process priors with same latent positions $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_M]^\top \in \mathbb{R}^{M \times Q}$. The joint probabilities $p(\mathbf{f}, \mathbf{u})$ and $p(\mathbf{g}, \mathbf{v})$ are also Gaussian distributed, and we can write the probability distributions of \mathbf{f} and \mathbf{g} respectively conditioned by \mathbf{u} and \mathbf{v} and the marginal distributions of the inducing points by the following equations:

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{f} | \mathbf{K}_{NM}^{(\mathbf{f})} \mathbf{K}_{MM}^{(\mathbf{f})^{-1}} \mathbf{u}, \mathbf{K}_{NN}^{(\mathbf{f})} - \mathbf{K}_{NM}^{(\mathbf{f})} \mathbf{K}_{MM}^{(\mathbf{f})^{-1}} \mathbf{K}_{MN}^{(\mathbf{f})}), \quad (10)$$

$$p(\mathbf{g}|\mathbf{v}) = \mathcal{N}(\mathbf{g} | \mathbf{K}_{NM}^{(\mathbf{g})} \mathbf{K}_{MM}^{(\mathbf{g})^{-1}} \mathbf{v}, \mathbf{K}_{NN}^{(\mathbf{g})} - \mathbf{K}_{NM}^{(\mathbf{g})} \mathbf{K}_{MM}^{(\mathbf{g})^{-1}} \mathbf{K}_{MN}^{(\mathbf{g})}), \quad (11)$$

$$p(\mathbf{u}|\mathbf{Z}) = \mathcal{N}(\mathbf{u} | \mathbf{0}, \mathbf{K}_{MM}^{(\mathbf{f})}), \quad p(\mathbf{v}|\mathbf{Z}) = \mathcal{N}(\mathbf{v} | \mathbf{0}, \mathbf{K}_{MM}^{(\mathbf{g})}), \quad (12)$$

where $\mathbf{K}_{NM}^{(\cdot)} \in \mathbb{R}^{N \times M}$ and $\mathbf{K}_{MN}^{(\cdot)} \in \mathbb{R}^{M \times N}$ ($\cdot = \mathbf{f}, \mathbf{g}$) are covariance matrices between the latent variables \mathbf{X} and the positions \mathbf{Z} , and $\mathbf{K}_{MM}^{(\cdot)} \in \mathbb{R}^{M \times M}$ is a covariance matrix of \mathbf{Z} . We define $\mathbf{Q}_{NN}^{(\cdot)} = \mathbf{K}_{NM}^{(\cdot)} \mathbf{K}_{MM}^{(\cdot)^{-1}} \mathbf{K}_{MN}^{(\cdot)}$ to simplify the notation, and they can be regarded as the low-rank (i.e., Nyström) approximation of the full matrix $\mathbf{K}_{NN}^{(\cdot)}$.

By those probability distributions, we introduce a lower bound of the log-posterior distribution based on a combination of the previous research [21, 38]. We explicitly marginalize the inducing points \mathbf{u} and \mathbf{v} as

$$\begin{aligned} \log p(\mathbf{X}|\mathbf{Y}) &= \sum_{d=1}^D \log \int p(\mathbf{y}_{:,d}|\mathbf{u}) p(\mathbf{u}|\mathbf{Z}) d\mathbf{u} + \sum_{q=1}^Q \log \int p(\mathbf{x}_{:,q}|\mathbf{v}) p(\mathbf{v}|\mathbf{Z}) d\mathbf{v}. \end{aligned} \quad (13)$$

Next, we evaluate two likelihood functions $p(\mathbf{y}_{:,d}|\mathbf{u})$ and $p(\mathbf{x}_{:,q}|\mathbf{v})$ by the following Jensen's inequality [38]:

$$\log p(\mathbf{y}_{:,d}|\mathbf{u}) \geq \mathbb{E}_{p(\mathbf{f}|\mathbf{u})} [\log p(\mathbf{y}_{:,d}|\mathbf{f})], \quad (14)$$

$$\log p(\mathbf{x}_{:,q}|\mathbf{v}) \geq \mathbb{E}_{p(\mathbf{g}|\mathbf{v})} [\log p(\mathbf{x}_{:,q}|\mathbf{g})]. \quad (15)$$

By substituting Eqs. (14) and (15) into Eq. (13), we can derive the following scalable lower bound of the log-posterior distribution:

$$\begin{aligned} & \log p(\mathbf{X}|\mathbf{Y}) \\ & \geq \sum_{d=1}^D \left\{ \log \mathcal{N}(\mathbf{y}_{:,d}|\mathbf{0}, \mathbf{Q}_{NN}^{(\mathbf{f})} + \beta^{-1}\mathbf{I}) - \frac{\beta}{2} \text{tr}(\mathbf{K}_{NN}^{(\mathbf{f})} - \mathbf{Q}_{NN}^{(\mathbf{f})}) \right\} \\ & \quad + \sum_{q=1}^Q \left\{ \log \mathcal{N}(\mathbf{x}_{:,q}|\mathbf{0}, \mathbf{Q}_{NN}^{(\mathbf{g})} + \gamma^{-1}\mathbf{I}) - \frac{\gamma}{2} \text{tr}(\mathbf{K}_{NN}^{(\mathbf{g})} - \mathbf{Q}_{NN}^{(\mathbf{g})}) \right\}, \quad (16) \end{aligned}$$

and we can also expand each summation as

$$\begin{aligned} & \sum_{d=1}^D \left\{ \log \mathcal{N}(\mathbf{y}_{:,d}|\mathbf{0}, \mathbf{Q}_{NN}^{(\mathbf{f})} + \beta^{-1}\mathbf{I}) - \frac{\beta}{2} \text{tr}(\mathbf{K}_{NN}^{(\mathbf{f})} - \mathbf{Q}_{NN}^{(\mathbf{f})}) \right\} \\ & = -\frac{ND}{2} \log(2\pi) - \frac{D}{2} \log |\mathbf{Q}_{NN}^{(\mathbf{f})} + \beta^{-1}\mathbf{I}| \\ & \quad - \frac{1}{2} \text{tr} \left[(\mathbf{Q}_{NN}^{(\mathbf{f})} + \beta^{-1}\mathbf{I})^{-1} \mathbf{Y} \mathbf{Y}^\top \right] - \frac{\beta D}{2} \text{tr}(\mathbf{K}_{NN}^{(\mathbf{f})} - \mathbf{Q}_{NN}^{(\mathbf{f})}), \\ & \sum_{q=1}^Q \left\{ \log \mathcal{N}(\mathbf{x}_{:,q}|\mathbf{0}, \mathbf{Q}_{NN}^{(\mathbf{g})} + \gamma^{-1}\mathbf{I}) - \frac{\gamma}{2} \text{tr}(\mathbf{K}_{NN}^{(\mathbf{g})} - \mathbf{Q}_{NN}^{(\mathbf{g})}) \right\} \\ & = -\frac{NQ}{2} \log(2\pi) - \frac{Q}{2} \log |\mathbf{Q}_{NN}^{(\mathbf{g})} + \gamma^{-1}\mathbf{I}| \\ & \quad - \frac{1}{2} \text{tr} \left[(\mathbf{Q}_{NN}^{(\mathbf{g})} + \gamma^{-1}\mathbf{I})^{-1} \mathbf{X} \mathbf{X}^\top \right] - \frac{\gamma Q}{2} \text{tr}(\mathbf{K}_{NN}^{(\mathbf{g})} - \mathbf{Q}_{NN}^{(\mathbf{g})}). \end{aligned}$$

Comparing Eqs. (9) and (16), the full covariance matrix $\mathbf{K}_{NN}^{(\cdot)}$ is replaced by the low-rank covariance matrix $\mathbf{Q}_{NN}^{(\cdot)}$ with the additional trace term $\text{tr}(\mathbf{K}_{NN}^{(\cdot)} - \mathbf{Q}_{NN}^{(\cdot)})$. By this replacement, we can avoid computing the $N \times N$ matrix inversion and can reduce the time complexity $O(N^3)$ to $O(NM^2)$ by applying the Woodbury matrix identity to Eq. (16).

3.3 Construction of Neighborhood Graph

We need the neighborhood graph to reflect the local geometry in the latent variables to achieve our objective. Intuitively, The Euclidean distance is appropriate for calculating the adjacency matrix, but this metric cannot precisely realize the local geometry. We visualize this in Fig. 2. From Fig. 2 (a), The distribution of

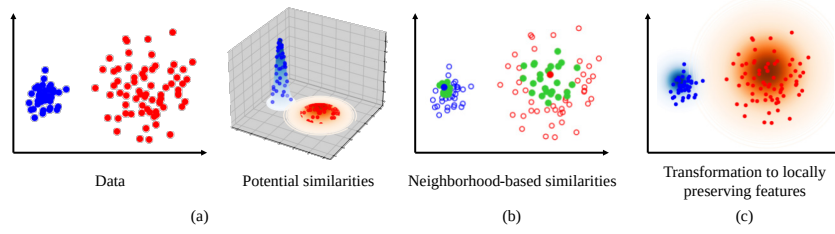


Fig. 2: Transformation of the latent variables into local preserving features. We show (a) data with two different density areas and different potential similarities, (b) neighborhood-based similarities to realize the local geometry and (c) a diffusion process to transform the latent variables into the locally preserving features.

data can be divided into two areas: dense and sparse, and the potential similarities between them are influenced by their local densities. However, we cannot realize the non-linear geometry by the Euclidean distance since it is a simple straight line in the Euclidean space. To overcome this difficulty, we peculiarly focus on *diffusion map* [10, 25], which captures the local geometry by propagating neighborhood similarities by the diffusion process, i.e., powering a random walk matrix. In LolaGP, we transform the latent variables into locally preserving features based on the diffusion process and calculate the weighted adjacency matrix using them.

We first calculate a similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ of the latent variables \mathbf{X} that realizes the local geometry based on the α -decay kernel [25] as

$$k_\alpha(\mathbf{x}, \mathbf{x}') = \frac{1}{2} \exp \left[- \left(\frac{\|\mathbf{x} - \mathbf{x}'\|_2}{\epsilon_k(\mathbf{x})} \right)^\alpha \right] + \frac{1}{2} \exp \left[- \left(\frac{\|\mathbf{x} - \mathbf{x}'\|_2}{\epsilon_k(\mathbf{x}')} \right)^\alpha \right], \quad (17)$$

where α is a hyperparameter that controls the decay rate of each exponential value, and $\epsilon_k(\mathbf{x}_*)$ is the Euclidean distance between \mathbf{x}_* and its k -nearest neighbor. The position of $\epsilon_k(\mathbf{x}_*)$ is the same as the *lengthscale* parameter of the Squared Exponential (SE) kernel, and we can reflect the neighborhood geometry of \mathbf{x}_* by setting k to appropriate values (Fig. 2 (b)). We calculate a random walk matrix of \mathbf{S} as $\mathbf{R} = \mathbf{D}_\mathbf{S}^{-1} \mathbf{S}$ ($\mathbf{D}_\mathbf{S}$ is the degree matrix of \mathbf{S}) and derive features after t -step diffusion of the neighborhood-based similarities (Fig. 2 (c)) by powering \mathbf{R} as

$$\mathbf{U}^{(t)} = \mathbf{R}^t, \mathbf{U}^{(t)} = [\mathbf{u}_1^{(t)}, \mathbf{u}_2^{(t)}, \dots, \mathbf{u}_N^{(t)}]^\top \in \mathbb{R}^{N \times N}. \quad (18)$$

Each row vector $\mathbf{u}_n^{(t)} \in \mathbb{R}^N$ indicates the strength of the interconnection between sample n and the other samples after the t -step diffusion, and it is reasonable to state that nearby samples on the local manifold have similar vector values. We regard $\mathbf{U}^{(t)}$ as the locally preserving features of \mathbf{X} and calculate the weighted adjacency matrix \mathbf{W} based on them. Since the vector $\mathbf{u}_n^{(t)}$ is the probability

value, we take the logarithm of it for efficient computation. Using the information above, we can calculate each entry of \mathbf{W} using the following equation:

$$W_{ij} = \exp \left(-\|\log \mathbf{u}_i^{(t)} - \log \mathbf{u}_j^{(t)}\|_2 \right). \quad (19)$$

The weighted matrix \mathbf{W} appears in the gram matrices $\mathbf{K}_{NN}^{(\mathbf{g})}$ and $\mathbf{Q}_{NN}^{(\mathbf{g})}$ in Eq. (16) as its normalized form \mathbf{P} , and we calculate the normalized matrix with respect to \mathbf{X} and \mathbf{Z} as $\mathbf{P}_\mathbf{X}$ and $\mathbf{P}_\mathbf{Z}$, respectively. Finally, we calculate each gram matrix as

$$\begin{aligned} \mathbf{K}_{NM}^{(\mathbf{g})} &= \mathbf{P}_\mathbf{X} \mathbf{K}_{NM}^{(\mathbf{h})} \mathbf{P}_\mathbf{Z}^\top, \\ \mathbf{K}_{MM}^{(\mathbf{g})} &= \mathbf{P}_\mathbf{Z} \mathbf{K}_{MM}^{(\mathbf{h})} \mathbf{P}_\mathbf{Z}^\top, \\ \mathbf{K}_{MN}^{(\mathbf{g})} &= \mathbf{P}_\mathbf{Z} \mathbf{K}_{MN}^{(\mathbf{h})} \mathbf{P}_\mathbf{X}^\top. \end{aligned}$$

These matrices allow the latent variables to realize the non-linear geometry.

To summarize, we extend GP-LVM by introducing locality-aware regularization via latent variables with the newly constructed neighborhood graph to avoid the influence of noise disruption or the curse of dimensionality. We also use sparse Gaussian process methods to derive the lower bound of the exact log-posterior distribution. Our method can embed high-dimensional data with high scalability and interpretability of both global and local structures, thanks to this innovation.

4 Experiments

In this section, we conduct an experimental analysis to validate the efficacy of our method. We assessed LolaGP and other comparative methods in both qualitative and quantitative ways, viz., by visualizing latent variables and computing quality metrics. We used the Gaussian process open library GPy [16] on an Intel Core i7-10700 CPU to implement our source code.

4.1 Experimental Settings

Datasets. We used two image datasets, COIL20 [29] and MNIST³ and one synthetic dataset, Spheres [28], described in 2.1. COIL20 contains 1,440 grayscale images of 20 objects, and each object is captured evenly in a single rotation across 72 images. We selected five objects (indexed as {1, 4, 6, 11, 13}) from COIL20 dataset. Furthermore, in order to get closer to our problem setting, we randomly lost pixels from the selected images with a probability of 35% and regarded them as *Noisy* COIL20 dataset referring to [24] (see Fig. 3). Both COIL20 and *Noisy* COIL20 include the global relationship based on the object separation and the local geometry according to the object rotation. In MNIST,

³ <http://yann.lecun.com/exdb/mnist/>

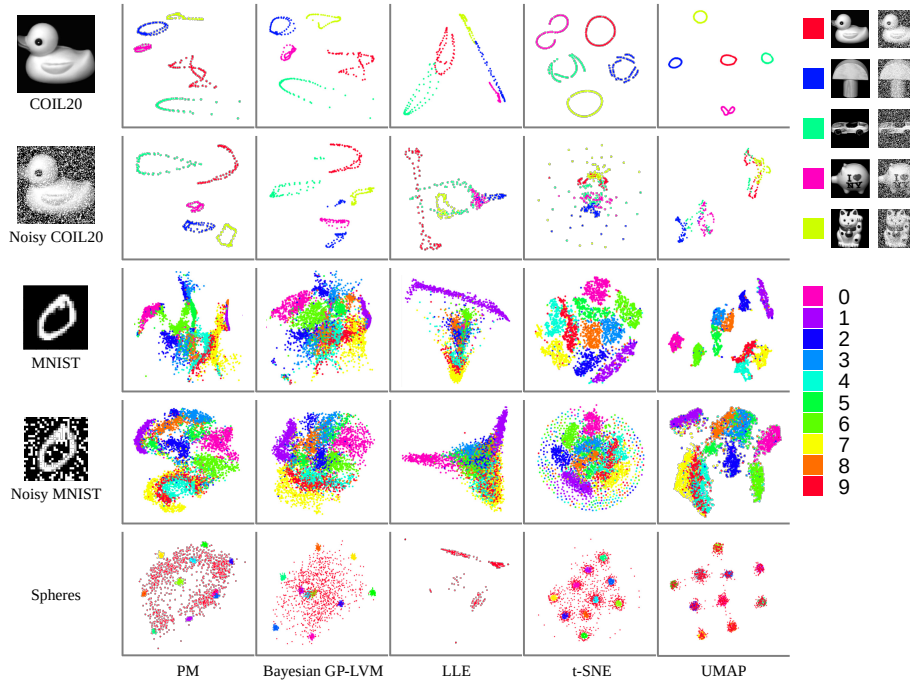


Fig. 3: Visualization results of the five datasets (COIL20, *Noisy* COIL20, MNIST, *Noisy* MNIST, and Spheres). Each color shows the separable class in the dataset, i.e., each object (COIL20 and *Noisy* COIL20), each digit (MNIST and *Noisy* MNIST), and each sphere (Spheres). We ignored some outliers for efficient visualization.

we randomly selected 5,000 images from the training set and also build *Noisy* MNIST by randomly losing their pixels with a probability of 25% which is the limit to remain the characteristics of each digit.

Comparative Methods. We compared our proposed method (PM) with LLE [33] and Bayesian GP-LVM [11] as benchmarks described in section 2. Furthermore, we took the commonly used method, t-SNE [43], and further adopted UMAP [27] as the state-of-the-art method in the same manner as the previous research [28]. All methods were compared from both qualitative and quantitative perspectives after embedding the observed high-dimensional data into two-dimensional latent space.

Training Procedure. The trainable parameters in our proposed method are the latent variables \mathbf{X} and the locations of the inducing points \mathbf{Z} . We initialized them by PCA [19] and by randomly picking up from the initialized \mathbf{X} , respectively. Furthermore, we need the normalized adjacency matrix \mathbf{P} in 3.3 and initialized it by computing \mathbf{P} from the observed variables \mathbf{Y} . Under this initialization, we iterated our method two times and changed \mathbf{P} following the latent variables \mathbf{X}

in the middle iteration. We selected the ‘SE+linear+whitenoise’ kernel with the automatic relevance determination (ARD) [26] as the kernel functions $k^{(f)}(\mathbf{x}, \mathbf{x}')$ and $k^{(h)}(\mathbf{x}, \mathbf{x}')$:

$$k^{(\cdot)}(\mathbf{x}, \mathbf{x}') = \sigma_{\text{SE}}^2 \exp \left[-\frac{1}{2} \sum_{q=1}^Q a_{\text{SE},q} (x_q - x'_q)^2 \right] + \sigma_{\text{lin.}}^2 \sum_{q=1}^Q a_{\text{lin.},q} x_q x'_q + \sigma_{\text{noise}}^2 \delta_{\mathbf{x}, \mathbf{x}'}, \quad (20)$$

where $\boldsymbol{\theta} = \{\sigma_{\{\text{SE}, \text{lin.}, \text{noise}\}}, \{a_{\{\text{SE}, \text{lin.}\}, q}\}_{q=1}^Q\}$ is a collection of the kernel parameters and was simultaneously optimized by maximizing Eq. (16). We also adopted this kernel to Bayesian GP-LVM following to [39]. The latent space is scaled along its axes by the ARD parameters $a_{\{\text{SE}, \text{linear}\}, q}$, and it is expected to find several geometric properties by this scaling. We optimized these parameters by the well-established quasi-Newton algorithm L-BFGS-B method [49].

Quality Metrics. To quantify the derived latent variables, we used two metrics, *Kullback-Leibler divergence* (KL_σ) and *trustworthiness* (Trust) [44]. We evaluated the global preservation based on KL_σ and the local one based on Trust. To calculate KL_σ , we estimated the density of the observed and latent space based on the kernel density estimation methods [8, 9] and then calculate the KL divergence between those densities. $\sigma \in \mathbb{R}_{>0}$ is the lengthscale parameter of the kernel function and multiple values were chosen based on previous research [28]. Trust measures whether the k -nearest neighbors in the observed space is preserved in the latent space, and we set k to 3. In *Noisy* COIL20 and *Noisy* MNIST, we were interested in the preservation of the data structure before the noise disruption and calculated each metric between the derived latent variables and vanilla COIL20 and MNIST, respectively.

4.2 Results

Visualization. Figure 3 shows the visualization results of all the datasets. In COIL20, all methods can preserve the global relationship and local geometry we expected. However, in *Noisy* COIL20, we confirm that the added missing pixel noise has a significant impact on LLE, t-SNE, and UMAP and that even the global object separation is barely preserved. Since PM and Bayesian GP-LVM learn the global relationship based on the sample similarity, they successfully separated each object under the noise disruption, and PM completely recovered the local rotation geometry. In MNIST and *Noisy* MNIST, we can find a similar tendency to the results of COIL20. Although t-SNE and UMAP behave well in MNIST, we can see the accuracy degradation in the *Noisy* case. PM has the best result under the noise (e.g., the separation of ‘3’, ‘5’, and ‘8’ manifolds) and has better boundary of each digit than Bayesian GP-LVM in MNIST and *Noisy* MNIST. We observe that PM only realizes the enclosing structure hidden in the dataset. Bayesian GP-LVM, t-SNE, and UMAP can preserve the separation of each small sphere. However, they cannot realize the large one due to the curse of dimensionality. LLE cannot separate even small spheres, and we observed they

Table 1: Quality evaluation based on the two quality metrics, KL divergence (KL_σ) and Trustworthiness (Trust). The best is boldfaced, and the second best is underlined.

Dataset	Method	$KL_{0.01} \downarrow$	$KL_{0.1} \downarrow$	$KL_1 \downarrow$	Trust \uparrow
COIL20	PM	0.140	0.0407	<u>0.00180</u>	0.981
	Bayesian GP-LVM	0.119	<u>0.0527</u>	0.00161	0.989
	LLE	0.191	0.0624	0.00336	0.948
	t-SNE	0.0299	0.120	0.00677	0.999
	UMAP	<u>0.0312</u>	0.113	0.00630	<u>0.998</u>
<i>Noisy</i> COIL20	PM	<u>0.145</u>	0.0682	0.00170	0.987
	Bayesian GP-LVM	0.104	<u>0.0751</u>	<u>0.00287</u>	<u>0.979</u>
	LLE	0.246	0.110	0.00419	0.880
	t-SNE	0.265	0.128	0.00392	0.892
	UMAP	0.488	0.0818	0.00456	0.905
MNIST	PM	0.108	0.140	0.00155	0.928
	Bayesian GP-LVM	0.130	<u>0.144</u>	<u>0.00157</u>	0.916
	LLE	0.577	0.309	0.00345	0.825
	t-SNE	0.108	0.180	0.00275	0.995
	UMAP	<u>0.129</u>	0.178	0.00296	<u>0.977</u>
<i>Noisy</i> MNIST	PM	0.0765	0.171	0.00254	0.933
	Bayesian GP-LVM	0.186	0.171	0.00206	0.911
	LLE	0.286	0.222	0.00276	0.750
	t-SNE	0.171	0.181	<u>0.00240</u>	<u>0.932</u>
	UMAP	<u>0.101</u>	<u>0.174</u>	0.00253	0.931
Spheres	PM	<u>0.299</u>	0.546	<u>0.0125</u>	0.650
	Bayesian GP-LVM	0.401	0.679	0.0158	0.647
	LLE	0.576	0.696	0.0210	<u>0.659</u>
	t-SNE	0.294	0.516	0.0114	0.687
	UMAP	0.339	<u>0.535</u>	0.0131	0.687

were covered with the point clouds of the large one. LolaGP successfully embeds the five datasets without significant degradation due to noise and dimensionality effects such as t-SNE and UMAP, implying the efficacy of our novelties.

Quantitative Results. We show the quantitative results in Table 1. In KL_σ , We confirm that our proposed method is the best in eight of fifteen entries and the second-best in four, indicating that PM can preserve the global data structure better than the other methods. In Trust, PM shows successful results on average as opposed to t-SNE and UMAP, which have significant accuracy degradation in *Noisy* COIL20 and *Noisy* MNIST. These results imply the validity of our robust neighborhood preservation via the latent variables. However, in Spheres, PM is slightly inferior to t-SNE and UMAP. Unfortunately, the reliable quality measurement of Spheres is a difficult task because these data contain the curse

of dimensionality problem described in 2.1 [28]. Although our method is inferior to t-SNE and UMAP on Trust in the result of Spheres, it is clear that PM outperforms the comparative methods in the other general image datasets and only detects the true neighbors on the manifold in Spheres from the visualization results.

5 Conclusions

We have introduced a novel latent variable model, LolaGP, that can summarize the complex high-dimensional data into the latent variables while preserving global and local structures. We focused on GP-LVM to preserve the global relationship and introduced a novel regularization based on the neighborhood graph to preserve the local geometry. The graph is built with latent variables, which promotes robustness to noise disruption and the curse of dimensionality. Furthermore, we introduced the scalable lower bound of the log-posterior distribution based on the low-rank matrix approximation, which allows us to handle larger datasets than the previous GP-LVM extensions. In the experimental results, we have shown the effectiveness of our proposed method from the qualitative and quantitative perspectives on the natural and even highly disrupted datasets like *Noisy* COIL20 and *Noisy* MNIST.

One drawback of our proposed method is its generativity. Although the latent variables should be visualized discretely for each independent manifold, such as t-SNE and UMAP, the generativity forces the latent variables to change continuously. Our GP-based approach aids in the discovery of the global relationship between the noise effect and the curse of dimensionality, which can be found in various situations. It would be preferable to modify the likelihood function to make it more suitable for visualization in future works.

Acknowledgements. This study was supported in part by JSPS KAKENHI Grant Number JP21H03456 and AMED Grant Number JP21zf0127004.

References

1. Attali, D., Lieutier, A., Salinas, D.: Vietoris–Rips complexes also provide topologically correct reconstructions of sampled shapes. *Computational Geometry* **46**(4), 448–465 (2013)
2. Balasubramanian, M., Schwartz, E.L., Tenenbaum, J.B., de Silva, V., Langford, J.C.: The isomap algorithm and topological stability. *Science* **295**(5552), 7 (2002)
3. Becht, E., McInnes, L., Healy, J., Dutertre, C.A., Kwok, I.W., Ng, L.G., Ginhoux, F., Newell, E.W.: Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology* **37**(1), 38–44 (2019)
4. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* **15**(6), 1373–1396 (2003)
5. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(8), 1798–1828 (2013)

6. Burgess, C.P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., Lerchner, A.: Understanding disentangling in β -VAE. arXiv preprint arXiv:1804.03599 (2018), <https://arxiv.org/abs/1804.03599>
7. Chang, H., Yeung, D.Y.: Robust locally linear embedding. *Pattern Recognition* **39**(6), 1053–1065 (2006)
8. Chazal, F., Cohen-Steiner, D., M  rigot, Q.: Geometric inference for probability measures. *Foundations of Computational Mathematics* **11**(6), 733–751 (2011)
9. Chazal, F., Fasy, B., Lecci, F., Michel, B., Rinaldo, A., Rinaldo, A., Wasserman, L.: Robust topological inference: Distance to a measure and kernel distance. *Journal of Machine Learning Research* **18**(1), 5845–5884 (2017)
10. Coifman, R.R., Lafon, S.: Diffusion maps. *Applied and Computational Harmonic Analysis* **21**(1), 5–30 (2006)
11. Damianou, A.C., Titsias, M.K., Lawrence, N.D.: Variational inference for latent variables and uncertain inputs in Gaussian processes. *Journal of Machine Learning Research* **17**(42), 1–62 (2016)
12. Diaz-Papkovich, A., Anderson-Trocm  , L., Ben-Eghan, C., Gravel, S.: UMAP reveals cryptic population structure and phenotype heterogeneity in large genomic cohorts. *PLoS genetics* **15**(11), e1008432 (2019)
13. Ferris, B., Fox, D., Lawrence, N.D.: Wifi-slam using Gaussian process latent variable models. In: *International Joint Conference on Artificial Intelligence (IJCAI)*, 2480–2485 (2007)
14. Hausmann, J.C.: On the Vietoris-Rips complexes and a cohomology theory for metric spaces. In: *Prospects in Topology: Proceedings of a Conference in Honor of William Browder*, 175–188 (1995)
15. He, X., Cai, D., Yan, S., Zhang, H.J.: Neighborhood preserving embedding. In: *International Conference on Computer Vision (ICCV)*, 1208–1213 (2005)
16. Hensman, J., Fusi, N., Andrade, R., Durrande, N., Saul, A., Zwiessele, M., Lawrence, N. D.: GPy: A Gaussian process framework in python. <http://github.com/SheffieldML/GPy> (2012)
17. Hensman, J., Fusi, N., Lawrence, N.D.: Gaussian processes for big data. arXiv preprint arXiv:1309.6835 (2013), <https://arxiv.org/abs/1309.6835>
18. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: β -VAE: Learning basic visual concepts with a constrained variational framework. In: *International Conference on Learning Representations (ICLR)*, 1–22 (2016)
19. Hotelling, H.: Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* **24**(6), 417–441 (1933)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv Preprint arXiv:1312.6114 (2013), <https://arxiv.org/abs/1312.6114>
21. Lawrence, N.D.: Learning for larger datasets with the Gaussian process latent variable model. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 243–250 (2007)
22. Lawrence, N.D., Hyv  rinen, A.: Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research* **6**(11) (2005)
23. Lu, C., Tang, X.: Surpassing human-level face verification performance on LFW with GaussianFace. In: *AAAI Conference on Artificial Intelligence (AAAI)*, 3811–3819 (2015)
24. Lu, Y., Lai, Z., Xu, Y., Li, X., Zhang, D., Yuan, C.: Low-rank preserving projections. *IEEE Transactions on Cybernetics* **46**(8), 1900–1913 (2015)

25. Moon, K.R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D.B., Chen, W.S., Yim, K., Elzen, A.v.d., Hirn, M.J., Coifman, R.R., et al.: Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology* **37**(12), 1482–1492 (2019)
26. MacKay, D.J.: Bayesian nonlinear modeling for the prediction competition. *ASHRAE transactions* **100**(2), 1053–1062 (1994)
27. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018), <https://arxiv.org/abs/1802.03426>
28. Moor, M., Horn, M., Rieck, B., Borgwardt, K.: Topological autoencoders. In: *International Conference on Machine Learning (ICML)*, 7045–7054 (2020)
29. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library (coil-20) (1996), <https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>
30. Ng, Y.C., Colombo, N., Silva, R.: Bayesian semi-supervised learning with graph Gaussian processes. In: *Advances in Neural Information Processing (NeurIPS)* (2018)
31. Ordun, C., Purushotham, S., Raff, E.: Exploratory analysis of COVID-19 tweets using topic modeling, UMAP, and DiGraphs. *arXiv preprint arXiv:2005.03082* (2020), <https://arxiv.org/abs/2005.03082>
32. Quinonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research* **6**(65), 1939–1959 (2005)
33. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**(5500), 2323–2326 (2000)
34. Saul, L.K.: A tractable latent variable model for nonlinear dimensionality reduction. *Proceedings of the National Academy of Sciences* **117**(27), 15403–15408 (2020)
35. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* **10**(5), 1299–1319 (1998)
36. Song, G., Wang, S., Huang, Q., Tian, Q.: Harmonized multimodal learning with Gaussian process latent variable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**(3), 858–872 (2021)
37. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **61**(3), 611–622 (1999)
38. Titsias, M.: Variational learning of inducing variables in sparse Gaussian processes. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 567–574 (2009)
39. Titsias, M., Lawrence, N.D.: Bayesian Gaussian process latent variable model. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 844–851 (2010)
40. Urtasun, R., Darrell, T.: Discriminative Gaussian process latent variable model for classification. In: *International Conference on Machine Learning (ICML)*, 927–934 (2007)
41. Urtasun, R., Fleet, D.J., Geiger, A., Popović, J., Darrell, T.J., Lawrence, N.D.: Topologically-constrained latent variable models. In: *International Conference on Machine Learning (ICML)*, 1080–1087 (2008)
42. Van Der Maaten, L., Postma, E., Van den Herik, J., et al.: Dimensionality reduction: a comparative. Technical Report TiCC-TR 2009-005 (2009)
43. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *Journal of Machine Learning Research* **9**(11), 2579–2605 (2008)

- 44. Venna, J., Kaski, S.: Visualizing gene interaction graphs with local multidimensional scaling. In: European Symposium on Artificial Neural Networks (ESANN), 557–562 (2006)
- 45. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(2), 283–298 (2007)
- 46. Wasserman, L.: Topological data analysis. *Annual Review of Statistics and Its Application* **5**(1), 501–532 (2018)
- 47. You, Z.H., Lei, Y.K., Gui, J., Huang, D.S., Zhou, X.: Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data. *Bioinformatics* **26**(21), 2744–2751 (2010)
- 48. Zhang, Y.J., Pan, S., He, L., Ling, Z.H.: Learning latent representations for style control and transfer in end-to-end speech synthesis. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6945–6949 (2019)
- 49. Zhu, C., Byrd, R.H., Lu, P., Nocedal, J.: Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software* **23**(4), 550–560 (1997)