Generative Locally Linear Embedding

Benyamin Ghojogh[1], Ali Ghodsi[2], Fakhri Karray[3], *Fellow, IEEE*, Mark Crowley[1]

Fig. 1. The probabilistic graphical model for GLLE algorithm.

Abstract—Locally Linear Embedding (LLE) is a nonlinear spectral dimensionality reduction and manifold learning method. It has two main steps which are linear reconstruction and linear embedding of points in the input space and embedding space, respectively. In this work, we propose two novel generative versions of LLE, named Generative LLE (GLLE), whose linear reconstruction steps are stochastic rather than deterministic. GLLE assumes that every data point is caused by its linear reconstruction weights as latent factors. The proposed GLLE algorithms can generate various LLE embeddings stochastically while all the generated embeddings relate to the original LLE embedding. We propose two versions for stochastic linear reconstruction, one using expectation maximization and another with direct sampling from a derived distribution by optimization. The proposed GLLE methods are closely related to and inspired by variational inference, factor analysis, and probabilistic principal component analysis. Our simulations show that the proposed GLLE methods work effectively in unfolding and generating submanifolds of data.

I. INTRODUCTION

Dimensionality reduction and manifold learning methods are widely useful for feature extraction, manifold unfolding, and data visualization [1]. The dimensionality reduction methods can be divided into three main categories, i.e., spectral methods, probabilistic methods, and neural network-based methods [2]. An example for spectral methods is Locally Linear Embedding (LLE) [3], [4]. Examples for probabilistic methods are factor analysis [5], [6] and probabilistic Principal Component Analysis (PCA) [7], [8]. An example for neural network-based methods is variational autoencoder [9] which formulates variational inference [10], [11], [12], [13] in an autoencoder framework. Another example for this category is adversarial autoencoder [14].

In this work, we propose two versions of Generative LLE (GLLE). GLLE is a combination of spectral and probabilistic methods. It is related to LLE, factor analysis, probabilistic PCA, and variational inference. Hence, we can say that it combines different categories of dimensionality reduction methods. The original LLE, which is a spectral method, has two main steps which are deterministic linear reconstruction of points in the input space and deterministic linear embedding in the embedding space. GLLE replaces deterministic linear reconstruction with a stochastic linear reconstruction while the linear embedding step is the same as in LLE.

In GLLE, we assume that reconstruction weights are latent factors causing the data points in a probabilistic graphical model; see Fig. 1. Every data point is obtained by a function of its stochastic reconstruction weights. In the first proposed version of GLLE, the formulation of this function is similar to, but not the same as, the formulation in factor analysis and probabilistic PCA. The covariance matrices of the reconstruction weights are calculated using Expectation Maximization (ME) for the sake of Maximum Likelihood Estimation (MLE). In the second version of GLLE, we propose linear reconstruction with direct sampling in which the distribution of weights is obtained directly by some optimization derivations. The former approach is more solid in terms of theory while the latter is simpler to implement and slightly faster.

The proposed GLLE algorithms can be categorized as generative models [15]. It should be noted that they are not deep or autoencoder generative models [16], [17]. Rather, they are combinations of spectral and probabilistic approaches for dimensionality reduction. The proposed GLLE methods are useful for generating several LLE embeddings stochastically where the generated embeddings are related to the original LLE embedding. One can obtain more insights into the LLE embedding by generating more related embeddings and investigating into the new generated embeddings. Moreover, having a stochastic method rather than deterministic one gives the opportunity to have more number of embeddings for visualization and feature extraction without the need to several algorithms.

The remainder of this paper is organized as follows. Section II reviews the related and required technical background. The proposed stochastic linear reconstruction with EM is explained in Section III. Section IV explains the proposed stochastic linear reconstruction with direct sampling. The last step of GLLE, linear embedding, is explained in Section V. Experimental results are provided in Section VI. Finally, Section VII concludes the paper.

[1]Benyamin Ghojogh and Mark Crowley are with the Machine Learning Laboratory, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. [2]Ali Ghodsi is with the Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, ON, Canada. [3]Fakhri Karray is with the Centre for Pattern Analysis and Machine Intelligence, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Emails: {bghojogh, ali.ghodsi, karray, mcrowley}@uwaterloo.ca
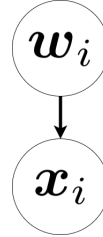
II. TECHNICAL BACKGROUND

In this section, we review some related technical background on joint distributions, linear reconstruction in LLE, variational inference, factor analysis, and probabilistic PCA.

A. Marginal Multivariate Gaussian Distribution

Consider two random variables $\boldsymbol{x}_1 \in \mathbb{R}^{d_1}$ and $\boldsymbol{x}_2 \in \mathbb{R}^{d_2}$ and let $\boldsymbol{x}_3 := [\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top]^\top \in \mathbb{R}^{d_1+d_2}$. Assume that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ jointly multivariate Gaussian, i.e., $\boldsymbol{x}_3 \sim \mathcal{N}(\boldsymbol{x}_3; \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$. The mean and covariance can be decomposed as:

$$\boldsymbol{\mu}_3 = [\boldsymbol{\mu}_1^\top, \boldsymbol{\mu}_2^\top]^\top \in \mathbb{R}^{d_1+d_2}, \tag{1}$$

$$\boldsymbol{\Sigma}_3 = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \in \mathbb{R}^{(d_1+d_2)\times(d_1+d_2)}, \tag{2}$$

where $\boldsymbol{\mu}_1 \in \mathbb{R}^{d_1}$, $\boldsymbol{\mu}_2 \in \mathbb{R}^{d_2}$, $\boldsymbol{\Sigma}_{11} \in \mathbb{R}^{d_1 \times d_2}$, $\boldsymbol{\Sigma}_{22} \in \mathbb{R}^{d_2 \times d_2}$, $\boldsymbol{\Sigma}_{12} \in \mathbb{R}^{d_1 \times d_2}$, and $\boldsymbol{\Sigma}_{21} = \boldsymbol{\Sigma}_{12}^\top$.

It can be shown that the marginal distributions for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are Gaussian distributions where $\mathbb{E}[\boldsymbol{x}_1] = \boldsymbol{\mu}_1$ and $\mathbb{E}[\boldsymbol{x}_2] = \boldsymbol{\mu}_2$. The covariance matrix of the joint distribution can be simplified as [13]:

$$\boldsymbol{\Sigma}_3 = \mathbb{E}\left[\begin{bmatrix} (\boldsymbol{x}_1-\boldsymbol{\mu}_1)(\boldsymbol{x}_1-\boldsymbol{\mu}_1)^\top, (\boldsymbol{x}_1-\boldsymbol{\mu}_1)(\boldsymbol{x}_2-\boldsymbol{\mu}_2)^\top \\ (\boldsymbol{x}_2-\boldsymbol{\mu}_2)(\boldsymbol{x}_1-\boldsymbol{\mu}_1)^\top, (\boldsymbol{x}_2-\boldsymbol{\mu}_2)(\boldsymbol{x}_2-\boldsymbol{\mu}_2)^\top \end{bmatrix} \right], \tag{3}$$

where $\mathbb{E}[.]$ is the expectation operator. According to the definition of the multivariate Gaussian distribution, the conditional distribution is also a Gaussian distribution, i.e., $\boldsymbol{x}_2 | \boldsymbol{x}_1 \sim \mathcal{N}(\boldsymbol{x}_2; \boldsymbol{\mu}_{x_2|x_1}, \boldsymbol{\Sigma}_{x_2|x_1})$ where [13]:

$$\mathbb{R}^{d_2} \ni \boldsymbol{\mu}_{x_2|x_1} := \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1), \tag{4}$$

$$\mathbb{R}^{d_2 \times d_2} \ni \boldsymbol{\Sigma}_{x_2|x_1} := \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}, \tag{5}$$

and likewise we have for $\boldsymbol{x}_1 | \boldsymbol{x}_2 \sim \mathcal{N}(\boldsymbol{x}_1; \boldsymbol{\mu}_{x_1|x_2}, \boldsymbol{\Sigma}_{x_1|x_2})$. Also, note that the probability density function of d-dimensional Gaussian distribution is:

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2} \right), \tag{6}$$

where $|.|$ denotes the determinant of matrix.

B. Linear Reconstruction in Locally Linear Embedding

A kNN graph is formed using pairwise Euclidean distance between the data points. Therefore, every data point has k neighbors. Let $\boldsymbol{x}_{ij} \in \mathbb{R}^d$ denote the j-th neighbor of $\boldsymbol{x}_i \in \mathbb{R}^d$ and let the matrix $\mathbb{R}^{d \times k} \ni \boldsymbol{X}_i := [\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{ik}]$ include the k neighbors of $\boldsymbol{x}_i$. Suppose $\boldsymbol{w}_i := [w_{i1}, \ldots, w_{ik}]^\top \in \mathbb{R}^k$ denotes the reconstruction weights of every point $\boldsymbol{x}_i$ by its neighbors $\boldsymbol{X}_i$.

The linear reconstruction in original LLE [3], [4] is deterministic. In LLE, the weights for linear reconstruction of every point by its kNN are optimized as [18]:

$$\begin{aligned} \underset{\{\boldsymbol{w}_i\}_{i=1}^n}{\text{minimize}} \quad & \sum_{i=1}^n \left\| \boldsymbol{x}_i - \sum_{j=1}^k w_{ij}\boldsymbol{x}_{ij} \right\|_2^2, \\ \text{subject to} \quad & \sum_{j=1}^k w_{ij} = 1, \quad \forall i \in \{1, \ldots, n\}, \end{aligned} \tag{7}$$

which can be restated as:

$$\begin{aligned} \underset{\{\boldsymbol{w}_i\}_{i=1}^n}{\text{minimize}} \quad & \sum_{i=1}^n \boldsymbol{w}_i^\top \boldsymbol{G}_i \boldsymbol{w}_i, \\ \text{subject to} \quad & \mathbf{1}^\top \boldsymbol{w}_i = 1, \quad \forall i \in \{1, \ldots, n\}, \end{aligned} \tag{8}$$

where $\mathbb{R}^{k \times k} \ni \boldsymbol{G}_i := (\boldsymbol{x}_i \mathbf{1}^\top - \boldsymbol{X}_i)^\top (\boldsymbol{x}_i \mathbf{1}^\top - \boldsymbol{X}_i)$ and its solution is [18]:

$$\boldsymbol{w}_i = \frac{\boldsymbol{G}_i^{-1}\mathbf{1}}{\mathbf{1}^\top \boldsymbol{G}_i^{-1}\mathbf{1}}. \tag{9}$$

The obtained weights $\{\boldsymbol{w}_i\}_{i=1}^n$ are then used for linear embedding in the low dimensional space. In generative LLE, we replace the deterministic linear embedding with stochastic linear reconstruction.

C. Variational Inference

Assume data $\boldsymbol{x}_i$ is caused by a latent factor $\boldsymbol{w}_i$, as shown in Fig. 1. Let the parameters of model be denoted by $\boldsymbol{\theta}$. In variational inference, the Evidence Lower Bound (ELBO) is a lower bound on the log likelihood of data and is defined as minus Kullback-Leibler (KL) divergence between a distribution $q(.)$ on $\boldsymbol{w}_i$ and the joint distribution of $\boldsymbol{x}_i$ and $\boldsymbol{w}_i$. [13], [10], [9]:

$$\mathcal{L}(q, \boldsymbol{\theta}) := -\text{KL}\big(q(\boldsymbol{w}_i) \,\|\, \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{w}_i \,|\, \boldsymbol{\theta})\big). \tag{10}$$

Maximizing this lower bound results in maximization of likelihood of data [11], [12]. Variational inference uses EM for MLE [13]:

$$q^{(t)}(\boldsymbol{w}_i) \leftarrow \mathbb{P}(\boldsymbol{w}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}^{(t-1)}), \tag{11}$$

$$\boldsymbol{\theta}^{(t)} \leftarrow \arg\max_{\boldsymbol{\theta}} \; \mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}\big[\log \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{w}_i \,|\, \boldsymbol{\theta})\big], \tag{12}$$

where $\mathbb{E}[.]$ denotes expectation. Here, we use the EM approach of variational inference for stochastic linear reconstruction in GLLE.

D. Factor Analysis

Factor analysis [5], [6] assumes that every data point $\boldsymbol{x}_i$ is generated from a latent factor $\boldsymbol{w}_i$. Its probabilistic graphical model is similar to Fig. 1 but with a small difference [19]. It assumes $\boldsymbol{x}_i$ is obtained by linear projection of $\boldsymbol{w}_i$ onto the space of data by projection matrix $\boldsymbol{\Lambda}$, then applying some linear translation, and finally adding a Gaussian noise $\boldsymbol{\epsilon}$ with covariance matrix $\boldsymbol{\Psi}$. This addition of noise is the main difference from the model depicted in Fig. 1. If $\boldsymbol{\mu}$ denotes the mean of data, factor analysis considers:

$$\boldsymbol{x}_i := \boldsymbol{\Lambda}\boldsymbol{w}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{13}$$

$$\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{w}_i, \boldsymbol{\Lambda}, \boldsymbol{\mu}, \boldsymbol{\Psi}) = \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\Lambda}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi}), \tag{14}$$

where $\mathbb{P}(\boldsymbol{w}_i) = \mathcal{N}(\boldsymbol{w}_i; \mathbf{0}, \boldsymbol{I})$ and $\mathbb{P}(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \boldsymbol{\Psi})$. Factor analysis uses EM algorithm for finding optimum $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$ (see [13] for details of EM in factor analysis). In stochastic linear reconstruction for GLLE, we are inspired by Eq. (13) for modeling the relation between data point and its reconstruction weights as its latent factor.

E. Probabilistic PCA

Probabilistic PCA [7], [8] is a special case of factor analysis where the variance of noise is equal in all dimensions of data space with covariance between dimensions, i.e.:

$$\boldsymbol{\Psi} = \sigma^2 \boldsymbol{I}. \tag{15}$$

In other words, probabilistic PCA considers an isotropic noise model. Similar to factor analysis, it can be solved iteratively using EM [7]. However, one can also find a closed-form solution to its EM approach [8]. Hence, by restricting the noise covariance to be isotropic, its solution becomes simpler and closed-form. See [13] for details of derivations and closed-form solution for probabilistic PCA. Inspired by probabilistic PCA, in stochastic linear reconstruction for GLLE, we relax the covariance matrix of reconstruction weight to be diagonal. Interestingly, without this relaxation, the solution of M-step in EM algorithm of GLLE is solved iteratively; however, this relaxation makes the solution of M-step in EM algorithm of GLLE closed-form, as we also see the effect of relaxation in probabilistic PCA.

III. STOCHASTIC LINEAR RECONSTRUCTION WITH EXPECTATION MAXIMIZATION

In this section, we propose the novel stochastic linear reconstruction for GLLE where EM is used for MLE and calculating the covariance matrices of the reconstruction weights of points.

A. Notations and Joint and Conditional Distributions

As Fig. 1 depicts, we assume that the every point $\boldsymbol{x}_i$ is generated by its reconstruction weights $\boldsymbol{w}_i$ as a latent factor. Therefore, $\boldsymbol{x}_i$ can be written as a stochastic function of $\boldsymbol{w}_i$ where we assume $\boldsymbol{w}_i$ has a multivariate Gaussian distribution:

$$\boldsymbol{x}_i = \boldsymbol{X}_i \boldsymbol{w}_i + \boldsymbol{\mu}, \tag{16}$$

$$\mathbb{P}(\boldsymbol{w}_i) = \mathcal{N}(\boldsymbol{w}_i; \boldsymbol{0}, \boldsymbol{\Omega}_i) \implies \mathbb{E}[\boldsymbol{w}_i] = \boldsymbol{0}, \quad \mathbb{E}[\boldsymbol{w}_i \boldsymbol{w}_i^\top] = \boldsymbol{\Omega}_i, \tag{17}$$

where $\boldsymbol{\Omega}_i \in \mathbb{R}^{k \times k}$ is covariance of $\boldsymbol{w}_i$ and $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean of data because:

$$\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i = \mathbb{E}[\boldsymbol{x}_i] = \boldsymbol{X}_i \mathbb{E}[\boldsymbol{w}_i] + \boldsymbol{\mu} \overset{(17)}{=} \boldsymbol{0} + \boldsymbol{\mu} = \boldsymbol{\mu}. \tag{18}$$

According to Eq. (3), for the joint distribution of $[\boldsymbol{x}_i^\top, \boldsymbol{w}_i^\top]^\top \in \mathbb{R}^{d+k}$, we have:

$$\boldsymbol{\Sigma}_{11} = \mathbb{E}[(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top] \overset{(16)}{=} \mathbb{E}[(\boldsymbol{X}_i \boldsymbol{w}_i)(\boldsymbol{X}_i \boldsymbol{w}_i)^\top]$$
$$= \boldsymbol{X}_i \mathbb{E}[\boldsymbol{w}_i \boldsymbol{w}_i^\top] \boldsymbol{X}_i^\top \overset{(17)}{=} \boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top,$$

$$\boldsymbol{\Sigma}_{12} = \mathbb{E}[(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{w}_i - \boldsymbol{0})^\top] \overset{(16)}{=} \boldsymbol{X}_i \mathbb{E}[\boldsymbol{w}_i \boldsymbol{w}_i^\top] \overset{(17)}{=} \boldsymbol{X}_i \boldsymbol{\Omega}_i,$$

$$\boldsymbol{\Sigma}_{22} = \mathbb{E}[(\boldsymbol{w}_i - \boldsymbol{0})(\boldsymbol{w}_i - \boldsymbol{0})^\top] = \mathbb{E}[\boldsymbol{w}_i \boldsymbol{w}_i^\top] \overset{(17)}{=} \boldsymbol{\Omega}_i.$$

Hence:

$$\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{w}_i \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{w}_i \end{bmatrix} ; \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top & \boldsymbol{X}_i \boldsymbol{\Omega}_i \\ \boldsymbol{\Omega}_i^\top \boldsymbol{X}_i^\top & \boldsymbol{\Omega}_i \end{bmatrix} \right). \tag{19}$$

We have:

$$\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{w}_i, \boldsymbol{\Omega}_i) \overset{(a)}{=} \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{X}_i \boldsymbol{w}_i + \boldsymbol{\mu}, \boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top), \tag{20}$$

where (a) is because of Eqs. (16) and (19). We use EM for MLE in stochastic linear reconstruction. In the following, the steps of EM are explained.

B. E-Step in Expectation Maximization

As we will see later in the M-step of EM, we will have two expectation terms which need to be computed in the E-step. These expectations, which are over the $q(\boldsymbol{w}_i) := \mathbb{P}(\boldsymbol{w}_i \,|\, \boldsymbol{x}_i)$ distribution, are $\mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}[\boldsymbol{w}_i] \in \mathbb{R}^k$ and $\mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}[\boldsymbol{w}_i \boldsymbol{w}_i^\top] \in \mathbb{R}^{k \times k}$ where t denotes the iteration index in EM iterations. According to Eqs. (4), (5), and (19), we have:

$$\mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}[\boldsymbol{w}_i] = \boldsymbol{\mu}_{w|x} = \boldsymbol{\Omega}_i^\top \boldsymbol{X}_i^\top (\boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top)^\dagger (\boldsymbol{x}_i - \boldsymbol{\mu}), \tag{21}$$

$$\mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}[\boldsymbol{w}_i \boldsymbol{w}_i^\top] = \boldsymbol{\Sigma}_{w|x}$$
$$= \boldsymbol{\Omega}_i - \boldsymbol{\Omega}_i^\top \boldsymbol{X}_i^\top (\boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top)^\dagger \boldsymbol{X}_i \boldsymbol{\Omega}_i, \tag{22}$$

where $\dagger$ denotes either inverse or pseudo-inverse of matrix.

C. M-Step in Expectation Maximization

In M-step of EM, we maximize the joint likelihood of data and weights over all n data points where the optimization variable is the covariance of prior distribution of weights:

$$\max_{\{\boldsymbol{\Omega}_i\}_{i=1}^{n}} \sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}\big[\log \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{w}_i \,|\, \boldsymbol{\Omega}_i)\big]$$

$$\overset{(a)}{=} \max_{\{\boldsymbol{\Omega}_i\}_{i=1}^{n}} \sum_{i=1}^{n} \Big(\mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}\big[\log \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{w}_i, \boldsymbol{\Omega}_i)\big]$$
$$+ \mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}\big[\log \mathbb{P}(\boldsymbol{w}_i)\big]\Big)$$

$$\overset{(b)}{=} \max_{\{\boldsymbol{\Omega}_i\}_{i=1}^{n}} \sum_{i=1}^{n} \Big(\mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}\big[\log \mathcal{N}(\boldsymbol{X}_i \boldsymbol{w}_i + \boldsymbol{\mu}, \boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top)\big]$$
$$+ \mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}\big[\log \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Omega}_i))\big]\Big)$$

$$\overset{(6)}{=} \max_{\{\boldsymbol{\Omega}_i\}_{i=1}^{n}} \Big(\underbrace{-\frac{d\,n}{2}\log(2\pi)}_{\text{constant}} - \frac{n}{2}\log|\boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top|$$
$$- \frac{1}{2}\sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}\big[(\boldsymbol{x}_i - \boldsymbol{X}_i \boldsymbol{w}_i - \boldsymbol{\mu})^\top (\boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top)^{-1}$$
$$(\boldsymbol{x}_i - \boldsymbol{X}_i \boldsymbol{w}_i - \boldsymbol{\mu})\big] - \underbrace{\frac{k\,n}{2}\log(2\pi)}_{\text{constant}}$$
$$- \frac{n}{2}\log|\boldsymbol{\Omega}_i| - \frac{1}{2}\sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)}\big[\boldsymbol{w}_i^\top \boldsymbol{\Omega}_i^{-1} \boldsymbol{w}_i\big]\Big)$$

$$\overset{(c)}{=} \max_{\{\boldsymbol{\Omega}_i\}_{i=1}^{n}} \Big(-\frac{n}{2}\log|\boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top| - \frac{n}{2}\mathbf{tr}\big((\boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top)^{-1} \boldsymbol{S}_1\big)$$
$$- \frac{n}{2}\log|\boldsymbol{\Omega}_i| - \frac{n}{2}\mathbf{tr}\big(\boldsymbol{\Omega}_i^{-1} \boldsymbol{S}_2\big)\Big), \tag{23}$$

where (a) is because of the chain rule $\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{w}_i \,|\, \boldsymbol{\Omega}_i) = \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{w}_i, \boldsymbol{\Omega}_i)\, \mathbb{P}(\boldsymbol{w}_i)$, and (b) is because of Eqs. (17) and (20), and (c) is because we define the scatters $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ as:

$$\mathbb{R}^{d \times d} \ni \boldsymbol{S}_1 := \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)} \big[(\boldsymbol{x}_i - \boldsymbol{X}_i \boldsymbol{w}_i - \boldsymbol{\mu})$$

$$(\boldsymbol{x}_i - \boldsymbol{X}_i \boldsymbol{w}_i - \boldsymbol{\mu})^\top \big] = \frac{1}{n} \sum_{i=1}^{n} \Big((\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top$$

$$- 2 \boldsymbol{X}_i \mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)} \big[\boldsymbol{w}_i \big] (\boldsymbol{x}_i - \boldsymbol{\mu})^\top$$

$$+ \boldsymbol{X}_i \mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)} \big[\boldsymbol{w}_i \boldsymbol{w}_i^\top \big] \boldsymbol{X}_i^\top \Big), \tag{24}$$

$$\mathbb{R}^{k \times k} \ni \boldsymbol{S}_2 := \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{w}_i)} \big[\boldsymbol{w}_i \boldsymbol{w}_i^\top \big], \tag{25}$$

where the expectation terms are found by Eqs. (21) and (22). The gradient of the joint likelihood is:

$$\mathbb{R}^{k \times k} \ni \frac{\partial \text{ Eq. (23)}}{\partial \boldsymbol{\Omega}_i^{-1}} = \frac{n}{2} \Big[\textbf{vec}_{k \times k}^{-1} \big[\boldsymbol{T}_i \, \textbf{vec}_{d^2 \times 1}(\boldsymbol{X}_i \boldsymbol{\Omega}_i \boldsymbol{X}_i^\top) \big]$$

$$- \textbf{vec}_{k \times k}^{-1} \big[\boldsymbol{T}_i \, \textbf{vec}_{d^2 \times 1}(\boldsymbol{S}_1) \big] + \boldsymbol{\Omega}_i - \boldsymbol{S}_2 \Big], \tag{26}$$

where we use the Magnus-Neudecker convention in which matrices are vectorized, $\textbf{vec}(.)$ vectorizes the matrix, $\textbf{vec}_{k \times k}^{-1}(.)$ is de-vectorization to $k \times k$ matrix, $\otimes$ denotes the Kronecker product, and $\mathbb{R}^{k^2 \times d^2} \ni \boldsymbol{T}_i := \boldsymbol{X}_i^\top \otimes \boldsymbol{X}_i^\top$.

In the M-step, one can update the variables $\{\boldsymbol{\Omega}_i\}_{i=1}^{n}$ using gradient descent [20] with the gradient in Eq. (26). However, inspired by probabilistic PCA, we can relax the covariance matrix and simplify the algorithm.

D. Relaxation of Covariance

Inspired by relaxation of factor analysis for probabilistic PCA, we can relax the covariance matrix to be diagonal and the variance of weights to be equal in all k dimensions:

$$\boldsymbol{\Omega}_i = \sigma_i \boldsymbol{I} \in \mathbb{R}^{k \times k}. \tag{27}$$

Substituting this covariance into Eq. (23) and noticing the properties of determinant and trace gives:

$$\max_{\{\sigma_i\}_{i=1}^{n}} \Big(-\frac{n}{2} \log(\sigma_i^d |\boldsymbol{X}_i \boldsymbol{X}_i^\top|) - \frac{n}{2} \sigma_i^{-1} \textbf{tr}((\boldsymbol{X}_i \boldsymbol{X}_i^\top)^{-1} \boldsymbol{S}_1)$$

$$- \frac{n}{2} \log \sigma_i^k - \frac{n}{2} \sigma_i^{-1} \textbf{tr}(\boldsymbol{S}_2) \Big)$$

$$\overset{(a)}{=} \max_{\{\sigma_i\}_{i=1}^{n}} \Big(-\frac{n}{2}(d+k) \log(\sigma_i)$$

$$- \frac{n}{2} \Big[\textbf{tr}((\boldsymbol{X}_i \boldsymbol{X}_i^\top)^{-1} \boldsymbol{S}_1) - \textbf{tr}(\boldsymbol{S}_2) \Big] \sigma_i^{-1} \Big), \tag{28}$$

where (a) is because $\log(\sigma_i^d |\boldsymbol{X}_i \boldsymbol{X}_i^\top|) = d \log(\sigma_i) + \log(|\boldsymbol{X}_i \boldsymbol{X}_i^\top|)$ whose second term is a constant w.r.t. σ_i. Setting the gradient of the joint likelihood to zero gives:

$$\mathbb{R} \ni \frac{\partial \text{ Eq. (28)}}{\partial \sigma_i^{-1}} = \frac{n}{2} \Big[(d+k) \sigma_i$$

$$- \Big(\textbf{tr}((\boldsymbol{X}_i \boldsymbol{X}_i^\top)^{-1} \boldsymbol{S}_1) + \textbf{tr}(\boldsymbol{S}_2) \Big) \Big] \overset{\text{set}}{=} 0$$

$$\implies \sigma_i = (d+k)^{-1} \Big(\textbf{tr}((\boldsymbol{X}_i \boldsymbol{X}_i^\top)^\dagger \boldsymbol{S}_1) + \textbf{tr}(\boldsymbol{S}_2) \Big), \tag{29}$$

Algorithm 1: Stochastic Linear Reconstruction with Expectation Maximization

1 **Input:** kNN graph or $\{\boldsymbol{X}_i\}_{i=1}^{n}$, $\{\boldsymbol{x}_i\}_{i=1}^{n}$
2 **Initialize:** $\{\boldsymbol{\Omega}_i\}_{i=1}^{n} = \boldsymbol{I}$
3 **while** *not converged* **do**
4   // E-step:
5   **for** *every data point i from 1 to n* **do**
6   Calculate expectations by Eqs. (21) and (22)
7   // Sampling:
8   Sample weights $\{\boldsymbol{w}_i\}_{i=1}^{n}$ using Eq. (31)
9   // M-step:
10   Calculate $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ using Eqs. (24) and (25)
11   **for** *every data point i from 1 to n* **do**
12   Calculate σ_i by Eq. (29)
13   Calculate $\boldsymbol{\Omega}_i$ using Eq. (27)
14 **Return** weights $\{\boldsymbol{w}_i\}_{i=1}^{n}$

---

where † denotes either inverse or pseudo-inverse of matrix. The EM algorithm for stochastic linear reconstruction in GLLE is summarized in Algorithm 1. We can sample $\{\boldsymbol{w}_i\}_{i=1}^{n}$ with the following prior and conditional distributions:

$$\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{w}_i; \boldsymbol{0}, \sigma_i \boldsymbol{I}), \tag{30}$$

$$\boldsymbol{w}_i \,|\, \boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{w}_i; \boldsymbol{\mu}_{w|x}, \boldsymbol{\Sigma}_{w|x}), \tag{31}$$

where $\boldsymbol{\mu}_{w|x}$ and $\boldsymbol{\Sigma}_{w|x}$ are defined in Eqs. (21) and (22), respectively.

IV. STOCHASTIC LINEAR RECONSTRUCTION WITH DIRECT SAMPLING

In this section, we propose the novel stochastic linear reconstruction for GLLE where the distributions of the reconstruction weights of points are calculated by derivations of optimization and the weights are sampled directly from those distributions.

The stochastic linear reconstruction with direct sampling is initialized by the original LLE [3], [4]. Assume we have the embedding of LLE; therefore, we have $\{\boldsymbol{X}_i \in \mathbb{R}^{d \times k}\}_{i=1}^{n}$ and $\{\boldsymbol{Y}_i \in \mathbb{R}^{p \times k}\}_{i=1}^{n}$ from the kNN graph of data and the LLE embedding, respectively. Note that columns of $\boldsymbol{X}_i$ are the neighbor points to $\boldsymbol{x}_i$ and columns of $\boldsymbol{Y}_i$ are the embeddings of $\boldsymbol{X}_i$. From the original LLE, we also have the embeddings and reconstruction weights of points, denoted by $\{\boldsymbol{y}_i\}_{i=1}^{n}$ and $\{\boldsymbol{w}_i^{\text{LLE}}\}_{i=1}^{n}$, respectively.

In stochastic linear reconstruction with direct sampling, we find the reconstruction weights in a way that every point is reconstructed best by a linear combination of its neighbors in both input and embedded spaces:

$$\underset{\{\boldsymbol{w}_i\}_{i=1}^{n}}{\text{minimize}} \quad \sum_{i=1}^{n} \Big(\|\boldsymbol{x}_i - \boldsymbol{X}_i \boldsymbol{w}_i\|_2^2 + \|\boldsymbol{y}_i - \boldsymbol{Y}_i \boldsymbol{w}_i\|_2^2 \Big). \tag{32}$$

This problem can be seen as minimization for every point. We assume a k-dimensional Gaussian distribution for the

Algorithm 2: Stochastic Linear Reconstruction with Direct Sampling

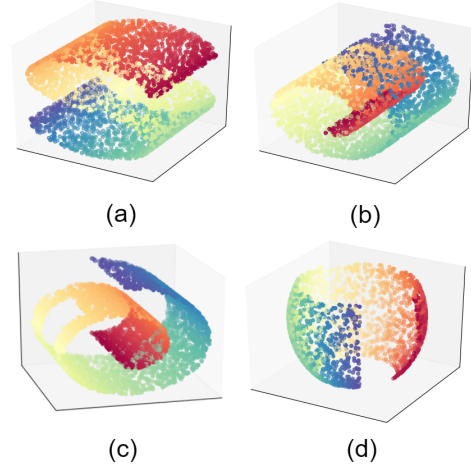(a)   (b)

(c)   (d)

Fig. 2. Three synthetic nonlinear manifolds used for unfolding experiments: (a) S-curve, (b) Swiss roll, (c) Swiss roll with hole, and (d) severed bowl.

conditional distribution $\mathbb{P}(\boldsymbol{w}_i \,|\, \boldsymbol{y}_i, \boldsymbol{x}_i)$. The minimization (32) can be restated to the following maximization:

$$\underset{\boldsymbol{w}_i}{\text{maximize}} \quad \mathbb{P}(\boldsymbol{w}_i \,|\, \boldsymbol{y}_i, \boldsymbol{x}_i) \propto$$
$$\exp\left(-\left(\|\boldsymbol{x}_i - \boldsymbol{X}_i \boldsymbol{w}_i\|_2^2 + \|\boldsymbol{y}_i - \boldsymbol{Y}_i \boldsymbol{w}_i\|_2^2\right)\right). \quad (33)$$

We can simplify the term in the exponential as:

$$\|\boldsymbol{x}_i - \boldsymbol{X}_i \boldsymbol{w}_i\|_2^2 + \|\boldsymbol{y}_i - \boldsymbol{Y}_i \boldsymbol{w}_i\|_2^2 = \boldsymbol{x}_i^\top \boldsymbol{x}_i - 2\boldsymbol{x}_i^\top \boldsymbol{X}_i \boldsymbol{w}_i$$
$$+ \boldsymbol{w}_i^\top \boldsymbol{X}_i^\top \boldsymbol{X}_i \boldsymbol{w}_i + \boldsymbol{y}_i^\top \boldsymbol{y}_i - 2\boldsymbol{y}_i^\top \boldsymbol{Y}_i \boldsymbol{w}_i + \boldsymbol{w}_i^\top \boldsymbol{Y}_i^\top \boldsymbol{Y}_i \boldsymbol{w}_i$$
$$= \boldsymbol{x}_i^\top \boldsymbol{x}_i + \boldsymbol{y}_i^\top \boldsymbol{y}_i + \boldsymbol{w}_i^\top (\boldsymbol{X}_i^\top \boldsymbol{X}_i + \boldsymbol{Y}_i^\top \boldsymbol{Y}_i) \boldsymbol{w}_i$$
$$- 2\boldsymbol{y}_i^\top \boldsymbol{Y}_i \boldsymbol{w}_i - 2\boldsymbol{x}_i^\top \boldsymbol{X}_i \boldsymbol{w}_i$$
$$\propto \left(\boldsymbol{w}_i - (\boldsymbol{X}_i^\top \boldsymbol{X}_i + \boldsymbol{Y}_i^\top \boldsymbol{Y}_i)^{-1}(\boldsymbol{X}_i^\top \boldsymbol{x}_i + \boldsymbol{Y}_i^\top \boldsymbol{y}_i)\right)^\top$$
$$(\boldsymbol{X}_i^\top \boldsymbol{X}_i + \boldsymbol{Y}_i^\top \boldsymbol{Y}_i)$$
$$\left(\boldsymbol{w}_i - (\boldsymbol{X}_i^\top \boldsymbol{X}_i + \boldsymbol{Y}_i^\top \boldsymbol{Y}_i)^{-1}(\boldsymbol{X}_i^\top \boldsymbol{x}_i + \boldsymbol{Y}_i^\top \boldsymbol{y}_i)\right). \quad (34)$$

Therefore, according to Eq. (6):

$$\mathbb{R}^{k \times k} \ni \boldsymbol{\Gamma}_i := \text{Cov}(\boldsymbol{w}_i \,|\, \boldsymbol{y}_i, \boldsymbol{x}_i) = (\boldsymbol{X}_i^\top \boldsymbol{X}_i + \boldsymbol{Y}_i^\top \boldsymbol{Y}_i)^{-1},$$
$$\mathbb{R}^k \ni \mathbb{E}[\boldsymbol{w}_i \,|\, \boldsymbol{y}_i, \boldsymbol{x}_i] = \boldsymbol{\Gamma}_i(\boldsymbol{X}_i^\top \boldsymbol{x}_i + \boldsymbol{Y}_i^\top \boldsymbol{y}_i),$$
$$\mathbb{P}(\boldsymbol{w}_i \,|\, \boldsymbol{y}_i, \boldsymbol{x}_i) = \mathcal{N}(\boldsymbol{w}_i; \mathbb{E}[\boldsymbol{w}_i \,|\, \boldsymbol{y}_i, \boldsymbol{x}_i], \boldsymbol{\Gamma}_i). \quad (35)$$

Note that, in stochastic linear reconstruction with direct sampling, we already have the weights obtained from the original LLE. Hence, we have the true mean of distribution of $\boldsymbol{w}_i$ which is the weight obtained from the original LLE, denoted by $\boldsymbol{w}_i^{\text{LLE}}$. Therefore, we can relax this distribution as:

$$\mathbb{P}(\boldsymbol{w}_i \,|\, \boldsymbol{y}_i, \boldsymbol{x}_i) = \mathcal{N}(\boldsymbol{w}_i; \boldsymbol{w}_i^{\text{LLE}}, \boldsymbol{\Gamma}_i). \quad (36)$$

The algorithm of stochastic linear reconstruction with direct sampling is Algorithm 2. Note that this direct sampling is less theoretically solid than the EM approach for stochastic linear reconstruction; however, it is simpler to implement and slightly faster.

V. LINEAR EMBEDDING

Using either EM or direct sampling, we found the weights, $\{\boldsymbol{w}_i := [w_{i1}, \ldots, w_{ik}]^\top\}_{i=1}^n$, stochastically for linear reconstruction in the high dimensional input space. Now we embed data in the low dimensional embedding space using the same weights as in the input space. This linear embedding, which is the same as linear embedding in deterministic LLE [3],

[4], can be formulated as the following optimization [18]:

$$\underset{\boldsymbol{Y}}{\text{minimize}} \quad \sum_{i=1}^n \left\|\boldsymbol{y}_i - \sum_{j=1}^n \breve{w}_{ij} \boldsymbol{y}_j\right\|_2^2,$$
$$\text{subject to} \quad \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i \boldsymbol{y}_i^\top = \boldsymbol{I}, \quad \sum_{i=1}^n \boldsymbol{y}_i = \boldsymbol{0}, \quad (37)$$

where the rows of $\mathbb{R}^{n \times p} \ni \boldsymbol{Y} := [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]^\top$ are the embedded data points (stacked row-wise), $\boldsymbol{y}_i \in \mathbb{R}^p$ is the i-th embedded data point, and $\breve{w}_{ij}$ is the weight obtained from the stochastic linear reconstruction if $\boldsymbol{x}_j$ is a neighbor of $\boldsymbol{x}_i$ and zero otherwise:

$$\breve{w}_{ij} := \begin{cases} w_{ij} & \text{if } \boldsymbol{x}_j \in k\text{NN}(\boldsymbol{x}_i), \\ 0 & \text{otherwise.} \end{cases} \quad (38)$$

The second constraint in Eq. (37) ensures the zero mean of embedded data points. The first and second constraints together satisfy having unit covariance for the embedded points.

Suppose $\mathbb{R}^n \ni \breve{\boldsymbol{w}}_i := [\breve{w}_{i1}, \ldots, \breve{w}_{in}]^\top$ and let $\mathbb{R}^n \ni \boldsymbol{1}_i := [0, \ldots, 1, \ldots, 0]^\top$ be the vector whose i-th element is one and other elements are zero. The objective function in Eq. (37) can be restated as:

$$\sum_{i=1}^n \left\|\boldsymbol{y}_i - \sum_{j=1}^n \breve{w}_{ij}\boldsymbol{y}_j\right\|_2^2 = \sum_{i=1}^n \|\boldsymbol{Y}^\top \boldsymbol{1}_i - \boldsymbol{Y}^\top \breve{\boldsymbol{w}}_i\|_2^2$$
$$= \|\boldsymbol{Y}^\top \boldsymbol{I} - \boldsymbol{Y}^\top \breve{\boldsymbol{W}}^\top\|_F^2 = \|\boldsymbol{Y}^\top (\boldsymbol{I} - \breve{\boldsymbol{W}})^\top\|_F^2$$
$$= \|\boldsymbol{Y}^\top (\boldsymbol{I} - \breve{\boldsymbol{W}})^\top\|_F^2 = \mathbf{tr}\left((\boldsymbol{I} - \breve{\boldsymbol{W}})\boldsymbol{Y}\boldsymbol{Y}^\top(\boldsymbol{I} - \breve{\boldsymbol{W}})^\top\right)$$
$$= \mathbf{tr}\left(\boldsymbol{Y}^\top(\boldsymbol{I} - \breve{\boldsymbol{W}})^\top(\boldsymbol{I} - \breve{\boldsymbol{W}})\boldsymbol{Y}\right) = \mathbf{tr}(\boldsymbol{Y}^\top \boldsymbol{M} \boldsymbol{Y}),$$

where $\|.\|_F$ denotes the Frobenius norm of matrix, the i-th row of $\mathbb{R}^{n \times n} \ni \breve{\boldsymbol{W}} := [\breve{\boldsymbol{w}}_1, \ldots, \breve{\boldsymbol{w}}_n]^\top$ includes the weights for the i-th data point and:

$$\mathbb{R}^{n \times n} \ni \boldsymbol{M} := (\boldsymbol{I} - \breve{\boldsymbol{W}})^\top(\boldsymbol{I} - \breve{\boldsymbol{W}}). \quad (39)$$
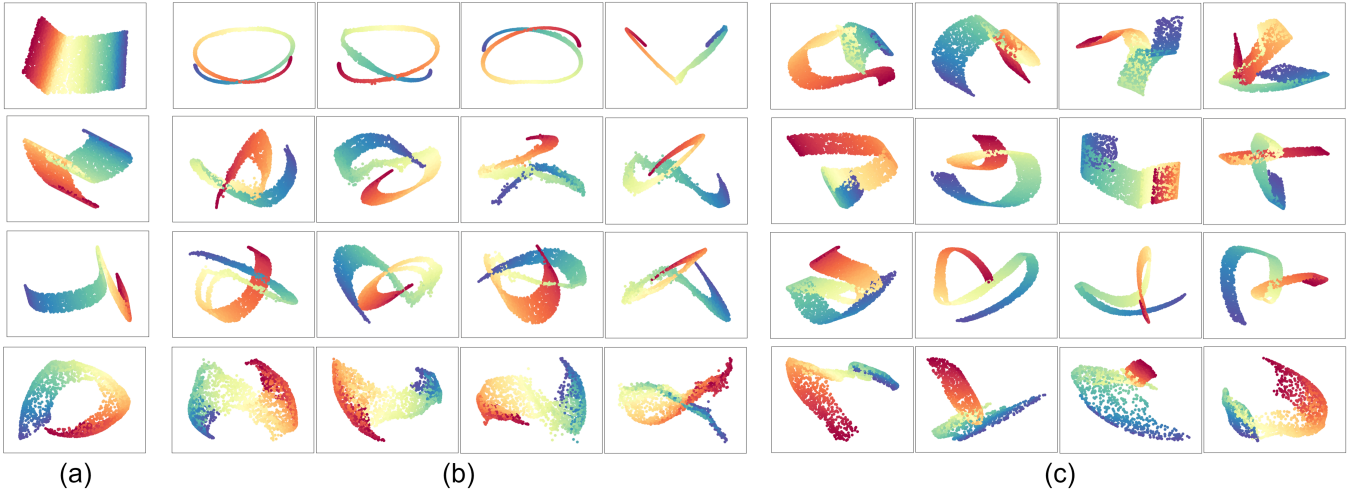
Fig. 3. (a) LLE unfolding and several unfolding generations of GLLE embedding with (b) EM algorithm and (c) direct sampling for the three nonlinear datasets. The first to fourth rows correspond to the S-curve, Swiss roll, Swiss roll with hole, and severed bowl, respectively.
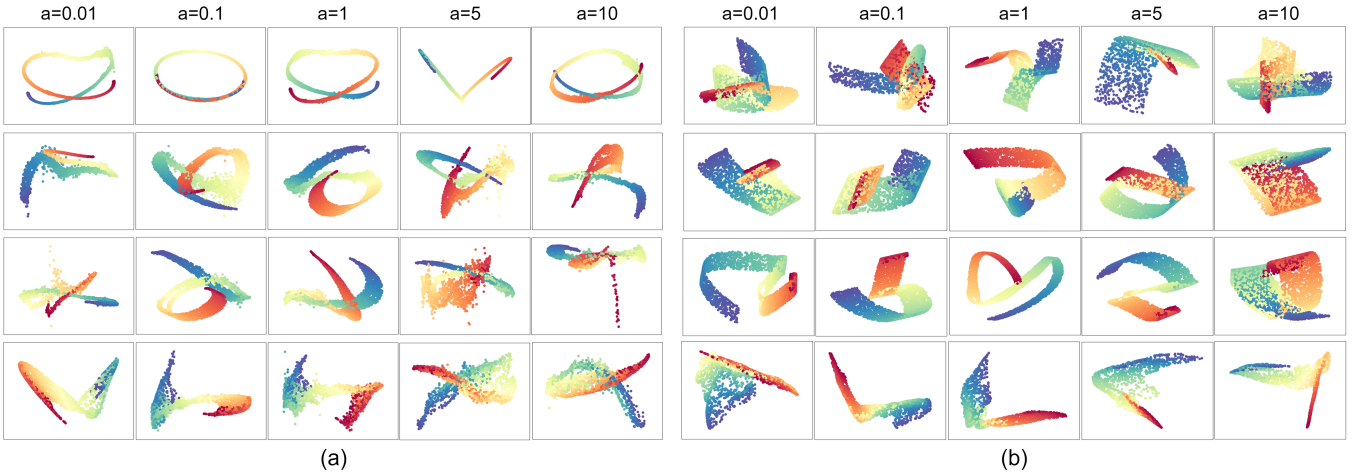

Fig. 4. The effect of covariance scaling on manifold unfolding by (a) GLLE with EM algorithm and (b) GLLE with direct sampling. The first to fourth rows correspond to S-curve, Swiss roll, Swiss roll with hole, and severed bowl, respectively. The columns correspond to different covariance scales, which are (a) $a\boldsymbol{\Omega}_i$ and (b) $a\boldsymbol{\Gamma}_i$ for $\forall i$ and $a \in \{0.01, 0.1, 1, 5, 10\}$.

Finally, Eq. (37) can be rewritten as:

$$\underset{\boldsymbol{Y}}{\text{minimize}} \quad \mathbf{tr}(\boldsymbol{Y}^\top \boldsymbol{M} \boldsymbol{Y}),$$
$$\text{subject to} \quad \frac{1}{n}\boldsymbol{Y}^\top \boldsymbol{Y} = \boldsymbol{I}, \quad \boldsymbol{Y}^\top \boldsymbol{1} = \boldsymbol{0}, \quad (40)$$

The second constraint is satisfied implicitly [18]. The Lagrangian for Eq. (40) is [21]:

$$\mathcal{L} = \mathbf{tr}(\boldsymbol{Y}^\top \boldsymbol{M} \boldsymbol{Y}) - \mathbf{tr}\big(\boldsymbol{\Lambda}^\top (\frac{1}{n}\boldsymbol{Y}^\top \boldsymbol{Y} - \boldsymbol{I})\big),$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal matrix including the Lagrange multipliers. Equating derivative of $\mathcal{L}$ to zero gives us:

$$\mathbb{R}^{n \times p} \ni \frac{\partial \mathcal{L}}{\partial \boldsymbol{Y}} = 2\boldsymbol{M}\boldsymbol{Y} - \frac{2}{n}\boldsymbol{Y}\boldsymbol{\Lambda} \overset{\text{set}}{=} \boldsymbol{0}$$
$$\implies \boldsymbol{M}\boldsymbol{Y} = \boldsymbol{Y}(\frac{1}{n}\boldsymbol{\Lambda}), \quad (41)$$

which is the eigenvalue problem for $\boldsymbol{M}$ [22]. Therefore, the columns of $\boldsymbol{Y}$ are the eigenvectors of $\boldsymbol{M}$ where eigenvalues

are the diagonal elements of $(1/n)\boldsymbol{\Lambda}$. As Eq. (40) is a minimization problem, the columns of $\boldsymbol{Y}$ should be sorted from the smallest to largest corresponding non-zero eigenvalues. Note that because of relation of $\boldsymbol{M}$ with Laplacian of kNN graph (see Eq. (39)), there is a zero eigenvalue whose eigenvector should be ignored.

VI. SIMULATIONS

In this section, we report the simulation results. The code for this paper and its simulations can be found in our Github repository[1].

A. Datasets

For evaluating the effectiveness of the proposed GLLE with EM algorithm and direct sampling and its comparison with original LLE embedding, we created four highly non-linear manifolds. The created synthetic datasets are S-curve,

[1]https://github.com/bghojogh/Generative-LLE

Swiss roll, Swiss roll with a hole inside it, and severed bowl; they are illustrated in Fig. 2. Each of these nonlinear datasets includes 5000 three-dimensional data points.

B. Manifold Unfolding Generations

Comparison of manifold unfolding using original LLE, GLLE with EM algorithm, and GLLE with direct sampling is shown in Fig. 3. For the simulations of this paper, we set $k = 10$ to consider ten neighbors in kNN graph. This figure depicts four different generations of the proposed GLLE algorithms for each dataset. As can be seen, the GLLE embeddings are closely related to LLE embeddings. The GLLE generations have correctly unfolded the manifold with acceptable precision. The different generations of GLLE are also related because their reconstruction weights are sampled from the same probability distributions. The embedding of GLLE with EM algorithm for S-curve has become narrow which is interpretable; this is because of the most informative direction of the S-curve which is the narrow S-shape from its side. In all different generations of GLLE algorithms, the relations of neighbor points have been preserved as expected.

C. Analyzing the Effect of Covariance Scaling on GLLE

Both the proposed GLLE versions work with the covariance matrix of reconstruction weights for every data point. We analyzed the impact of covariance scaling on the embedding generations of GLLE algorithms. Figure 4 illustrates several generations of GLLE algorithms with five different scales of covariance matrices of weights. We used the scaled covariance matrices $a\mathbf{\Omega}_i$ and $a\mathbf{\Gamma}_i$ with $\forall i \in \{1, \ldots, n\}$ and $a \in \{0.01, 0.1, 1, 5, 10\}$ for GLLE with EM algorithm and GLLE with direct sampling, respectively. This figure shows that the learned covariance matrices in GLLE with EM algorithm and the derived covariance matrices for GLLE with direct sampling work properly because the manifolds have been correctly unfolded. As this figure shows, both GLLE versions have acceptable robustness to different scales of covariance matrices; although, the correct scale, i.e. $a = 1$, often has the best performance as expected. Some large scales of covariance have resulted in high-variance and slightly less accurate embeddings, which is expected.

VII. CONCLUSION

In this paper, we proposed a generative version of LLE, named GLLE, where the linear reconstruction step in the input space is performed stochastically rather than deterministically. Two versions of stochastic linear reconstruction were proposed, one of which used EM and the other used direct sampling. The proposed GLLE was closely related to LLE, variational inference, factor analysis, and probabilistic PCA. Simulations on seevral datasets showed the effectiveness of the generated embeddings in comparison to LLE embedding.

ACKNOWLEDGMENT

REFERENCES

[1] B. Ghojogh, M. N. Samad, S. A. Mashhadi, T. Kapoor, W. Ali, F. Karray, and M. Crowley, "Feature selection and feature extraction in pattern analysis: A literature review," *arXiv preprint arXiv:1905.02845*, 2019.

[2] B. Ghojogh, "Data reduction algorithms in machine learning and data science," Ph.D. dissertation, University of Waterloo, 2021.

[3] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[4] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *Journal of machine learning research*, vol. 4, no. Jun, pp. 119–155, 2003.

[5] B. Fruchter, *Introduction to factor analysis.* Van Nostrand, 1954.

[6] D. Child, *The essentials of factor analysis.* Cassell Educational, 1990.

[7] S. Roweis, "EM algorithms for PCA and SPCA," *Advances in neural information processing systems*, vol. 10, pp. 626–632, 1997.

[8] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *International Conference on Learning Representations*, 2014.

[10] C. M. Bishop, *Pattern recognition and machine learning.* Springer, 2006.

[11] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[12] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 2008–2026, 2018.

[13] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "Factor analysis, probabilistic principal component analysis, variational inference, and variational autoencoder: Tutorial and survey," *arXiv preprint arXiv:2101.00734*, 2021.

[14] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[15] G. Harshvardhan, M. K. Gourisaria, M. Pandey, and S. S. Rautaray, "A comprehensive survey and analysis of generative models in machine learning," *Computer Science Review*, vol. 38, p. 100285, 2020.

[16] A. Oussidi and A. Elhassouny, "Deep generative models: Survey," in *2018 International Conference on Intelligent Systems and Computer Vision (ISCV).* IEEE, 2018, pp. 1–8.

[17] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (GANs): A survey," *IEEE Access*, vol. 7, pp. 36 322–36 333, 2019.

[18] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "Locally linear embedding and its variants: Tutorial and survey," *arXiv preprint arXiv:2011.10925*, 2020.

[19] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers," Technical Report CRG-TR-96-1, University of Toronto, Tech. Rep., 1996.

[20] J. Nocedal and S. Wright, *Numerical optimization.* Springer Science & Business Media, 2006.

[21] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[22] B. Ghojogh, F. Karray, and M. Crowley, "Eigenvalue and generalized eigenvalue problems: Tutorial," *arXiv preprint arXiv:1903.11240*, 2019.