Preventing Model Collapse in Gaussian Process Latent Variable Models

Ying Li [*1] **Zhidi Lin** [*2] **Feng Yin** [2] **Michael Minyi Zhang** [1]

Abstract

Gaussian process latent variable models (GPLVMs) are a versatile family of unsupervised learning models commonly used for dimensionality reduction. However, common challenges in modeling data with GPLVMs include inadequate kernel flexibility and improper selection of the projection noise, leading to a type of model collapse characterized by vague latent representations that do not reflect the underlying data structure. This paper addresses these issues by, first, theoretically examining the impact of projection variance on model collapse through the lens of a linear GPLVM. Second, we tackle model collapse due to inadequate kernel flexibility by integrating the spectral mixture (SM) kernel and a differentiable random Fourier feature (RFF) kernel approximation, which ensures computational scalability and efficiency through off-the-shelf automatic differentiation tools for learning the kernel hyperparameters, projection variance, and latent representations within the variational inference framework. The proposed GPLVM, named *advised*RFLVM, is evaluated across diverse datasets and consistently outperforms various salient competing models, including state-of-the-art variational autoencoders (VAEs) and other GPLVM variants, in terms of informative latent representations and missing data imputation.

1. Introduction

A latent variable model (LVM) represents each observed datum $\mathbf{y}_i \in \mathbb{R}^M$ using a low-dimensional latent variable $\mathbf{x}_i \in \mathbb{R}^Q$, where $Q \ll M$. As a classic tool in statistical analysis, LVMs unveil hidden structures within the data, providing valuable insights into intricate systems across various domains (Bishop, 2006), such as signal processing (Zarzoso et al., 2010) and economics (Aigner et al., 1984).

One of the critical aspects of LVM is the choice of mapping function from the latent variables to the observed variables. A series of early works assumed that the mapping is linear, as seen in factor analysis (Kim & Mueller, 1978), principal component analysis (PCA) (Pearson, 1901; Tipping & Bishop, 1999), and canonical correlation analysis (CCA) (Hotelling, 1936), among others. However, the linearity assumption limits the capacity of these models to capture complex, nonlinear patterns in the data, rendering them incapable of providing an optimal latent representation for complex data sets. To tackle this issue, more advanced methods like the variational autoencoder (VAE) (Kingma & Welling, 2019; 2013) utilizes neural networks, while the Gaussian process latent variable model (GPLVM) (Lawrence, 2005; Titsias & Lawrence, 2010) employs the Gaussian process (GP) (Rasmussen & Williams, 2006), as the nonlinear mapping modules in LVM, providing enhanced capacity in capturing nonlinear relationships.

GPLVMs benefit from the incorporation of the GP, which offers enhanced interpretability through explicit uncertainty calibration and the interpretable kernel functions (Theodoridis, 2020; Cheng et al., 2022). Additionally, the implicit regularization imposed by the GP prior prevents GPLVMs from severe overfitting (Lotfi et al., 2022; Wilson & Izmailov, 2020). Consequently, GPLVMs often achieve superior performance in practice, even with small sample sizes. Due to these favorable and unique properties, GPLVM has been applied to various applications, such as intrusion detection (Abolhasanzadeh, 2015), image recognition (Eleftheriadis et al., 2013; Li et al., 2017), human pose estimation (Ek et al., 2008), and image-text retrieval (Song et al., 2015).

Despite the popularity of GPLVM and the recent efforts dedicated to enhancing its learning and inference capabilities (Titsias & Lawrence, 2010; Gundersen et al., 2021; Ramchandran et al., 2021; de Souza et al., 2021; Lalchand et al., 2022; Zhang et al., 2023), the existing work still lacks an in-depth understanding of how to optimally learn a compact and informative latent representation using the

[*]Equal contribution. The authors are ordered alphabetically. [1]Department of Statistics & Actuarial Science, The University of Hong Kong, Hong Kong, China [2]School of Science & Engineering, The Chinese University of Hong Kong, Shenzhen, China. Correspondence to: Feng Yin <yinfeng@cuhk.edu.cn>, Michael Minyi Zhang <mzhang18@hku.hk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

GPLVM. This ambiguity hinders our ability to overcome "model collapse" (see Definition 2.1), which is characterized by learning vague latent representations with practical implementations. This paper elucidates the two key factors that lead to model collapse–the improper selection of model projection noise and inadequate kernel flexibility. To this end, we propose a new GPLVM that is immune to model collapse. Our contributions are:

- We provide a theoretical investigation of the impact that projection variance has on encouraging model collapse through the lens of linear GPLVMs. Our empirical validation further demonstrates the relevance of these analyses to general GPLVMs. These findings collectively emphasize the importance of learning the model projection variance.

- We propose a novel GPLVM that integrates a spectral mixture (SM) kernel (Wilson & Adams, 2013), capable of approximating arbitrary stationary kernels, to overcome model collapse arising from inadequate kernel flexibility. To reduce computational complexity and avoid introducing additional parameters like those in inducing point-based sparse GP methods (Titsias, 2009; Hensman et al., 2013), we leverage a differentiable random Fourier feature (RFF) approximation for the SM kernel (Jung et al., 2022; Lopez-Paz et al., 2014). This deliberate introduction of differentiability in the RFF approximation allows us to readily use modern off-the-shelf automatic differentiation tools (Paszke et al., 2019) to efficiently and scalably learn the kernel hyperparameters, projection variance, and latent representations of the proposed GPLVM within a variational inference framework (Bishop, 2006).

- Our proposed GPLVM is subjected to rigorous evaluation across diverse datasets, consistently outperforming various models, including the state-of-the-art (SOTA) VAEs and some representative GPLVM variants. Specifically, it excels in learning compact and informative latent representations, addressing the issues of model collapse in existing GPLVMs.

2. Preliminaries

Gaussian Process. The GP is a generalization of the Gaussian distribution defined across infinite index sets (Rasmussen & Williams, 2006), thereby enabling the specification of distribution over functions $f : \mathbb{R}^Q \mapsto \mathbb{R}$. A GP is fully characterized by its mean function $\mu(\mathbf{x})$, frequently set as zero, and its covariance function, a.k.a. kernel function, $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_{gp})$, where $\boldsymbol{\theta}_{gp}$ is a set of hyperparameters that needs to be tuned for model selection. According to the definition of GP, the function values $\mathbf{f} = \{f(\mathbf{x}_i)\}_{i=1}^N$ at any finite set of points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ follow a joint Gaussian distribution, i.e.,

$$\mathbf{f} \mid \mathbf{X} = \mathcal{N}(\mathbf{f} \mid \mathbf{0}, \mathbf{K}), \tag{1}$$

where $\mathbf{K}$ denotes the covariance matrix evaluated on the finite input $\mathbf{X}$ with $[\mathbf{K}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. Given the observed function values $\mathbf{f}$ at the input $\mathbf{X}$, the GP prediction distribution, $p(f(\boldsymbol{x}_*)|\mathbf{x}_*, \mathbf{f}, \mathbf{X})$, at any new input $\mathbf{x}_*$, is Gaussian, fully characterized by the posterior mean ξ and the posterior variance Ξ. Concretely,

$$\xi(\mathbf{x}_*) = \mathbf{K}_{\mathbf{x}_*,\mathbf{X}} \mathbf{K}^{-1} \mathbf{f}, \tag{2a}$$

$$\Xi(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{K}_{\mathbf{x}_*,\mathbf{X}} \mathbf{K}^{-1} \mathbf{K}_{\mathbf{x}_*,\mathbf{X}}^\top, \tag{2b}$$

where $\mathbf{K}_{\mathbf{x}_*,\mathbf{X}}$ is the cross covariance matrix evaluated on the new input $\mathbf{x}_*$ and the observed input $\mathbf{X}$.

Spectral Mixture Kernel. The behavior of a GP-distributed function is generally defined by the choice of the kernel function. However, subjectively selecting an appropriate kernel for complex applications is considerably challenging. By resorting to the fact that, according to Bochner's theorem, any stationary kernel and its spectral density are Fourier duals, we know that one type of popular kernel learning methods is to approximate the spectral density of the underlying stationary kernel (Bochner, 1934). In the spectral mixture (SM) kernel (Wilson & Adams, 2013), the underlying spectral density is approximated using a Gaussian mixture:

$$\begin{aligned} s_i(\mathbf{w}) &= \frac{\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2)) + \mathcal{N}(-\mathbf{w}|\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2))}{2}, \\ p_{\mathrm{sm}}(\mathbf{w}) &= \sum_{i=1}^m \alpha_i s_i(\mathbf{w}), \end{aligned} \tag{3}$$

where α_i is the mixture weight, $\boldsymbol{\mu}_i \in \mathbb{R}^Q$ and $\boldsymbol{\sigma}_i^2 \in \mathbb{R}^Q$ are the mean and variance of the i-th Gaussian density, m is the number of mixture components. Taking the inverse Fourier transform, we readily get the SM kernel, $k_{\mathrm{sm}}(\mathbf{x}, \mathbf{x}') =$

$$\sum_{i=1}^m \alpha_i \exp\left(-2\pi^2 \|\boldsymbol{\sigma}_i^\top (\mathbf{x} - \mathbf{x}')\|^2\right) \cos\left(2\pi \boldsymbol{\mu}_i^\top (\mathbf{x} - \mathbf{x}')\right),$$

where $\boldsymbol{\theta}_{\mathrm{sm}} = \{\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\}_{i=1}^m$ is the set of hyperparameters. Given that Gaussian mixture is dense, the SM kernel is guaranteed to be able to approximate any stationary kernel arbitrarily well (Wilson & Adams, 2013).

Gaussian Process Latent Variable Models. The GPLVM is a generative model where each observed datum $\mathbf{y}_i \in \mathbb{R}^M$ is generated through a noisy Gaussian process from a latent variable $\mathbf{x}_i \in \mathbb{R}^Q$ (Lawrence, 2005):

$$\mathbf{y}_i = f(\mathbf{x}_i) + \boldsymbol{v}_i, \quad \boldsymbol{v}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_M), \tag{4}$$

where $f(\cdot)$ follows a zero-mean GP prior, and σ^2 is the projection variance, which can be interpreted as information lost in dimensionality reduction. A standard normal density is conventionally assigned as the prior to the latent variable, $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_Q)$. In the case of having N observations

$\mathbf{Y} \in \mathbb{R}^{N \times M}$ from the GPLVM, the marginal likelihood after integrating out the latent GP, is expressed as:

$$p(\mathbf{Y} \mid \mathbf{X}) = \prod_{j=1}^{M} \mathcal{N}(\mathbf{y}_{:,j} \mid \mathbf{0},\ \mathbf{K} + \sigma^2 \mathbf{I}_N) \tag{5}$$

where $\mathbf{y}_{:,j} \in \mathbb{R}^N$ denotes the j-th column of $\mathbf{Y}$. Consequently, the maximum likelihood estimate (MLE) of the latent variables $\mathbf{X}$ can be obtained by solving the following optimization problem,

$$\hat{\mathbf{X}} = \max_{\mathbf{X}}\ L(\mathbf{X}) = \max_{\mathbf{X}}\ \log p(\mathbf{Y} \mid \mathbf{X}), \tag{6}$$

using e.g. gradient-based methods (Kingma & Ba, 2014).

In the context of GPLVM, the primary objective is to obtain a compact and informative latent representation of the observed data. Unlike the general definition of model collapse in machine learning models, which is primarily characterized by a gradual shift toward homogeneous output and increased deviations from accurate predictions (Bau et al., 2019), model collapse in GPLVM is closely tied to the effectiveness of latent variable inference, as outlined below:

Definition 2.1 (**Model Collapse**)**.** When the latent variables in GPLVMs become more homogeneous and/or their crucial feature details are sacrificed or distorted, we identify this phenomenon as model collapse.

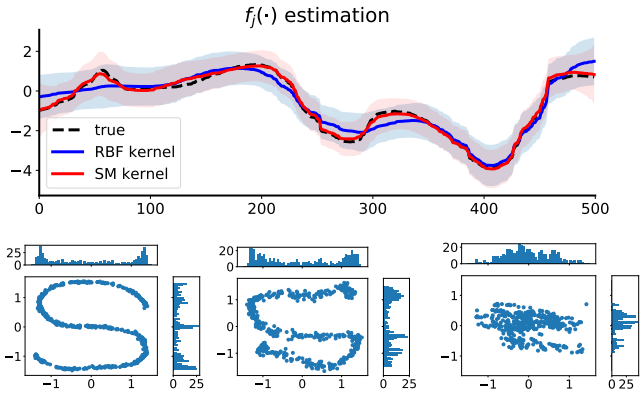
Definition 2.1 posits that two distinct manifestations of model collapse can be identified: distortion and homogeneity. Distortion occurs when the latent manifold, representing the underlying data structure, is warped or twisted, failing to accurately describe the underlying data structures. Homogeneity, on the other hand, manifests as a reduction in diversity among latent variables, resulting in a loss of crucial data features.

3. Causes of Model Collapse

In this section, we will elucidate that the distortion and homogeneity in the latent manifold are attributed to two crucial factors: improper selection of projection variance and inadequate kernel function flexibility. To further illustrate these concepts, Figs. 1b and 1c depict examples where the learned latent manifolds are distorted and homogeneous, respectively.

3.1. Projection Variance Matters

This subsection investigates the impact of projection variance on encouraging model collapse. To achieve this, we scrutinize the stationary points with respect to the latent variables $\mathbf{X}$, and establish their connection to the projection variance. However, the computation of the stationary points is intractable due to the non-convex and nonlinear nature of GPLVMs in general. In light of this, we alternatively seek the lens of the linear GPLVM by assuming that the kernel function used in the GPLVM is the inner product kernel, i.e., $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$. This simplified GPLVM is also known as the dual probabilistic principal component analysis (DPPCA) model (Lawrence, 2005). See more details in App. A.1. The main analyses are outlined below.



(a) Normal **(b)** Distortion **(c)** Homogeneity

Figure 1: Top: Latent function estimation using GPLVM with preliminary (—) or advanced/flexible kernels (—). **Bottom:** (**a**): 2-D S-shape latent manifold learned by the proposed *advised* RFLVM. (**b**): 2-D S-shape latent manifold learned by using a preliminary (RBF) kernel. (**c**): 2-D S-shape latent manifold learned without optimizing projection variance. (**a**)–(**c**) also show histograms in different dimensions of the learned latent manifold.

Theorem 3.1. *Given the maximization problem in Eq. (6), the stationary points, $\hat{\mathbf{X}}$, in the case of the linear GPLVM is:*

$$\hat{\mathbf{X}} = \mathbf{U}_Q \left(\mathbf{\Lambda}_Q - \sigma^2 \mathbf{I}_Q\right)^{1/2} \mathbf{R}, \tag{7}$$

where $\mathbf{U}_Q \triangleq [\mathbf{u}_1, \ldots, \mathbf{u}_Q] \in \mathbb{R}^{N \times Q}$ represents arbitrary eigenvectors of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$, $\mathbf{R} \in \mathbb{R}^{Q \times Q}$ is an arbitrary orthogonal matrix, and $\mathbf{\Lambda}_Q \in \mathbb{R}^{Q \times Q}$ is a diagonal matrix with:

$$[\mathbf{\Lambda}_Q]_{i,i} = \begin{cases} \lambda_i, & \text{the corresponding eigenvalue to } \mathbf{u}_i, \textbf{ or} \\ \sigma^2. \end{cases}$$

Proof. See App. A.2 or App. A in (Lawrence, 2005). □

Theorem 3.1 reveals that the stationary point, $\hat{\mathbf{X}}$, depends on the projection variance σ^2 and eigenvalues of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$. However, it remains unclear which specific value of σ^2 may trigger the model collapse. Our findings, succinctly summarized in the following propositions, provide additional insight into the impact of the σ^2 on the type of the stationary point and the cause of the model collapse.

Proposition 3.2. *In the case that σ^2 equals to its MLE estimator, $\hat{\sigma}^2$:*

$$\sigma^2 = \hat{\sigma}^2 = \frac{1}{N - Q'} \sum_{j=Q'+1}^{N} \lambda_j, \tag{8}$$

where Q' is the number of eigenvalues retained in $\mathbf{\Lambda}_Q$ from $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$, then the only stable maximum[1] *is the global optimum.*

Proof. See App. A.3. □

Proposition 3.2 suggests that adhering to the principle $\sigma^2 = \hat{\sigma}^2$ during the optimization of the log marginal likelihood (see Eq. (6) or Eq. (22)), it is expected to yield the global optimum, thereby mitigating the risk of the model collapse.

Proposition 3.3. *If $\sigma^2 \in (\lambda^o_{Q-q+1}, \lambda^o_{Q-q})$, $q = 1, \dots, Q-1$, where λ^o_q denotes the q-th largest eigenvalues of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$, then the only stable maximum is the local optimum, with the maximizer $\hat{\mathbf{X}}$ having q zero columns. In addition, when $q = Q$, $\sigma^2 > \lambda^o_1$, the only stable maximum occurs when $\hat{\mathbf{X}} = \mathbf{0}$ (i.e., homogeneity).*

If $\sigma^2 < \lambda^o_N$, the stationary points comprise a cluster of local minimum points, accompanied by the emergence of zero columns in the $\hat{\mathbf{X}}$.

Proof. See App. A.4. □

Proposition 3.3 implies that an improper choice of σ^2 can hinder the optimization process, preventing it from reaching the optimum and leading to a loss of information (homogeneity) in $\hat{\mathbf{X}}$, i.e., the undesirable model collapse.

The aforementioned findings in the linear GPLVM underscore the importance of learning the projection variance σ^2 and demonstrate how this learning can help mitigate the risk of model collapse. While it is challenging to generalize these results to the broader GPLVM framework due to the model's non-convexity and nonlinearity, they still offer valuable insights into the role of projection variance in preventing model collapse within general GPLVMs (see § 6.1).

3.2. Kernel Function Flexibility Matters

The occurrence of model collapse is closely linked to the choice of kernel function as well, as the kernel plays a key role in learning the underlying mapping $f(\mathbf{x})$ in GPLVMs. In particular, if the learned mapping function characterized by the GP posterior diverges from the underlying one, there is a significant possibility that the estimated latent manifold will become distorted or lose crucial feature details, resulting in the model collapse.

This phenomenon is depicted in Fig. 1, where it is evident that the limited flexibility of the preliminary kernels prevents them from adequately exploring the corresponding reproducing kernel Hilbert space (RKHS) to capture the structure of the underlying function $f(\mathbf{x})$ (Theodoridis, 2020). Consequently, using the preliminary (RBF) kernel can only roughly fit the underlying function, leading to learning a distorted latent manifold–refer to the top of Fig. 1 and the associated latent manifold estimation in Fig. 1b, where we can see the struggle to fit the model that exhibits short-term irregularities.

Conversely, employing a flexible kernel capable of approximating arbitrary kernels allows for thorough exploration of the kernel space, enabling the automatic discovery of the most suitable kernel to capture hidden and possibly complex data patterns and structures, such as periodicity and long tails (Wilson & Adams, 2013; Duvenaud, 2014). This enhances the capacity to effectively learn the underlying mapping functions and estimate an accurate latent manifold, as evidenced by the learned function using a flexible (SM) kernel in Fig. 1 (top sub-figure) and the latent manifold estimate in Fig. 1a.

In summary, Fig. 1 demonstrates the importance of kernel flexibility in GPLVMs for mitigating model collapse (distortion) in practice. In this paper, we will employ a kernel capable of approximating arbitrary stationary kernels, namely the SM kernel (Wilson & Adams, 2013). In the next section, we detail our proposed GPLVM that incorporates the SM kernel while learning projection variance to prevent model collapse.

4. Preventing Model Collapse

Integrating general GPLVM with the SM kernel poses two distinct challenges: 1) high computational costs and 2) intractable model learning (de Souza et al., 2021; Jung et al., 2022; Chang et al., 2023). Specifically, the computational complexity of training the GPLVM with the SM kernel scales as $\mathcal{O}(N^3)$ with N data points (Rasmussen & Williams, 2006), rendering it prohibitive in the context of big data. To tackle the scalability issue of GPLVM, one representative variational method presented by Titsias & Lawrence (2010) involves utilizing sparse GPs based on inducing points (Titsias, 2009). However, this variational method is computationally tractable only for limited preliminary kernel functions, such as the RBF kernel. Recent work has tried to enhance the scalability and flexibility of the GPLVM by using the stochastic variational inference approach proposed by Hensman et al. (2013) (Lalchand et al., 2022; de Souza et al., 2021; Ramchandran et al., 2021) . Despite these endeavors, the need to optimize additional inducing points still leads to increased computational burden and the risk of getting stuck in suboptimal solutions. Thus, despite the enhanced model capability, these models often face challenges in achieving their theoretical potential to address model collapse (see § 6).

To address the aforementioned issues, we resort to the varia-

[1] In this case, the stationary points comprise only saddle points and the global optimum; no local optimum exists.

tional inference technique (Jordan et al., 1999) and a random Fourier features (RFF) approximation (Jung et al., 2022; Rahimi & Recht, 2007), which will enable us to efficiently and scalably learn the SM kernel-embedded GPLVM without introducing extra parameters (inducing points) as required in sparse GP-based methods (Titsias & Lawrence, 2010; Lalchand et al., 2022). The vanilla RFF approximates any stationary kernel $k(\mathbf{x}, \mathbf{x}')$ using Monte Carlo integration (Rahimi & Recht, 2007), i.e.,

$$k(\mathbf{x}, \mathbf{x}') \approx \varphi(\mathbf{x})^\top \varphi(\mathbf{x}'),\ \varphi(\mathbf{x}) \triangleq \sqrt{\frac{2}{L}} \Big[\sin\big(2\pi \mathbf{w}_1^\top \mathbf{x}\big), \cos\big(2\pi \mathbf{w}_1^\top \mathbf{x}\big), \ldots, \sin\Big(2\pi \mathbf{w}_{\frac{L}{2}}^\top \mathbf{x}\Big), \cos\Big(2\pi \mathbf{w}_{\frac{L}{2}}^\top \mathbf{x}\Big)\Big] \tag{9}$$

where $\{\mathbf{w}_l\}_{l=1}^{L/2}$ are $L/2$ i.i.d. spectral points drawn from the density function $p(\mathbf{w})$ of the associated kernel function $k(\mathbf{x}, \mathbf{x}')$, where L is an positive, even, integer.

Leveraging the RFF approximation, we can obtain the following SM kernel-embedded GPLVM:

$$\mathbf{y}_{:,j} \sim \mathcal{N}(\mathbf{0},\ \varphi(\mathbf{X})\varphi(\mathbf{X})^\top + \sigma^2 \mathbf{I}_N),\ j = 1, \ldots, M, \tag{10a}$$

$$\mathbf{w}_l \sim p_{\text{sm}}(\mathbf{w}), \quad l = 1, \ldots, L/2, \tag{10b}$$

$$\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_Q), \quad i = 1, \ldots, N, \tag{10c}$$

ensuring both computational scalability and modeling flexibility[2]. The following subsections will further detail our proposed variational inference algorithm to manage the learning tractability and efficacy in addressing the model collapse.

4.1. Approximate Bayesian Inference

Given the SM kernel-embedded GPLVM defined in Eq. (10), we utilize the variational inference technique (Theodoridis, 2020) to learn the model hyperparameters $\boldsymbol{\theta} = [\boldsymbol{\theta}_{\text{sm}}, \sigma^2]$, aiming to mitigate the risk of model collapse. Specifically, we can immediately obtain the joint distribution of the GPLVM in Eq. (10) as

$$\begin{aligned} p(\mathbf{Y}, \mathbf{X}, \mathbf{W}) &= p(\mathbf{X})p(\mathbf{W})p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) \\ &= p(\mathbf{W}) \prod_{i=1}^{N} p(\mathbf{x}_i) \prod_{j=1}^{M} p(\mathbf{y}_{:,j}|\mathbf{X}, \mathbf{W}), \end{aligned} \tag{11}$$

where $p(\mathbf{W}) = \prod_{l=1}^{L/2} p_{\text{sm}}(\mathbf{w})$ is the joint distribution of the spectral points. The variational inference method involves constructing a variational lower bound $\mathcal{L}$ of the log marginal likelihood that has the Kullback–Leibler (KL) divergence from approximating the underlying posterior as its slack: $\log p(\mathbf{Y}) - \mathcal{L} = \text{KL}[q(\mathbf{X}, \mathbf{W}) \| p(\mathbf{X}, \mathbf{W}|\mathbf{Y})]$. By maximizing $\mathcal{L}$ w.r.t. $q(\cdot)$, we improve the quality of the approximation (Cao et al., 2023; Cheng et al., 2022).

[2]Similar to Gundersen et al. (2021), we consider $\mathbf{W}$ as part of the data-generating process. We then constrain its prior $p(\mathbf{W})$ to be Gaussian mixtures, thereby defining SM kernels. For a detailed interpretation of $\mathbf{W}$, see App. B.2.

For this purpose, we introduce the following variational distribution to approximate the posterior over all the latent variables, $\{\mathbf{W}, \mathbf{X}\}$:

$$q(\mathbf{X}, \mathbf{W}) \triangleq p(\mathbf{W})q(\mathbf{X}) = p(\mathbf{W}) \prod_{i=1}^{N} q(\mathbf{x}_i), \tag{12}$$

where $q(\mathbf{X}) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_i, \boldsymbol{S}_i)$, and $\boldsymbol{\mu}_i \in \mathbb{R}^Q, \boldsymbol{S}_i \in \mathbb{R}^{Q \times Q}$ are the associated free variational parameters. The variational distribution $q(\mathbf{W})$ is constrained to be the prior distribution, which is essentially equivalent to explicitly assuming that $q(\mathbf{W})$ is Gaussian mixtures. See App. B.2 for detailed discussions on this equivalence and other more complex variational distributions of $\mathbf{W}$. Consequently, the variational lower bound for simultaneous learning and inference is ready to be derived and summarized in the following theorem.

Theorem 4.1. *With the model joint distribution in Eq. (11) and the assumed variational distribution in Eq. (12), the evidence lower bound (ELBO), $\mathcal{L} = \mathbb{E}_{q(\mathbf{X},\mathbf{W})}[\log p(\mathbf{Y}, \mathbf{X}, \mathbf{W}) - \log q(\mathbf{X}, \mathbf{W})]$, for the joint learning and inference is*

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(\mathbf{X},\mathbf{W})} \left[\log \frac{p(\mathbf{W}) \prod_{i=1}^{N} p(\mathbf{x}_i) \prod_{j=1}^{M} p(\mathbf{y}_{:,j}|\mathbf{X}, \mathbf{W})}{p(\mathbf{W}) \prod_{i=1}^{N} q(\mathbf{x}_i)}\right] \\ &= \underbrace{\sum_{j=1}^{M} \mathbb{E}_{q(\mathbf{X},\mathbf{W})}[\log p(\mathbf{y}_{:,j}|\mathbf{X}, \mathbf{W})]}_{\textit{Term 1: data reconstruction}} - \underbrace{\sum_{i=1}^{N} \text{KL}(q(\mathbf{x}_i) \| p(\mathbf{x}_i))}_{\textit{Term 2: regularization}}. \end{aligned}$$

Here, the first term corresponds to the data reconstruction error, which encourages any latent variables $\mathbf{X}$ and $\mathbf{W}$ sampled from the variational distribution, $q(\mathbf{X}, \mathbf{W})$, to accurately reconstruct the observations/likelihood. The second term represents a regularization for $q(\mathbf{X})$, which discourages significant deviations of $q(\mathbf{X})$ from the prior $p(\mathbf{X})$.

For the evaluation of $\mathcal{L}$, the second term can be evaluated analytically due to the Gaussian nature of the distributions. The first term needs to be handled numerically with Monte Carlo estimation, i.e.,

$$\text{Term 1} = \sum_{j=1}^{M} \mathbb{E}_{q(\mathbf{X},\mathbf{W})}[\log p(\mathbf{y}_{:,j}|\mathbf{X}, \mathbf{W})] \tag{13a}$$

$$\approx \sum_{j=1}^{M} \frac{1}{I} \sum_{i=1}^{I} \log \mathcal{N}(\mathbf{y}_{:,j}|\mathbf{0}, \hat{\mathbf{K}}_{\text{sm}}^{(i)} + \sigma^2 \mathbf{I}_N), \tag{13b}$$

where I denotes the number of Monte Carlo samples drawn from $q(\mathbf{X})$ and $p(\mathbf{W})$, and $\hat{\mathbf{K}}_{\text{sm}}$ is the SM kernel matrix approximation constructed by the feature map $\varphi(\cdot)$. See App. B.1 for more computational details.

Note that in Eq. (10b), we need to sample $\mathbf{w}_l$ from a Gaussian mixture, involving that first generates an index i from the discrete probability distribution, $P(i) =$

$\alpha_i/\sum_{j=1}^m \alpha_j, i = 1,\ldots,m$, and then draws sample $\mathbf{w}_l$ from $s_i(\mathbf{w})$. However, due to the difficulty of reparameterizing the discrete distribution over mixture weights (Graves, 2016), maximizing the ELBO w.r.t. the weights α_i using modern off-the-shelf automatic differentiation tools (e.g., PyTorch (Paszke et al., 2019)) becomes challenging. To this end, we similarly leverage a differentiable RFF feature map construction approach developed for GP regression models by Jung et al. (2022) to ensure inherent differentiability w.r.t. the mixture weights.

4.2. Differentiable RFF Approximation for SM Kernel

Rather than directly sampling from the Gaussian mixture, we first apply the vanilla RFF to get the corresponding feature map $\varphi_i(\mathbf{x}) \triangleq \sqrt{\alpha_i} \cdot \varphi(\mathbf{x}; \{\mathbf{w}_l^{(i)}\}_{l=1}^{L/2})$, $i = 1,\ldots,m$, for each mixture component, where the reparametrization trick (Kingma & Welling, 2019) is employed to sample $\mathbf{w}_l^{(i)}$ from $s_i(\mathbf{w})$. Subsequently, the stacking of m feature maps yields the ultimate new RFF approximation for the SM kernel, denoted as $\phi(\mathbf{x})$, i.e.,

$$\phi(\mathbf{x}) = \left[\varphi_1(\mathbf{x})^\top, \varphi_2(\mathbf{x})^\top, \ldots, \varphi_m(\mathbf{x})^\top\right]^\top \in \mathbb{R}^{mL\times 1}. \quad (14)$$

It can be shown that $\phi(\mathbf{x})^\top \phi(\mathbf{x})$ is an unbiased estimator of the SM kernel characterized by the hyperparameters $\boldsymbol{\theta}_{\text{sm}} = \{\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\}_{i=1}^m$. The result is succinctly encapsulated in the following proposition (Lopez-Paz et al., 2014).

Proposition 4.2. *Let* $\mathbf{W} = \{\mathbf{w}_1^{(i)}, \mathbf{w}_2^{(i)}, \ldots, \mathbf{w}_{L/2}^{(i)}\}_{i=1}^m$ *be the spectral points sampled from the distribution* $p(\mathbf{W}) = \prod_{i=1}^m \prod_{l=1}^{L/2} s_i(\mathbf{w})$ *using the reparameterization trick (Kingma & Welling, 2019). With the RFF feature map constructed in Eq. (14), given any inputs* $\mathbf{x}$ *and* $\mathbf{x}'$, $\phi(\mathbf{x})^\top \phi(\mathbf{x}')$ *is an unbiased estimator of* $k_{\text{sm}}(\mathbf{x}, \mathbf{x}')$ *with the hyperparameters* $\boldsymbol{\theta}_{\text{sm}}$*, i.e.,*

$$\mathbb{E}_{p(\mathbf{W})}\left[\phi(\mathbf{x})^\top \phi(\mathbf{x}')\right] = k_{\text{sm}}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_{\text{sm}}) \quad (15)$$

Proof. See App. C.1 □

In fact, given inputs $\mathbf{X}$ and the new feature map defined in Eq. (14), we can further characterize the approximation error bound for the constructed SM kernel matrix approximation, $\hat{\mathbf{K}}_{\text{sm}} = \Phi_{\text{sm}}(\mathbf{X})\Phi_{\text{sm}}(\mathbf{X})^\top$, where the random feature matrix $\Phi_{\text{sm}}(\mathbf{X}) = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]^\top \in \mathbb{R}^{N\times mL}$ (Jung et al., 2022; Lopez-Paz et al., 2014).

Theorem 4.3. *For all small* $\epsilon > 0$*, the approximation error between the underlying SM kernel matrix* $\mathbf{K}_{\text{sm}}$ *and its RFF approximation* $\hat{\mathbf{K}}_{\text{sm}}$ *is characterized by*

$$P\left(\left\|\hat{\mathbf{K}}_{\text{sm}} - \mathbf{K}_{\text{sm}}\right\|_2 \geq \epsilon\right) \leq N \exp\left(\frac{-3\epsilon^2 L}{2Na\left(6\left\|\mathbf{K}_{\text{sm}}\right\|_2 + 3Na\sqrt{m} + 8\epsilon\right)}\right), \quad (16)$$

where $a = \sqrt{\sum_{i=1}^m \alpha_i^2}$ *and* $\|\cdot\|_2$ *denotes the matrix spectral norm.*

Algorithm 1: *advised*RFLVM: Auto-Differentiable Variational Inference for SM-Embedded RFLVMs

Input: Dataset $\mathbf{Y}$; Initialized model hyperparameters $\boldsymbol{\theta}$ and variational parameters $\boldsymbol{\zeta}$.

while *iterations not terminated* **do**
- Sample $\mathbf{X}$ from $q(\mathbf{X}) = \prod_{i=1}^N \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_i, \boldsymbol{S}_i)$ using the reparameterization trick
- Sample $\mathbf{W}$ from $p(\mathbf{W}) = \prod_{i=1}^m \prod_{l=1}^{L/2} s_i(\mathbf{w})$ using the reparameterization trick
- Construct $\Phi_{\text{sm}}(\mathbf{X})$ using the sampled $\mathbf{X}$ and $\mathbf{W}$
- Evaluate Term 1 of $\mathcal{L}$ through Eq. (13)
- Evaluate Term 2 of $\mathcal{L}$ analytically
- Maximize $\mathcal{L}$ and update $\boldsymbol{\theta}$, $\boldsymbol{\zeta}$ using Adam

Output: $\boldsymbol{\theta}$, $\boldsymbol{\zeta}$.

Proof. See App. C.2 □

Beyond the theoretical guarantees of the approximation, the new feature map in Eq. (14) offers a crucial advantage–it renders the variational lower bound $\mathcal{L}$ differentiable w.r.t. mixture weights α_i, leading to the straightforward applicability of the automatic differentiation tools for hyperparameter optimization. Leveraging the new feature map, we can apply gradient-based methods (e.g., Adam (Kingma & Ba, 2014)) to maximize $\mathcal{L}$ w.r.t. model hyperparamters $\boldsymbol{\theta}$ and the variational parameters $\boldsymbol{\zeta} = \{\boldsymbol{\mu}_i, \boldsymbol{S}_i\}_{i=1}^N$. The pseudocode summarized in Algorithm 1 outlines the implementation of the proposed method, called auto-differentiable variational inference for SM-embedded RFF-LVM, abbreviated as *advised*RFLVM. It is noteworthy that for scenarios where $N \gg mL$, the computational complexity per iteration of *advised*RFLVM scales as $\mathcal{O}(N(mL)^2)$, as elaborated in App. B.1. Notably, this computational complexity aligns with that of the inducing point-based sparse GP method (Titsias & Lawrence, 2010). However, *advised*RFLVM enhances the capacity of the GPLVM and mitigates the need for optimizing the inducing points, resulting in a lightweight optimization problem while alleviating the model collapse.

5. Related Work

We have already described the main differences between our method and inducing points-based methods throughout the paper, e.g., in § 4. Below we briefly introduce other related work on latent variable modeling and refer the reader to App. D for more details.

VAEs. Variational autoencoders (VAEs) (Kingma & Welling, 2013) skillfully integrate LVMs typically modeled by neural networks with variational inference (Bishop, 2006), empowering the model to generate novel data. Unfortunately, despite the considerable success demonstrated by VAEs in generative tasks (Kingma & Welling, 2013; Zhao

et al., 2020; Nakagawa et al., 2023; Tran et al., 2023), they struggle to capture the underlying compact and informative latent representations of the observed data, resulting in the well-known posterior collapse issue (Menon et al., 2022; Wang & Liu, 2022; Lucas et al., 2019; Razavi et al., 2019), a facet of model collapse (see App. D). This phenomenon is partially attributed to the overfitting, stemming from optimizing a large number of parameters in the encoder of VAE, leading to homogeneous latent spaces (Bowman et al., 2016; Sønderby et al., 2016; Zhu et al., 2023).

RFLVMs. In addition to inducing points-based GPLVMs, the random feature latent variable model (RFLVM) adopts the RFF approximation of the kernel function as a variant of the GPLVM and leverages a Dirichlet process (DP) mixture of Gaussians to learn the associated spectral density of the kernel function (Rahimi & Recht, 2007; Oliva et al., 2016; Gundersen et al., 2021; Zhang et al., 2023). Despite the capacity to approximate arbitrary stationary kernels, the effectiveness in addressing model collapse in the RFLVM might be compromised by the "rich-get-richer" property inherent in the DP mixture prior (Gundersen et al., 2021), which places a strong assumption regarding the data generation process (Poux-Médard et al., 2023). A comprehensive comparison between our *advised*RFLVM and the SOTA models can be found in Table 3, App. D.

6. Experiments

We showcase the impact of the projection variance and kernel flexibility on model collapse in § 6.1 and § 6.2. In § 6.3 and § 6.4, we further corroborate the superior performance of *advised*RFLVM in latent representation learning on various real-world datasets. More experimental details can be found in App. E, and the code is publicly available at https://github.com/zhidilin/advisedGPLVM.

6.1. Projection Variance Matters

To evaluate the impact of the projection variance in general GPLVM, we apply the *advised*RFLVM on the MNIST dataset (LeCun, 1998). We quantify the degree of model collapse under two configurations of σ^2: learned and fixed. The degree of the model collapse is evaluated by counting the number of zero-columns in the learned latent variable $\hat{\mathbf{X}}$ and measuring its K-nearest neighbors (KNN) classification accuracy. Detailed results are depicted in Fig. 2.

On the left-hand side of Fig. 2, it is observed that, when σ^2 is fixed, the latent variable learned by the *advised*RFLVM rapidly collapses to zero as the value of σ^2 increases. This observation aligns with the findings in the linear GPLVM (see Proposition 3.3). Additionally, the inferior performance of the KNN accuracy depicted on the right-hand side of Fig. 2 illustrates that, without learning σ^2, the proposed *advised*RFLVM tends to recover a vague and uninformative

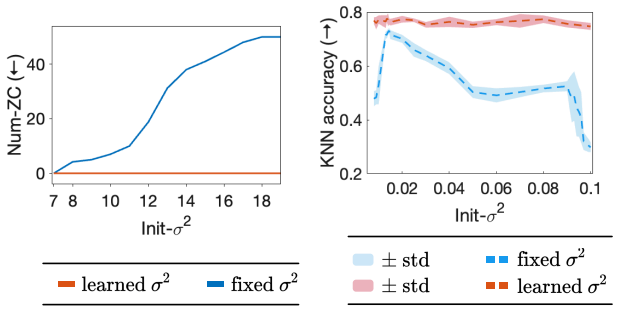


Figure 2: Left: The number of zero-columns (short as Num-ZC) in the latent variable $\mathbf{X}$ versus the initialization value of σ^2 (defined as Init-σ^2). **Right**: KNN classification accuracy against Init-σ^2. Standard deviation is calculated over five experiments.

latent representation. In stark contrast, *advised*RFLVM with a learned σ^2 effectively mitigates the risk of model collapse, irrespective of the initialization value of σ^2 or the metric employed. This supports our hypothesis regarding the importance of learning σ^2 to prevent model collapse in general GPLVMs.

6.2. S-shaped Latent Manifold Learning

Next, we demonstrate the importance of kernel flexibility in preventing model collapse, utilizing two synthetic datasets, each consisting of $N{=}500$ observations with $M{=}100$ dimensions. Both datasets are generated from a GPLVM with a two-dimensional (2-D) latent S-shaped manifold, but employing distinct kernel configurations. One employs a basic RBF kernel, while the other utilizes a more complex combination of an RBF kernel and a periodic kernel (Rasmussen & Williams, 2006). We compare our *advised*RFLVM with three GPLVM variants: BGPLVM (Titsias & Lawrence, 2010), GPLVM-SVI (Lalchand et al., 2022), and RFLVM (Zhang et al., 2023; Gundersen et al., 2021). In the case of BGPLVM and GPLVM-SVI, the default setting (see App. E) is used except that the number of inducing points is selected from the set $\{6, 10, 20, 30, 60, 120\}$, which yields the best inference performance.

Figure 3 reports the results for the S-shaped manifold learning, where the coefficient of determination (R^2 score) (Chicco et al., 2021) is used to quantify the "closeness" between the inferred manifold (after post-affine transformation) and the ground truth manifold. The results indicate that *advised*RFLVM and RFLVM consistently outperform BGPLVM and GPLVM-SVI in both synthetic datasets. It is obvious that GPLVM-SVI exhibits the worst performance, and BGPLVM shows fluctuated performance, although, in some realizations, they can reasonably estimate the shape of $\mathbf{X}$ (see the left illustration in Fig. 3). The fluctuated performance of BGPLVM and GPLVM-SVI suggests that optimizing the additional inducing points (variational parameters) can complicate the learning process and incur such instability.

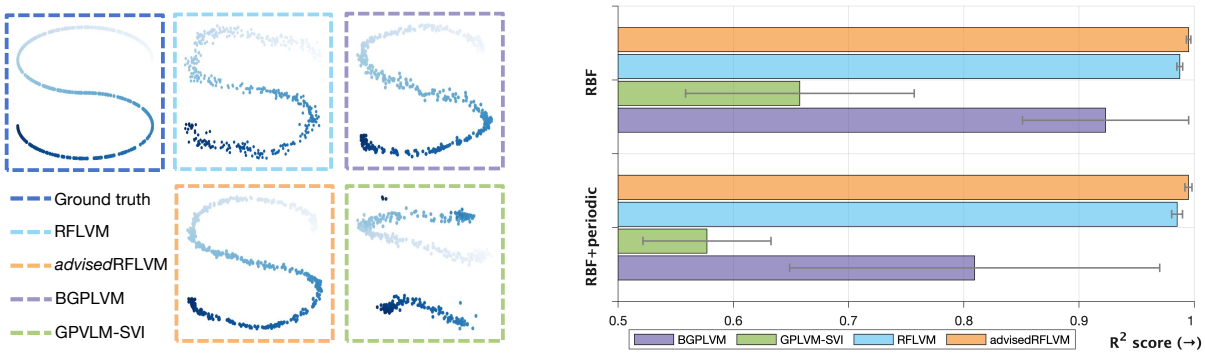


Figure 3: Left: Learned latent manifold in "RBF+periodic" dataset. **Right**: R^2 score performance over different models in two datasets.

Table 1: Classification accuracy evaluated by fitting a KNN classifier $(k = 1)$ with five-fold cross-validation. Mean and standard deviation are computed over five experiments, and the top performance is in bold.

DATASET	PCA	LDA	Isomap	HPF	BGPLVM	GPLVM-SVI
BRIDGES	0.841 ± 0.007	0.668 ± 0.053	0.797 ± 0.025	0.544 ± 0.109	0.818 ± 0.037	0.796 ± 0.019
CIFAR-10	0.267 ± 0.002	0.227 ± 0.006	0.272 ± 0.006	0.208 ± 0.006	0.271 ± 0.014	0.251 ± 0.012
MNIST	0.365 ± 0.012	0.233 ± 0.026	0.444 ± 0.021	0.314 ± 0.040	0.567 ± 0.033	0.344 ± 0.054
MONTREAL	0.678 ± 0.013	0.602 ± 0.028	0.709 ± 0.005	0.618 ± 0.001	0.725 ± 0.012	0.676 ± 0.010
NEWSGROUPS	0.392 ± 0.005	0.391 ± 0.018	0.397 ± 0.010	0.334 ± 0.019	0.385 ± 0.010	0.378 ± 0.018
YALE	0.543 ± 0.008	0.338 ± 0.023	0.588 ± 0.017	0.511 ± 0.019	0.553 ± 0.036	0.521 ± 0.015
DATASET	VAE	NBVAE	DCA	CVQ-VAE	RFLVM	*advised*RFLVM
BRIDGES	0.751 ± 0.016	0.758 ± 0.038	0.702 ± 0.036	0.688 ± 0.013	0.846 ± 0.039	$\mathbf{0.846 \pm 0.015}$
CIFAR-10	0.266 ± 0.002	0.259 ± 0.005	0.255 ± 0.019	0.224 ± 0.012	0.284 ± 0.103	$\mathbf{0.290 \pm 0.006}$
MNIST	0.643 ± 0.021	0.281 ± 0.012	0.171 ± 0.075	0.128 ± 0.005	0.602 ± 0.055	$\mathbf{0.795 \pm 0.015}$
MONTREAL	0.668 ± 0.012	0.716 ± 0.009	0.685 ± 0.716	0.646 ± 0.003	0.769 ± 0.010	$\mathbf{0.789 \pm 0.013}$
NEWSGROUPS	0.385 ± 0.002	0.398 ± 0.010	0.399 ± 0.034	0.356 ± 0.019	0.413 ± 0.009	$\mathbf{0.418 \pm 0.007}$
YALE	0.611 ± 0.020	0.456 ± 0.046	0.284 ± 0.054	0.338 ± 0.002	0.653 ± 0.067	$\mathbf{0.765 \pm 0.010}$

The performance gain of the *advised*RFLVM and RFLVM can be attributed to the kernel flexibility, which is particularly evident when the dataset is generated from the underlying GPLVM with a hybrid of RBF kernel and periodic kernel. This validates the crucial role of kernel function flexibility in preventing model collapse. Nevertheless, *advised*RFLVM consistently outperforms the RFLVM, although RFLVM theoretically is capable of approximating arbitrary stationary kernels as well. This discrepancy may stem from the biased assumption of DP priors for the spectral densities in RFLVM (Zhang et al., 2023). Such bias can lead to unfair exposure for the density weights, resulting in only a few effective densities and a degenerated approximation capacity (Gundersen et al., 2021). Moreover, the RFLVM is based on MCMC sampling which may be inferior in this setting to the *advised*RFLVM, which optimizes the ELBO in terms of inference efficiency.

6.3. Real Dataset Evaluation

This subsection further demonstrates the ability of *advised*RFLVM to capture the latent space on multiple real-world datasets (see Table 1), where the dataset sizes of MNIST and CIFAR are reduced for accommodating the high complexity of RFLVM (see App. E.1 for further details). For each dataset, we hold the labels and employ them to evaluate the estimated latent space using KNN classifier with five-fold cross-validation. In addition to the GPLVM variants used in § 6.2, we also encompass various recent VAEs (Kingma & Welling, 2019; Zhao et al., 2020; Eraslan et al., 2019; Zheng & Vedaldi, 2023) and classic dimensionality reduction methods. The KNN classification accuracy results for all the competing methods are presented in Table 1.

The results demonstrate that *advised*RFLVM consistently achieves the highest KNN accuracy across all datasets. This suggests that the latent variables estimated by *advised*RFLVM are more informative compared to the other methods. The four classic methods, PCA (Wold et al., 1987; Pearson, 1901), hierarchical Poisson factorization (HPF) (Gopalan et al., 2015), latent Dirichlet allocation (LDA) (Blei et al., 2003), and Isomap (Balasubramanian & Schwartz, 2002) showing inferior performance, are primary attributed to their limited model flexibility.

For the VAE models, despite their impressive approximation capabilities through neural network-based decoders and

Table 2: Missing data imputation on the MNIST and BRENDAN datasets.

DATASET	METRIC	VAE				BGPLVM				RFLVM				*advised*RFLVM			
		0%	10%	30%	60%	0%	10%	30%	60%	0%	10%	30%	60%	0%	10%	30%	60%
MNIST	KNN ACC (↑)	0.715	0.689	0.660	0.585	0.603	0.598	0.541	0.476	0.602	0.391	0.345	0.273	**0.806**	**0.802**	**0.777**	**0.636**
	TEST MSE (↓)	0.035	0.038	0.045	**0.068**	0.048	0.040	0.057	0.098	0.066	0.067	0.070	0.120	**0.025**	**0.028**	**0.039**	0.068
BRENDAN	TEST MSE (↓)	0.005	0.009	**0.043**	**0.150**	0.006	0.041	0.087	0.197	0.010	0.015	0.049	0.153	**0.003**	**0.009**	0.045	0.152

encoders (Kingma & Welling, 2019), they often fall short in their latent space learning performance. This is because optimizing numerous neural network parameters can result in overfitting, rendering these deterministic neural networks directed toward wrong latent spaces. In contrast, GPLVM variants prevent the need for neural network parameter optimization. More importantly, the inherent regularization imposed by the GP prior mitigates overfitting and thus enhances the generalization capability for latent space learning (Wilson & Izmailov, 2020). These lead to GPLVM-based models being expected to attain higher KNN accuracy. Nevertheless, the results in Table 1 show that BGPLVM and GPLVM-SVI can only attain comparable performance compared to the PCA. This mainly attributed to the inherently inadequate kernel flexibility and the additional optimization burden of the variational parameters. RFLVM consistently exhibits a slightly inferior performance compared to *advised*RFLVM, primarily due to the unfair exposure of density weights and the inefficient and unscalable MCMC inference algorithm mentioned in § 6.2 and § 5. We also conducted additional simulations on larger datasets. The results, presented in Appendix E.4.4, emphasize the superiority of *advised*RFLVM over state-of-the-art variants regardless of the dataset size.

6.4. Missing Data Imputation

This subsection further evaluates the performance of *advised*RFLVM in the task of imputing missing data on two image datasets, namely MNIST and BRENDAN (Roweis & Saul, 2000). Specifically, we randomly hold out a certain proportion (0%, 10%, 30%, and 60%) of the elements in the observed data matrix, $\mathbf{Y}$, and subsequently we utilize *advised*RFLVM to estimate latent variables $\mathbf{X}$ from the incomplete dataset (denoted as $\mathbf{Y}_{obs}$). We then impute the missing values $\mathbf{Y}_{miss}$ by their posterior mean $\hat{\mathbf{Y}}_{miss} = \mathbb{E}[\mathbf{Y}_{miss} \mid \mathbf{X}, \mathbf{Y}_{obs}, -]$. The imputation performance is evaluated through the mean square error (MSE) between $\hat{\mathbf{Y}}_{miss}$ and the ground-truth $\mathbf{Y}_{miss}$. Additionally, KNN classification accuracy is reported for the MNIST dataset to illustrate the latent representation learning results. Table 2 presents the performance of the *advised*RFLVM against competing methods. The results indicate that *advised*RFLVM outperforms most competitors in reconstructing observations and recovering latent representations, regardless of the proportion of missing data. Despite VAE exhibiting reconstruction capabilities comparable to *advised*RFLVM, it still lags behind in recovering informative latent variables due to its potential overfiting and inherent posterior collapse issues (Wang & Liu, 2022). More details about the reconstruction performance of *advised*RFLVM are provided in App. E.4.2, showing its superior ability to restore missing pixels.

7. Conclusions

We have introduced our novel *advised*RFLVM to address model collapse due to inadequate kernel flexibility and inappropriate projection variance selection in GPLVMs. By integrating the SM kernel and the differentiable RFF approximation, our *advised*RFLVM not only enhances model flexibility but also enables the use of modern automatic differentiation tools for optimizing essential parameters, including the projection variance within the variational inference framework. Empirical results across diverse datasets corroborate the superiority of our *advised*RFLVM in learning compact and informative latent representations, highlighting the importance of learning projection variance and kernel flexibility in mitigating model collapse. Furthermore, our model outperforms various state-of-the-art latent variable models, including VAEs and other GPLVM variants. In future work, we are focusing on how to further enhance the variational inference algorithm presented in this paper. We hope that, through our endeavors, we may scale up our LVM for scenarios with massive data sets as an efficient alternative to the resource-intensive deep learning models.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments that improved the quality of the paper. The work of Feng Yin was supported by the NSFC under Grant No. 62271433, and in part by the Shenzhen Science and Technology Program under Grant No. JCYJ20220530143806016. The work of Michael Minyi Zhang was supported by the HKU-URC Seed Fund for Basic Research for New Staff.

Impact Statement

This work introduces a novel probabilistic latent variable model tailored to effectively capture the underlying struc-

tures of the observed data, which allows us to provide informative but concise foundational knowledge for analyzing highly complex tasks, such as the analysis of social issues, research on human behavior, and exploration of cognitive mechanisms. Technically, this work, conducting theoretical analyses on the impact of the projection variance on model collapse, will strengthen the understanding of broader researchers and engineers on the "default" learning of the projection variance. We also carefully examine the impact of kernel flexibility, and all these rigorous examinations of the potential reasons for model collapse enhance model interpretability, which is crucial for safety-critical systems such as autonomous driving and intelligent healthcare.

Limitations and future works. Our model faces limitations in handling out-of-distribution data, which requires explicitly learning an encoding function from observable data points into a latent representation. One potential solution to address this is to assume a $\mathbf{Y}$-dependent parametric variational distribution of latent variables, $q(\mathbf{X}|\mathbf{Y})$, where the parameters of the distribution are modeled by an encoder network that takes the observation $\mathbf{Y}$ as input. Consequently, upon completion of the training process, the encoder network can be employed to infer the latent variables of the out-of-distribution data. Another limitation is that despite the reduction in the complexities (linear with N), the practical training time of our method may not be endurable for massive datasets.

References

Abolhasanzadeh, B. Gaussian process latent variable model for dimensionality reduction in intrusion detection. In *2015 23rd Iranian Conference on Electrical Engineering*, pp. 674–678. IEEE, 2015.

Aigner, D. J., Hsiao, C., Kapteyn, A., and Wansbeek, T. Latent variable models in econometrics. *Handbook of econometrics*, 2:1321–1393, 1984.

Balasubramanian, M. and Schwartz, E. L. The Isomap algorithm and topological stability. *Science*, 295(5552): 7–7, 2002.

Bau, D., Zhu, J.-Y., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., and Torralba, A. Seeing what a GAN cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4502–4511, 2019.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3 (Jan):993–1022, 2003.

Bochner, S. A theorem on Fourier-Stieltjes integrals. *Bulletin of the American Mathematical Society*, 40(4):271–276, 1934.

Bowman, S., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, 2016.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

Cao, J., Kang, M., Jimenez, F., Sang, H., Schaefer, F. T., and Katzfuss, M. Variational sparse inverse Cholesky approximation for latent Gaussian processes via double Kullback-Leibler minimization. In *International Conference on Machine Learning*, pp. 3559–3576. PMLR, 2023.

Chang, P. E., Verma, P., John, S., Solin, A., and Khan, M. E. Memory-based dual Gaussian processes for sequential learning. In *International Conference on Machine Learning*, pp. 4035–4054. PMLR, 2023.

Cheng, L., Yin, F., Theodoridis, S., Chatzis, S., and Chang, T.-H. Rethinking Bayesian learning for data analysis: The art of prior and inference in sparsity-aware modeling. *IEEE Signal Processing Magazine*, 39(6):18–52, 2022.

Chicco, D., Warrens, M. J., and Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7: e623, 2021.

de Souza, D., Mesquita, D., Gomes, J. P., and Mattos, C. L. Learning GPLVM with arbitrary kernels using the unscented transformation. In *International Conference on Artificial Intelligence and Statistics*, pp. 451–459. PMLR, 2021.

Duvenaud, D. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

Ek, C. H., Torr, P. H. S., and Lawrence, N. D. Gaussian process latent variable models for human pose estimation. In *Machine Learning for Multimodal Interaction*, pp. 132–143. Springer, 2008.

Eleftheriadis, S., Rudovic, O., and Pantic, M. Shared Gaussian process latent variable model for multi-view facial expression recognition. In *International Symposium on Visual Computing*, pp. 527–538. Springer, 2013.

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., and Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications*, 10(1):390, 2019.

Gopalan, P., Hofman, J. M., and Blei, D. M. Scalable recommendation with hierarchical Poisson factorization. In *Conference on Uncertainty in Artificial Intelligence*, pp. 326–335, 2015.

Graves, A. Stochastic backpropagation through mixture density distributions. *arXiv preprint arXiv:1607.05690*, 2016.

Gulrajani, I., Kumar, K., Ahmed, F., Taiga, A. A., Visin, F., Vazquez, D., and Courville, A. PixelVAE: A latent variable model for natural images. In *International Conference on Learning Representations*, 2016.

Gundersen, G., Zhang, M., and Engelhardt, B. Latent variable modeling with random features. In *International Conference on Artificial Intelligence and Statistics*, pp. 1333–1341. PMLR, 2021.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*, pp. 282–290, Arlington, Virginia, USA, 2013.

Hotelling, H. Relations between two sets of variates. *Biometrika*, 1936.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pp. 1724–1732. PMLR, 2017.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

Jung, Y., Song, K., and Park, J. Efficient approximate inference for stationary kernel on frequency domain. In *International Conference on Machine Learning*, pp. 10502–10538. PMLR, 2022.

Kim, J.-O. and Mueller, C. W. *Factor analysis: Statistical Methods and Practical Issues*, volume 14. sage, 1978.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, D. P. and Welling, M. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Lalchand, V., Ravuri, A., and Lawrence, N. D. Generalised GPLVM with stochastic variational inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 7841–7864. PMLR, 2022.

Lawrence, N. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6(60):1783–1816, 2005.

LeCun, Y. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Li, J., Zhang, B., and Zhang, D. Shared autoencoder Gaussian process latent variable model for visual classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4272–4286, 2017.

Lopez-Paz, D., Sra, S., Smola, A., Ghahramani, Z., and Schölkopf, B. Randomized nonlinear component analysis. In *International Conference on Machine Learning*, pp. 1359–1367. PMLR, 2014.

Lotfi, S., Izmailov, P., Benton, G., Goldblum, M., and Wilson, A. G. Bayesian model selection, the marginal likelihood, and generalization. In *International Conference on Machine Learning*, pp. 14223–14247. PMLR, 2022.

Lucas, J., Tucker, G., Grosse, R. B., and Norouzi, M. Don't blame the ELBO! A linear VAE perspective on posterior collapse. *Advances in Neural Information Processing Systems*, 32, 2019.

Menon, S., Blei, D., and Vondrick, C. Forget-me-not! Contrastive critics for mitigating posterior collapse. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1360–1370. PMLR, 2022.

Nakagawa, N., Togo, R., Ogawa, T., and Haseyama, M. Gromov-Wasserstein autoencoders. In *Proceedings of International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sbS10BCtc7.

Oliva, J. B., Dubey, A., Wilson, A. G., Póczos, B., Schneider, J., and Xing, E. P. Bayesian nonparametric kernel-learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1078–1086. PMLR, 2016.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural iInformation Processing Systems*, 32, 2019.

Pearson, K. LIII. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11): 559–572, 1901.

Poux-Médard, G., Velcin, J., and Loudcher, S. Powered Dirichlet process-controlling the "rich-get-richer" assumption in bayesian clustering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 611–626. Springer, 2023.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pp. 1177–1184, 2007.

Ramchandran, S., Koskinen, M., and Lähdesmäki, H. Latent Gaussian process with composite likelihoods and numerical quadrature. In *International Conference on Artificial Intelligence and Statistics*, pp. 3718–3726. PMLR, 2021.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Razavi, A., Oord, A. v. d., Poole, B., and Vinyals, O. Preventing posterior collapse with delta-VAEs. *arXiv preprint arXiv:1901.03416*, 2019.

Roweis, S. T. and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500): 2323–2326, 2000.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder variational autoencoders. *Advances in Neural Information Processing Systems*, 29, 2016.

Song, G., Wang, S., Huang, Q., and Tian, Q. Similarity Gaussian process latent variable model for multi-modal data analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4050–4058, 2015.

Theodoridis, S. *Machine Learning: A Bayesian and Optimization Perspective*. Academic Press, 2nd edition, 2020.

Tipping, M. E. and Bishop, C. M. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999.

Titsias, M. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 567–574. PMLR, 2009.

Titsias, M. and Lawrence, N. D. Bayesian Gaussian process latent variable model. In *International Conference on Artificial Intelligence and Statistics*, pp. 844–851. PMLR, 2010.

Tran, B.-H., Shahbaba, B., Mandt, S., and Filippone, M. Fully Bayesian autoencoders with latent sparse Gaussian processes. In *International Conference on Machine Learning*, pp. 34409–34430. PMLR, 23–29 Jul 2023.

Tropp, J. A. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8 (1-2):1–230, 2015.

Wang, Y., Blei, D., and Cunningham, J. P. Posterior collapse and latent variable non-identifiability. *Advances in Neural Information Processing Systems*, 34:5443–5455, 2021.

Wang, Z. and Liu, Z. Posterior collapse of a linear latent variable model. In *Advances in Neural Information Processing Systems*, 2022.

Wilson, A. and Adams, R. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, pp. 1067–1075. PMLR, 2013.

Wilson, A. G. and Izmailov, P. Bayesian deep learning and a probabilistic perspective of generalization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 4697–4708, 2020.

Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Yang, Z., Hu, Z., Salakhutdinov, R., and Berg-Kirkpatrick, T. Improved variational autoencoders for text modeling using dilated convolutions. In *International conference on machine learning*, pp. 3881–3890. PMLR, 2017.

Zarzoso, V., Moreau, E., Gribonval, R., and Vincent, E. *Latent Variable Analysis and Signal Separation*. Springer, 2010.

Zhang, M. M., Gundersen, G. W., and Engelhardt, B. E. Bayesian non-linear latent variable modeling via random fourier features. *arXiv preprint arXiv:2306.08352*, 2023.

Zhao, H., Rai, P., Du, L., Buntine, W., Phung, D., and Zhou, M. Variational autoencoders for sparse and overdispersed discrete data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1684–1694. PMLR, 2020.

Zheng, C. and Vedaldi, A. Online clustered codebook. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22798–22807, 2023.

Zhu, H., Balsells-Rodas, C., and Li, Y. Markovian Gaussian process variational autoencoders. In *International Conference on Machine Learning*, pp. 42938–42961. PMLR, 2023.

APPENDICES

A Model Collapse Mechanism Revelation **13**

A.1 Special Case of GPLVM: Dual Probabilistic Principal Analysis (DPPCA) . . . 14

A.2 Proof of Theorem 3.1 . . . 14

A.3 Proof of Proposition 3.2 . . . 15

A.3.1 Auxiliary Theorem . . . 15

A.3.2 Proof of Proposition 3.2 . . . 17

A.4 Proof of Proposition 3.3 . . . 18

B Modeling and Variational Approximation **19**

B.1 ELBO Derivation and Evaluation . . . 19

B.2 Interpretation of Modeling and Variational Distribution . . . 20

C Auto-differentiable SM Kernel using RFF Approximation **21**

C.1 Proof of Proposition 4.2 . . . 21

C.2 Proof of Theorem 4.3 . . . 21

D Extended Related Work **24**

E Experiment Details **25**

E.1 Data Descriptions and Preprocessing . . . 25

E.2 Benchmark Methods Descriptions . . . 26

E.3 Default Hyperparameter Configurations . . . 27

E.4 Additional Results . . . 27

E.4.1 S-shaped Latent Manifold Estimation . . . 27

E.4.2 Missing Data Imputation . . . 27

E.4.3 KNN Classification Accuracy with Varying K . . . 31

E.4.4 Larger Datasets Extension . . . 31

A. Model Collapse Mechanism Revelation

In § A.1, we provide a detailed introduction to dual probabilistic principal analysis (DPPCA) (Lawrence, 2005) and establish its connection with the linear GPLVM. Building upon this connection, a detailed derivation of Theorem 3.1 is provided, delineating the forms of stationary points. Through further exploration of the optimization landscapes around stationary points, we provide detailed proofs for Proposition 3.2 and Proposition 3.3, located in § A.3 and §. A.4, respectively.

A.1. Special Case of GPLVM: Dual Probabilistic Principal Analysis (DPPCA)

In DPPCA (Lawrence, 2005), each observed data point $\mathbf{y}_i \in \mathbb{R}^M$ is generated from a latent variable $\mathbf{x}_i \in \mathbb{R}^Q$ through a linear transformation $\mathbf{A} \in \mathbb{R}^{M\times Q}$, i.e.,

$$\mathbf{y}_i \sim \mathcal{N}\left(\mathbf{A}\mathbf{x}_i, \sigma^2\mathbf{I}_M\right), \tag{17a}$$

$$p(\mathbf{A}) \sim \prod^{M} \mathcal{N}\left(\mathbf{0}, \mathbf{I}_Q\right), \tag{17b}$$

where σ^2 represents the projection variance, representing the uncertainty. For N observed data points in DPPCA, denoted as $\mathbf{Y} \in \mathbb{R}^{N\times M}$, the marginal likelihood, obtained by marginalizing the transformation matrix $\mathbf{A}$, can be represented as follows:

$$\mathbf{y}_{:,j}|\mathbf{X} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N\right), \quad j = 1, \ldots, M, \tag{18}$$

where $\mathbf{y}_{:,j}$ denotes j-th column in the observed data $\mathbf{Y}$. Consequently, the maximum likelihood estimate (MLE) for the latent variable, denoted as $\hat{\mathbf{X}}_{\text{DPPCA}}$, can be derived by maximizing the logarithm of Eq. (18) through, e.g., gradient-based methods, i.e.,

$$\hat{\mathbf{X}}_{\text{DPPCA}} = \max_{\mathbf{X}} \log \prod_{j=1}^{M} \mathcal{N}\left(\mathbf{y}_{:,j} \mid \mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N\right). \tag{19}$$

Building upon Eq. (19) and the optimization problem given in Eq. (6), a connection between GPLVM and DPPCA can be established (Lawrence, 2005), encapsulated in the following corollary:

Corollary A.1. *Assuming the kernel function in GPLVM is defined as the inner product kernel with $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top\mathbf{x}'$, the stationary points for the linear GPLVM, as expressed in Eq. (6), are identical to the stationary points of DPPCA, $\hat{\mathbf{X}}_{DPPCA}$.*

Proof. If the kernel function is the inner product kernel, i.e., $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top\mathbf{x}'$, the marginal likelihood of the linear GPLVM can be reformulated as,

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{M} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N\right). \tag{20}$$

Then, the stationary points of the linear GPLVM, $\hat{\mathbf{X}}$, is given by

$$\hat{\mathbf{X}} = \max_{\mathbf{X}} \log p(\mathbf{Y}|\mathbf{X}) = \max_{\mathbf{X}} M\left\{-\frac{N}{2}\log 2\pi - \frac{1}{2}\log\left|\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N\right|\right\} - \frac{1}{2}\operatorname{tr}\left(\left(\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N\right)^{-1}\mathbf{Y}\mathbf{Y}^\top\right). \tag{21}$$

The stationary points of DPPCA, $\hat{\mathbf{X}}_{\text{DPPCA}}$, given in Eq. (19), can be reformulated as

$$\hat{\mathbf{X}}_{\text{DPPCA}} = \max_{\mathbf{X}} M\left\{-\frac{N}{2}\log 2\pi - \frac{1}{2}\log\left|\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N\right|\right\} - \frac{1}{2}\operatorname{tr}\left(\left(\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N\right)^{-1}\mathbf{Y}\mathbf{Y}^\top\right). \tag{22}$$

It is evident that the stationary points of the linear GPLVM is identical to the stationary points of DPPCA. □

A.2. Proof of Theorem 3.1

This subsection conducts a comprehensive derivation, elucidating the stationary points of the linear GPLVM. Our derivation generally adheres to the one in (Lawrence, 2005), albeit with subtle distinctions.

Proof. Recall that, the log marginal likelihood can be expressed as

$$L \triangleq M\left\{-\frac{N}{2}\log 2\pi - \frac{1}{2}\log\left|\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N\right|\right\} - \frac{1}{2}\operatorname{tr}\left(\left(\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N\right)^{-1}\mathbf{Y}\mathbf{Y}^\top\right). \tag{23}$$

Define $\mathbf{K} \triangleq \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N$, Eq. (23) could be reformulated as

$$L = M\left\{-\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}|\right\} - \frac{1}{2}\operatorname{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^\top). \tag{24}$$

Taking the gradient of Eq. (24) with respect to $\mathbf{X}$, we have

$$\frac{\partial L}{\partial \mathbf{X}} = \mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^\top\mathbf{K}^{-1}\mathbf{X} - M\mathbf{K}^{-1}\mathbf{X}. \tag{25}$$

Setting this gradient to zero, the stationary points of Eq. (24) should satisfy

$$\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top\mathbf{K}^{-1}\mathbf{X} = \mathbf{X}. \tag{26}$$

According to Lemma B.2, we have

$$\mathbf{K}^{-1}\mathbf{X} = \left[\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N\right]^{-1}\mathbf{X} = \mathbf{X}\left[\mathbf{X}^\top\mathbf{X} + \sigma^2\mathbf{I}_Q\right]^{-1}. \tag{27}$$

We conduct singular value decomposition (SVD) to $\mathbf{X}$, and get $\mathbf{X} = \mathbf{U}\mathbf{L}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{N\times Q}$, $\mathbf{L} = \mathrm{diag}(l_1, l_2, \ldots, l_Q) \in \mathbb{R}^{Q\times Q}$ is a diagonal matrix, and $\mathbf{V} \in \mathbb{R}^{Q\times Q}$. Together with Eq. (27) and Eq. (26), we have

$$\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top\mathbf{U}\mathbf{L}\left[\mathbf{L}^2 + \sigma^2\mathbf{I}_Q\right]^{-1}\mathbf{V}^\top = \mathbf{U}\mathbf{L}\mathbf{V}^\top, \tag{28a}$$

$$\Rightarrow \quad \frac{1}{M}\mathbf{Y}\mathbf{Y}^\top\mathbf{U}\mathbf{L}\left[\mathbf{L}^2 + \sigma^2\mathbf{I}_Q\right]^{-1} = \mathbf{U}\mathbf{L}, \tag{28b}$$

$$\Rightarrow \quad \frac{1}{M}\mathbf{Y}\mathbf{Y}^\top\mathbf{U}\mathbf{L} = \mathbf{U}(\sigma^2\mathbf{I}_Q + \mathbf{L}^2)\mathbf{L}. \tag{28c}$$

Then, we have:

- If $l_i \neq 0$, it indicates that $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top\mathbf{u}_i = \mathbf{u}_i(\sigma^2 + l_i^2)$, implying $\mathbf{u}_i$ is an eigenvector of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$ corresponding to the eigenvalue $\lambda_i = \sigma^2 + l_i^2$.
- If $l_i = 0$, the vector $\mathbf{u}_i$ is arbitrary. We can set it to be an eigenvector of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$ for consistency.

Consequently, all potential stationary solutions for $\mathbf{X}$ can be written as

$$\hat{\mathbf{X}} = \mathbf{U}_Q\left(\mathbf{\Lambda}_Q - \sigma^2\mathbf{I}_Q\right)^{1/2}\mathbf{R}, \tag{29}$$

where $\mathbf{U}_Q \in \mathbb{R}^{N\times Q}$ is a matrix whose columns are eigenvectors of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$, $\mathbf{R} \in \mathbb{R}^{Q\times Q}$ is an arbitrary orthogonal matrix and $\mathbf{\Lambda}_Q \in \mathbb{R}^{Q\times Q}$ is a diagonal matrix with:

$$[\mathbf{\Lambda}_Q]_{i,i} = \begin{cases} \lambda_i, \text{ the corresponding eigenvalue to } \mathbf{u}_i, \text{ or }, \\ \sigma^2. \end{cases} \tag{30}$$

□

A.3. Proof of Proposition 3.2

A.3.1. Auxiliary Theorem

Before delving into the proof, we first proceed to characterize the stationary point of σ^2 in the linear GPLVM, which is summarized in the following theorem.

Theorem A.2. *Given $\hat{\mathbf{X}}$, stationary points of the projection variance, denoted as $\hat{\sigma}^2$, could be obtained by solving the following optimization problem*

$$\max_{\sigma^2} \log P(\mathbf{Y} \mid \hat{\mathbf{X}}). \tag{31}$$

It turns out that $\hat{\sigma}^2$ takes the following form:

$$\hat{\sigma}^2 = \frac{1}{N - Q'}\sum_{j=Q'+1}^{N}\lambda_j, \tag{32}$$

where Q' is the number of eigenvalues retained in $\mathbf{\Lambda}_Q$.

Proof. To obtain the stationary point for σ^2, we substitute the stationary point for $\mathbf{X}$, defined in Eq. (7), into the log marginal likelihood function Eq. (24) to give

$$L = -\frac{M}{2}\left\{N\log 2\pi + \sum_{j=1}^{Q'}\log(\lambda_j) + (N-Q')\ln\sigma^2 + \frac{1}{\sigma^2}\sum_{j=Q'+1}^{N}\lambda_j + Q'\right\}, \tag{33}$$

where Q' represents the number of $[\mathbf{\Lambda}_Q]_{i,i}, i \in 1, ..., Q$ that are not equal to σ^2, see Eq. (30). Consequently, $\lambda_1, ..., \lambda_{Q'}$ denote the eigenvalues associated with the eigenvectors "retained" in $\mathbf{X}$, while $\lambda_{Q'+1}, ..., \lambda_N$ refer to the eigenvalues that are "discarded".

By taking the gradient of Eq. (33) with respect to σ^2 and setting it to zero, we obtain:

$$\hat{\sigma}^2 = \frac{1}{N-Q'}\sum_{j=Q'+1}^{N}\lambda_j.$$

□

Remark A.3. Note that the eigenvalues $\{\lambda_{Q'+1}, \dots \lambda_N\}$ can be interpreted as the discarded/lost information in the inverse projection process ($\mathbf{Y} \to \mathbf{X}$), and the corresponding eigenvectors are treated as **discarded vectors**.

In addition, with Theorem A.2, we can immediately get the following corollary.

Corollary A.4. *If $\mathbf{\Lambda}_Q$ contains the first Q principal eigenvalues of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$, then the corresponding stationary point becomes the global maximum, which could be represented as:*

$$\sigma^{2\star} = \frac{1}{N-Q}\sum_{j=Q+1}^{N}\lambda_j^o, \tag{34a}$$

$$\mathbf{X}^\star = \mathbf{U}_Q^\star\left(\mathbf{\Lambda}_Q^\star - (\sigma^2)^\star\mathbf{I}_Q\right)^{1/2}\mathbf{R}, \tag{34b}$$

where $[\lambda_1^o, ..., \lambda_N^o]$ representing the eigenvalues of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$ with $\lambda_1^o \geq \lambda_2^o, ..., \geq \lambda_N^o$. Additionally, $\mathbf{U}_Q^\star \in \mathbb{R}^{N\times Q}$ are the first Q principal eigenvectors of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$, with the associated eigenvalues $\mathbf{\Lambda}_Q^\star = \mathrm{diag}(\lambda_1^o, \lambda_2^o, \dots, \lambda_Q^o)$. The optimal projection variance, $\sigma^{2\star}$, represents the average variance lost in the projection process.

Proof. With the stationary point of σ^2 given in Eq.(32), the log marginal likelihood, given in Eq. (24), becomes

$$L = -\frac{M}{2}\left\{\sum_{j=1}^{Q'}\log(\lambda_j) + (N-Q')\log\left(\frac{1}{N-Q'}\sum_{j=Q'+1}^{N}\lambda_j\right) + N\log(2\pi) + N\right\}. \tag{35}$$

Because of the constancy of the sum of all eigenvalues λ_j (given the data $\mathbf{Y}$), maximizing Eq. (35) is equivalently to minimize the following quantity

$$E = \log\left(\frac{1}{N-Q'}\sum_{i=Q'+1}^{N}\lambda_i\right) - \frac{1}{N-Q'}\sum_{i=Q'+1}^{N}\log(\lambda_i), \tag{36}$$

which solely relies on the discarded eigenvalues and remains non-negative (indeed due to Jensen's inequality). Remarkably, the minimization of E necessitates only that the discarded λ_j values are contiguous within the spectrum of the ordered eigenvalues of matrix $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$. However, in addition to this, Eq. (29) imposes the condition that $\lambda_j > \sigma^2$ for all i in the set $\{1, 2, \dots, Q'\}$. Consequently, based on Eq. (32), it can be inferred that the smallest eigenvalue must be among the discarded ones. This deduction is sufficient to establish that E is minimized when $\lambda_{Q'+1}, \dots, \lambda_N$ represent the smallest $N - Q'$ eigenvalues. As a result, the likelihood L is maximized when $\lambda_1, \dots, \lambda_{Q'}$ are the largest eigenvalues of matrix $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$. It is worth noting that the maximization of L concerning Q' is achieved when there are the fewest terms in the sums outlined in Eq. (36). This occurs when $Q' = Q$, ensuring that none of the $l_i, i \in 1, ..., Q$ terms are zero. □

A.3.2. Proof of Proposition 3.2

1) **Outline of Proof.**

Without loss of generality, we assume the rotation matrix $\mathbf{R} = \mathbf{I}_Q$ in Eq. (7), Theorem 3.1, resulting in the stationary points of the latent variable

$$\hat{\mathbf{X}} = \mathbf{U}_Q \left(\mathbf{\Lambda}_Q - \hat{\sigma}^2 \mathbf{I}_Q\right)^{1/2}. \tag{37}$$

Based upon this form, we seek to explore the structure of the optimization landscape around $\hat{\mathbf{X}}$ by examining the variation trend of the log marginal likelihood L at $\hat{\mathbf{X}}$ in the spanned space of discarded vectors, denoted as $\mathrm{Span}(\mathbf{U}_D)$, where

$$\mathbf{U}_D \triangleq [\mathbf{u}_{Q'+1}, ..., \mathbf{u}_N].$$

Intuitively, if the evaluation of L at a stationary point consistently decreases for all axes in $\mathrm{Span}(\mathbf{U}_D)$, then we can consider the corresponding stationary point as a ***local optimum*** or ***global optimum***, and vice versa; If the evaluation of L at a stationary point consistently increases along any axis and decreases along any others within $\mathrm{Span}(\mathbf{U}_D)$, the corresponding stationary point can be recognized as a ***saddle point***.

2) **Quantitative Analysis.**

To quantitatively analyze the variation in L at $\hat{\mathbf{X}}$ within $\mathrm{Span}(\mathbf{U}_D)$, we introduce a small perturbation to the i-th column of $\hat{\mathbf{X}}$ in the form of $\epsilon \mathbf{u}_j$, resulting in the perturbed stationary point $\hat{\mathbf{X}}^\epsilon$ with

$$[\hat{\mathbf{X}}^\epsilon]_{:,i} = \hat{\mathbf{x}}_i + \epsilon \mathbf{u}_j, \quad i = 1, 2, \ldots, Q, \tag{38}$$

where $\hat{\mathbf{x}}_i$ denotes the i-th column of $\hat{\mathbf{X}}^\epsilon$, with a bit abuse of notation, and ϵ is an arbitrarily small positive constant and $\mathbf{u}_j, j \in Q'+1, ..., N$ represents a principal axis in $\mathrm{Span}(\mathbf{U}_D)$. The variation trends from $L(\hat{\mathbf{X}})$ to $L(\hat{\mathbf{X}}^\epsilon)$ can be determined by examining the sign of the dot product of the perturbation $\mathbf{u}_j$ with the gradient at $\hat{\mathbf{x}}_i + \epsilon \mathbf{u}_j$. More precisely, when the sign is positive, the evaluation of L at $\hat{\mathbf{X}}$ will ascend as $\hat{\mathbf{x}}_i$ shifts towards the direction of $\mathbf{u}_j$, and vice versa. For clarity, let us denote the sign of the dot product as $\mathrm{sgn}(D_{ij})$, where D_{ij} denotes the dot product and is expressed as

$$D_{ij} = \mathbf{u}_j^\top \left\{\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^\top \mathbf{K}^{-1} \left(\hat{\mathbf{x}}_i + \epsilon \mathbf{u}_j\right) - M\mathbf{K}^{-1}\left(\hat{\mathbf{x}}_i + \epsilon \mathbf{u}_j\right)\right\}, \tag{39}$$

with $\mathbf{K} = \hat{\mathbf{X}}^\epsilon \hat{\mathbf{X}}^{\epsilon^\top} + \hat{\sigma}^2 \mathbf{I}_N$.

According to Lemma B.2 and Eq. (37), we have

$$\begin{aligned} \mathbf{K}^{-1}\hat{\mathbf{X}}^\epsilon &= \hat{\mathbf{X}}^\epsilon \left[(\hat{\mathbf{X}}^\epsilon)^\top \hat{\mathbf{X}}^\epsilon + \hat{\sigma}^2 \mathbf{I}_Q\right]^{-1}, \\ &= \hat{\mathbf{X}}^\epsilon \left[\mathbf{\Lambda}_Q^\epsilon\right]^{-1}, \end{aligned} \tag{40}$$

where $\mathbf{\Lambda}_Q^\epsilon$ is a diagonal matrix with:

$$[\mathbf{\Lambda}_Q^\epsilon]_{k,k} = \begin{cases} [\mathbf{\Lambda}_Q]_{k,k}, & \forall k \neq i, \\ [\mathbf{\Lambda}_Q]_{i,i} + \epsilon^2, & \text{otherwise}. \end{cases} \tag{41}$$

Checking the i-th column of the matrices on both sides of Eq. (40), we find that

$$\mathbf{K}^{-1}\left(\hat{\mathbf{x}}_i + \epsilon \mathbf{u}_j\right) = \frac{\hat{\mathbf{x}}_i + \epsilon \mathbf{u}_j}{[\mathbf{\Lambda}_Q]_{i,i} + \epsilon^2}. \tag{42}$$

Substituting $\mathbf{K}^{-1}\left(\hat{\mathbf{x}}_i + \epsilon \mathbf{u}_j\right) = \frac{(\hat{\mathbf{x}}_i + \epsilon \mathbf{u}_j)}{[\mathbf{\Lambda}_Q]_{i,i} + \epsilon^2}$ into the first term of Eq. (39) yields

$$\begin{aligned} D_{ij} &= M\mathbf{u}_j^\top \mathbf{K}^{-1}\left\{\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top \frac{\hat{\mathbf{x}}_i + \epsilon \mathbf{u}_j}{[\mathbf{\Lambda}_Q]_{i,i} + \epsilon^2} - \left(\hat{\mathbf{x}}_i + \epsilon \mathbf{u}_j\right)\right\}, \\ &= M\mathbf{u}_j^\top \mathbf{K}^{-1}\left\{\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top \frac{1}{[\mathbf{\Lambda}_Q]_{i,i} + \epsilon^2} - \mathbf{I}_N\right\}\hat{\mathbf{x}}_i + M\mathbf{u}_j^\top \mathbf{K}^{-1}\left\{\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top \frac{1}{[\mathbf{\Lambda}_Q]_{i,i} + \epsilon^2} - \mathbf{I}_N\right\}\epsilon \mathbf{u}_j, \qquad (43) \\ &\approx M\mathbf{u}_j^\top \mathbf{K}^{-1}\left\{\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top \frac{1}{[\mathbf{\Lambda}_Q]_{i,i}} - \mathbf{I}_N\right\}\hat{\mathbf{x}}_i + \epsilon M\mathbf{u}_j^\top \mathbf{K}^{-1}\left\{\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top \frac{1}{[\mathbf{\Lambda}_Q]_{i,i}} - \mathbf{I}_N\right\}\mathbf{u}_j. \qquad (44) \end{aligned}$$

According to Eq. (26), we have

$$\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top \frac{\hat{\mathbf{x}}_i}{[\mathbf{\Lambda}_Q]_{i,i}} = \hat{\mathbf{x}}_i. \tag{45}$$

Therefore, Eq. (44) can be rewritten as

$$D_{ij} = \epsilon M \left(\frac{\lambda_j}{[\mathbf{\Lambda}_Q]_{i,i}} - 1 \right) \mathbf{u}_j^\top \mathbf{K}^{-1} \mathbf{u}_j, \tag{46}$$

where λ_j represents the eigenvalues corresponding to $\mathbf{u}_j$.

Due to the positive definite property of $\mathbf{K}^{-1}$, $\mathrm{sgn}(D_{ij})$, see Eq. (46), relies solely on

$$\mathrm{sgn}\left(\frac{\lambda_j}{[\mathbf{\Lambda}_Q]_{i,i}} - 1 \right), \tag{47}$$

implying that the type of stationary points is dictated by the discarded and retained eigenvalues. Specifically,

(i) For $\hat{\mathbf{x}}_i, \forall i = 1, ..., Q$, if $[\mathbf{\Lambda}_Q]_{i,i} > \lambda_j, \forall j \in Q' + 1, ..., N$, then the corresponding stationary point should be recognized as a ***local or global optimum point***;

(ii) For $\hat{\mathbf{x}}_i, \forall i = 1, ..., Q$, if $[\mathbf{\Lambda}_Q]_{i,i} < \lambda_j, \forall j \in Q' + 1, ..., N$, then the corresponding stationary point should be recognized as a ***local minimum point***;

(iii) For $\hat{\mathbf{x}}_i, i = 1, ..., Q$, if $[\mathbf{\Lambda}_Q]_{i,i} > \lambda_j, [\mathbf{\Lambda}_Q]_{i,i} < \lambda_k,\ \exists j, k \in Q' + 1, ..., N$, then the corresponding stationary point should be identified as a ***saddle point***.

3) **Final Results.**

If $[\mathbf{\Lambda}_Q]_{i,i} = \lambda_i, \forall i \in 1, ..., Q$, the stationary point represents a global optimum when

$$\lambda_i > \lambda_j, \forall i \in 1, ..., Q, \text{ and } \forall j \in Q' + 1, ..., N.$$

However, if there exists a $\lambda_i < \lambda_j$, these stationary points correspond to saddle points. Additionally, when $\exists i \in 1, ..., Q, [\mathbf{\Lambda}_Q]_{i,i} = \hat{\sigma}^2$, the associated stationary points are deemed saddle points due to the existence of cases where

$$\hat{\sigma}^2 < \lambda_j, j \in Q' + 1, ..., N,$$

considering that $\hat{\sigma}^2$ is the average of the discarded eigenvalues. Because the saddle points could be escaped efficiently, they are generally regarded as unstable stationary points (Jin et al., 2017). Therefore, during the optimization process, when we set $\sigma^2 = \hat{\sigma}^2$, the only stable maximum point is the global optimum point.

Remark A.5. The analysis does not account for the equality of eigenvalues. This is because: (1) Equality among the first Q principal eigenvalues does not influence the presented analysis; (2) The equality of all discarded eigenvalues is trivial.

A.4. Proof of Proposition 3.3

Proof. Suppose the projection variance σ^2 takes a value within the range $(\lambda_Q^o, \lambda_{Q-1}^o)$, where λ_Q^o and λ_{Q-1}^o represent the Q-th and $(Q-1)$-th principal eigenvalues of $\frac{1}{M}\mathbf{Y}\mathbf{Y}^\top$, respectively. In this scenario, the eigenvectors with associated eigenvalues less than λ_Q^o are unambiguously discarded. Furthermore, in such a case, the only stable local optimum point[3] comprises the following $\mathbf{\Lambda}_Q$,

$$[\mathbf{\Lambda}_Q]_{i,i} = \begin{cases} \lambda_i^o, \text{ for } i \in 1, ..., Q-1, \text{ or }, \\ \sigma^2. \end{cases}$$

It is evident that, for either $[\mathbf{\Lambda}_Q]_{i,i} = \sigma^2$ or λ_i^o, $[\mathbf{\Lambda}_Q]_{i,i} > \lambda_j^o$ for all $i \in 1, ..., Q$ and for all $j \in Q-1, ..., N$, leading to the corresponding stationary points being the local optimum point, with one zero-column in $\hat{\mathbf{X}}$.

[3] Other stationary points, manifested as saddle points, are unstable as discussed in App. A.3.

If the projection variance falls within the range $(\lambda^o_{Q-1}, \lambda^o_{Q-2})$, the only stable local optimum point comprises the following $\mathbf{\Lambda}_Q$,

$$[\mathbf{\Lambda}_Q]_{i,i} = \begin{cases} \lambda^o_i, \text{ for } i \in 1, ..., Q-2, \text{ or }, \\ \sigma^2, \end{cases}$$

with two zero-columns in $\mathbf{X}$. By deduction, when $\sigma^2 > \lambda^o_1$, the only stable local optimum point comprises the $\mathbf{\Lambda}_Q$ with $[\mathbf{\Lambda}_Q]_{i,i} = \sigma^2$ for all $i \in 1, ..., Q$ with $\mathbf{X} = \mathbf{0}$.

It is worth noting that we deliberately avoid considering the equality of any of the Q principal eigenvalues to streamline the quantitative analysis, as introducing such equality might exacerbate the complexity and hasten the occurrence of model collapse. For instance, when the projection variance falls within the range $(\lambda^o_Q, \lambda^o_{Q-1})$ and there exist two eigenvectors with eigenvalues equal to λ^o_Q, the stable local optimum point entails two zero-columns.

Suppose $\sigma^2 < \lambda_N$, then, there exist a set of local minima point characterized by the following $\mathbf{\Lambda}_Q$,

$$[\mathbf{\Lambda}_Q]_{i,i} = \begin{cases} \lambda^o_i, \text{ for } i \in N-Q+k, ..., N, \text{ or }, \\ \sigma^2, \end{cases}$$

where k represents the last k principal eigenvalues that are selected. It is also noteworthy that these local minima point[4] will feature k zero-columns in $\mathbf{X}$.

□

B. Modeling and Variational Approximation

B.1. ELBO Derivation and Evaluation

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{q(\mathbf{X},\mathbf{W})}\left[\frac{p(\mathbf{Y},\mathbf{X},\mathbf{W})}{q(\mathbf{X},\mathbf{W})}\right] \\
&= \mathbb{E}_{q(\mathbf{X},\mathbf{W})}\left[\log \frac{p(\mathbf{W})\prod_{i=1}^N p(\mathbf{x}_i)\prod_{j=1}^M p(\mathbf{y}_{:,j}|\mathbf{X},\mathbf{W})}{p(\mathbf{W})\prod_{i=1}^N q(\mathbf{x}_i)}\right] \\
&= \underbrace{\sum_{j=1}^M \mathbb{E}_{q(\mathbf{X},\mathbf{W})}\left[\log p(\mathbf{y}_{:,j}|\mathbf{X},\mathbf{W})\right]}_{\text{Term 1: data reconstruction}} - \underbrace{\sum_{i=1}^N \mathrm{KL}(q(\mathbf{x}_i)\|p(\mathbf{x}_i))}_{\text{Term 2: regularization}} \\
&\approx \sum_{j=1}^M \frac{1}{I}\sum_{i=1}^I \log \mathcal{N}(\mathbf{y}_{:,j}|\mathbf{0}, \hat{\mathbf{K}}^{(i)}_{\text{sm}} + \sigma^2\mathbf{I}_N) - \frac{1}{2}\sum_{i=1}^N \Big[\mathrm{tr}(\mathbf{S}_i) + \boldsymbol{\mu}_i^\top\boldsymbol{\mu}_i - \log|\mathbf{S}_i| - Q\Big] \\
&\approx \sum_{j=1}^M \frac{1}{I}\sum_{i=1}^I \left\{-\frac{N}{2}\log 2\pi - \frac{1}{2}\log\left|\hat{\mathbf{K}}^{(i)}_{\text{sm}} + \sigma^2\mathbf{I}_N\right| - \frac{1}{2}\mathbf{y}_{:,j}^\top\left(\hat{\mathbf{K}}^{(i)}_{\text{sm}} + \sigma^2\mathbf{I}_N\right)^{-1}\mathbf{y}_{:,j}\right\} \\
&\quad - \frac{1}{2}\sum_{i=1}^N \Big[\mathrm{tr}(\mathbf{S}_i) + \boldsymbol{\mu}_i^\top\boldsymbol{\mu}_i - \log|\mathbf{S}_i| - Q\Big]
\end{aligned}$$

where $\mathbf{S}_i$ is typically assumed to be a diagonal matrix. Note that $\hat{\mathbf{K}}^{(i)}_{\text{sm}} = \Phi_{\text{sm}}(\mathbf{X}^{(i)};\mathbf{W})\Phi_{\text{sm}}(\mathbf{X}^{(i)};\mathbf{W})^\top$, where $\Phi_{\text{sm}}(\mathbf{X}^{(i)};\mathbf{W}) \in \mathbb{R}^{N\times mL}$.

Lemma B.1. *Suppose* $\mathbf{A}$ *is an invertible* n*-by-*n *matrix and* $\mathbf{U}, \mathbf{V}$ *are* n*-by-*m *matrices. Then the following determinant equality holds.*

$$\left|\mathbf{A} + \mathbf{U}\mathbf{V}^\top\right| = \left|\mathbf{I}_{\text{m}} + \mathbf{V}^\top\mathbf{A}^{-1}\mathbf{U}\right|\,|\mathbf{A}|$$

[4]Equality among the last Q principal eigenvalues does not impact the analysis presented.

Lemma B.2 (Woodbury matrix identity)**.** *Suppose* $\mathbf{A}$ *is an invertible* n*-by-*n *matrix and* $\mathbf{U}, \mathbf{V}$ *are* n*-by-*m *matrices. Then*

$$\left(\mathbf{A}+\mathbf{U}\mathbf{V}^\top\right)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I}_\mathrm{m} + \mathbf{V}^\top\mathbf{U})^{-1}\mathbf{V}^\top$$

According to the above two lemmas (Rasmussen & Williams, 2006), in the case that $N \gg mL$, we can compute the determinant and inversion of $\hat{\mathbf{K}}_{\text{sm}}^{(i)} + \sigma^2\mathbf{I}_N$, reducing the computational complexity of the ELBO evaluation from the original $\mathcal{O}(N^3)$ to $\mathcal{O}(N(mL)^2)$.

$$\left|\hat{\mathbf{K}}_{\text{sm}}^{(i)} + \sigma^2\mathbf{I}_N\right| = \left|\mathbf{I}_{mL} + \frac{1}{\sigma^2}\Phi_{\text{sm}}^\top\Phi_{\text{sm}}\right| \left|\sigma^2\mathbf{I}_N\right| = \sigma^{2N}\left|\mathbf{I}_{mL} + \frac{1}{\sigma^2}\Phi_{\text{sm}}^\top\Phi_{\text{sm}}\right|, \tag{48}$$

$$\left(\hat{\mathbf{K}}_{\text{sm}}^{(i)} + \sigma^2\mathbf{I}_N\right)^{-1} = \frac{1}{\sigma^2}\left[\mathbf{I}_N - \Phi_{\text{sm}}(\mathbf{I}_{mL} + \Phi_{\text{sm}}^\top\Phi_{\text{sm}})^{-1}\Phi_{\text{sm}}^\top\right]. \tag{49}$$

B.2. Interpretation of Modeling and Variational Distribution

In Eqs. (10) and (11), we have modeled the spectral points $\mathbf{W}$ as a part of the data generation process. However, this might cause some confusion, which is clarified as follows.

- If we have selected the kernel function, the probability model for the data, i.e., Eqs. (10) and (11), can be interpreted as independent of $p(\mathbf{W})$, as it is inherent to the kernel function.
- In this paper, we provide another interpretation perspective: Following the setting from RFLVM by Gundersen et al. (2021), we consider the data-generating process for observations $\mathbf{Y}$ as outlined in Eq. (10) or (11), which is dependent on $\mathbf{W}$. Subsequently, we constrain its prior $p(\mathbf{W})$ to be Gaussian mixtures, defining the prior SM kernels functions. This alternative perspective is explained as follows:
  - Let us explicitly assume a parametric variational distribution $q_{\boldsymbol{\eta}}(\mathbf{W})$, assuming it to be another Gaussian mixture (thus still defines an SM kernel) with parameters denoted as $\boldsymbol{\eta}$ to approximate $p(\mathbf{W}|\mathbf{Y})$. In this case, Eq. (12) becomes:
    $$q(\mathbf{X}, \mathbf{W}) = q_{\boldsymbol{\eta}}(\mathbf{W})q(\mathbf{X}).$$
    Combining the joint distribution in Eq. (11), we derive the following ELBO:
    $$\begin{aligned}\mathcal{L} &= E_{q(\mathbf{X},\mathbf{W})}\left[\log\frac{p(\mathbf{X})p_{\boldsymbol{\theta}}(\mathbf{W})p_\sigma(\mathbf{Y}|\mathbf{X},\mathbf{W})}{q(\mathbf{X})q_{\boldsymbol{\eta}}(\mathbf{W})}\right] \\ &= E_{q(\mathbf{X},\mathbf{W})}\left[\log p_\sigma(\mathbf{Y}|\mathbf{X},\mathbf{W})\right] - KL(q(\mathbf{X})\|p(\mathbf{X})) - KL(q_{\boldsymbol{\eta}}(\mathbf{W})\|p_{\boldsymbol{\theta}}(\mathbf{W})).\end{aligned} \tag{50}$$
    In this ELBO, prior distribution $p_{\boldsymbol{\theta}}(\mathbf{W})$ is only related to the last KL divergence term. When maximizing the ELBO, we will obtain that $\boldsymbol{\theta} = \boldsymbol{\eta}$, ensuring that the last KL divergence term becomes 0. Ultimately, this aligns with the optimization objective in our paper.
- **More complicated** $p(\mathbf{W}|\mathbf{Y})$ **approximations**. It is possible to consider assuming the variational distribution of spectral points is $\mathbf{Y}$-dependent $q(\mathbf{W}|\mathbf{Y})$, such as a parametric Gaussian mixture and other distributions.
  - **Gaussian mixture**: Suppose we use a parametric variational distribution $q_{\boldsymbol{\eta}}(\mathbf{W}|\mathbf{Y})$, in the form of
    $$q_{\boldsymbol{\eta}}(\mathbf{W}|\mathbf{Y}) = \prod_{l=1}^{L/2}\sum_{i=1}^{m}\alpha_i\mathcal{N}_{\boldsymbol{\eta}_i}(\mu_i, \sigma_i^2), \tag{51}$$
    where $(\alpha_i, \mu_i, \sigma_i^2)$ in each mixture component is modeled by an encoder parametrized by $\boldsymbol{\eta}_i$ with $\mathbf{Y}$ as input. Similarly, we can get the ELBO:
    $$\begin{aligned}\mathcal{L} &= E_{q(\mathbf{X},\mathbf{W}|\mathbf{Y})}\left[\log\frac{p(\mathbf{X})p_{\boldsymbol{\theta}}(\mathbf{W})p_\sigma(\mathbf{Y}|\mathbf{X},\mathbf{W})}{q(\mathbf{X})q_{\boldsymbol{\eta}}(\mathbf{W}|\mathbf{Y})}\right] \\ &= E_{q(\mathbf{X},\mathbf{W}|\mathbf{Y})}\left[\log p_\sigma(\mathbf{Y}|\mathbf{X},\mathbf{W})\right] - KL(q(\mathbf{X})\|p(\mathbf{X})) - KL(q_{\boldsymbol{\eta}}(\mathbf{W}|\mathbf{Y})\|p_{\boldsymbol{\theta}}(\mathbf{W})).\end{aligned} \tag{52}$$
    When maximizing the ELBO, the last KL divergence term will also be 0. The difference between the remaining terms and our objective function lies in the first term, where $\mathbf{W}$ includes information learnt from $\mathbf{Y}$. This potentially enhances the kernel selection process and contributes to preventing model collapse, though coming at the cost of increased computational complexity from the encoder evaluation.

– **Other distribution forms**: In this case, the variational inference algorithm heavily depends on the specific form of $q(\mathbf{W}|\mathbf{Y})$. While this variational distribution can be more general, such an assumption generally introduces greater intractability, making the evaluation of the ELBO more challenging. Employing Monte Carlo sampling to approximate the ELBO in such scenarios could result in larger approximation variances compared to the case where $q(\mathbf{W}) = p(\mathbf{W})$, thus potentially leading to less robust model performance.

C. Auto-differentiable SM Kernel using RFF Approximation

C.1. Proof of Proposition 4.2

Proof. With the RFF feature map defined in Eq. (14), we can write down the inner product of the feature maps

$$\phi\left(\mathbf{x};\mathbf{W}\right)^{\top}\phi\left(\mathbf{x}';\mathbf{W}\right)=\sum_{i=1}^{m}\alpha_i\sum_{l=1}^{\frac{L}{2}}\frac{2}{L}\cos\Big(2\pi\mathbf{w}_l^{(i)\top}\left(\mathbf{x}-\mathbf{x}'\right)\Big) \tag{53}$$

where $\mathbf{W}\triangleq\{\mathbf{w}_1^{(i)},\mathbf{w}_2^{(i)},\ldots,\mathbf{w}_{L/2}^{(i)}\}_{i=1}^{m}$, and each $\mathbf{w}_l^{(i)}$ are i.i.d. sampled from the symmetric distribution

$$s_i(\mathbf{w})=\frac{\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_i,\mathrm{diag}(\boldsymbol{\sigma}_i^2))+\mathcal{N}(-\mathbf{w}|\boldsymbol{\mu}_i,\mathrm{diag}(\boldsymbol{\sigma}_i^2))}{2}$$

using reparameterization trick (Kingma & Welling, 2019). Taking the expectation w.r.t. $p\left(\mathbf{W}\right)=\prod_{i=1}^{m}\prod_{l=1}^{L/2}s_i(\mathbf{w})$, we can get

$$\begin{aligned}
&\mathbb{E}_{p(\mathbf{W})}\left[\phi\left(\mathbf{x};\mathbf{W}\right)^{\top}\phi\left(\mathbf{x}';\mathbf{W}\right)\right]=\mathbb{E}_{p(\mathbf{W})}\left[\sum_{i=1}^{m}\alpha_i\sum_{l=1}^{L/2}\frac{2}{L}\cos\Big(2\pi\mathbf{w}_l^{(i)\top}\left(\mathbf{x}-\mathbf{x}'\right)\Big)\right]\\
&=\sum_{i=1}^{m}\alpha_i\mathbb{E}_{p\left(\mathbf{w}_{1:L/2}^{(i)}\right)}\left[\sum_{l=1}^{L/2}\frac{2}{L}\cos\Big(2\pi\mathbf{w}_l^{(i)\top}\left(\mathbf{x}-\mathbf{x}'\right)\Big)\right] && \text{(linearity of expectation)} \quad (54a)\\
&=\sum_{i=1}^{m}\alpha_i\mathbb{E}_{s_i(\mathbf{w})}\left[\cos\Big(2\pi\mathbf{w}_1^{(i)\top}\left(\mathbf{x}-\mathbf{x}'\right)\Big)\right] && \text{(i.i.d. of } \mathbf{w}_l^{(i)}\text{)} \quad (54b)\\
&=\sum_{i=1}^{m}\alpha_i\mathbb{E}_{s_i(\mathbf{w})}\left[\frac{\exp\Big(2\pi j\mathbf{w}_1^{(i)\top}\left(\mathbf{x}-\mathbf{x}'\right)\Big)+\exp\Big(-2\pi j\mathbf{w}_1^{(i)\top}\left(\mathbf{x}-\mathbf{x}'\right)\Big)}{2}\right] && \text{(Euler's identity)} \quad (54c)\\
&=\sum_{i=1}^{m}\alpha_i k_i(\mathbf{x},\mathbf{x}';\boldsymbol{\mu}_i,\boldsymbol{\sigma}_i^{\mathbf{2}}) && \text{(symmetrity of } s_i(\mathbf{w})\text{)} \quad (54d)\\
&=k_{\mathrm{sm}}(\mathbf{x},\mathbf{x}';\{\alpha_i,\boldsymbol{\mu}_i,\boldsymbol{\sigma}_i^{\mathbf{2}}\}_{i=1}^{m}) && \text{(SM kernel definition)} \quad (54e)
\end{aligned}$$

Hence concludes that $\phi\left(\mathbf{x};\mathbf{W}\right)^{\top}\phi\left(\mathbf{x};\mathbf{W}\right)$ is an unbiased estimator of the SM kernel characterized by parameter $\{\alpha_i,\boldsymbol{\mu}_i,\boldsymbol{\sigma}_i^{\mathbf{2}}\}_{i=1}^{m}$. □

C.2. Proof of Theorem 4.3

Proof. Similar theorem has been proven in the Gaussian process regression model; see Proposition 3.1 in (Jung et al., 2022), and Theorem 3 in (Lopez-Paz et al., 2014). For ease of reference, we follow the existing results and show the proof as follows.

- To prove Theorem 4.3, we first introduce the following Lemma for Matrix Bernstein inequality (Tropp, 2015).

Lemma C.1 (Matrix Bernstein Inequality)**.** *Consider a finite sequence $\{\boldsymbol{X}_i\}$ of independent, random, Hermitian matrices with dimension N. Assume that*

$$\mathbb{E}[\boldsymbol{X}_i] = \mathbf{0} \text{ and } \|\boldsymbol{X}_i\|_2 \leq H \text{ for each index } i,$$

where $\|\cdot\|_2$ denotes the matrix spectral norm. Introduce the random matrix $\boldsymbol{Y} = \sum_i \boldsymbol{X}_i$, and let $v(\boldsymbol{Y})$ be the matrix variance statistic of the sum:

$$v(\boldsymbol{Y}) = \left\|\mathbb{E}[\boldsymbol{Y}^2]\right\| = \left\|\sum_i \mathbb{E}[\boldsymbol{X}_i^2]\right\|.$$

Then we have

$$\mathbb{E}\left[\|\boldsymbol{Y}\|_2\right] \leq \sqrt{2v(\boldsymbol{Y})\log N} + \frac{1}{3}L\log N. \tag{55}$$

Furthermore, for all $\epsilon \geq 0$.

$$P\left\{\|\boldsymbol{Y}\|_2 \geq \epsilon\right\} \leq N \cdot \exp\left(\frac{-\epsilon^2/2}{v(\boldsymbol{Y}) + H\epsilon/3}\right). \tag{56}$$

Proof. The proof of Lemma C.1 can be found in Theorem 6.6.1, § 6.6, (Tropp, 2015). □

- Next, we show how to apply Lemma C.1 to prove Theorem. 4.3.

1). Factorization of Approximation Error Matrix. With the constructed SM kernel matrix approximation, $\hat{\mathbf{K}}_{\text{sm}} = \Phi_{\text{sm}}(\mathbf{X})\Phi_{\text{sm}}(\mathbf{X})^\top$, where the random feature matrix $\Phi_{\text{sm}}(\mathbf{X}) = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]^\top \in \mathbb{R}^{N\times mL}$, we have the following approximation error matrix:

$$\mathbf{E} = \hat{\mathbf{K}}_{\text{sm}} - \mathbf{K}_{\text{sm}}. \tag{57}$$

We are going to show that $\mathbf{E}$ can be factorized as

$$\mathbf{E} = \sum_{i=1}^{m}\sum_{l=1}^{L/2}\mathbf{E}_l^{(i)} \tag{58}$$

where $\mathbf{E}_l^{(i)}$ is a sequence of independent, random, Hermitian matrices with dimension N.

Specifically, we define $\mathbf{Z}_l^{(i)}$ as

$$\mathbf{Z}_l^{(i)} = \left[\exp\left(2\pi j\mathbf{w}_l^{(i)\top}\mathbf{x}_1\right), \ldots, \exp\left(2\pi j\mathbf{w}_l^{(i)\top}\mathbf{x}_N\right)\right]^\top \in \mathbb{R}^{N\times 1}, \text{ where } \mathbf{w}_l^{(i)} \sim s_i(\mathbf{w}), \tag{59}$$

and we can show that

$$\begin{aligned}
[\hat{\mathbf{K}}_{\text{sm}}]_{h,g} &= \sum_{i=1}^{m}\frac{2\alpha_i}{L}\sum_{l=1}^{L/2}\cos\left(2\pi\mathbf{w}_l^{(i)\top}(\mathbf{x}_h - \mathbf{x}_g)\right) \\
&= \sum_{i=1}^{m}\sum_{l=1}^{L/2}\frac{2\alpha_i}{L}\operatorname{Re}\left(\exp\left(2\pi j\mathbf{w}_l^{(i)\top}(\mathbf{x}_h - \mathbf{x}_g)\right)\right) \\
&= \sum_{i=1}^{m}\sum_{l=1}^{L/2}\frac{2\alpha_i}{L}\operatorname{Re}\left(\left[\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right]_{h,g}\right)
\end{aligned} \tag{60}$$

where $\mathbf{Z}_l^{(i)*}$ is the conjugate transpose of $\mathbf{Z}_l^{(i)}$. Thus, we have $\hat{\mathbf{K}}_{\text{sm}} = \sum_{i=1}^{m}\sum_{l=1}^{L/2}\frac{2\alpha_i}{L}\operatorname{Re}(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*})$. Based on this factorization and Eq. (54) in Proposition 4.2, we have that

$$\mathbf{K}_{\text{sm}} = \sum_{i=1}^{m}\sum_{l=1}^{L/2}\frac{2\alpha_i}{L}\mathbb{E}[\operatorname{Re}(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*})].$$

Therefore, the approximation error matrix $\mathbf{E}$ can be factorized as $\mathbf{E} = \sum_{i=1}^{m}\sum_{l=1}^{L/2}\mathbf{E}_l^{(i)}$ where

$$\mathbf{E}_l^{(i)} = \frac{2\alpha_i}{L}\left(\mathrm{Re}(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}) - \mathbb{E}[\mathrm{Re}(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*})]\right) \tag{61}$$

is a sequence of independent, random, Hermitian matrices with dimension N that satisfy the condition of $\mathbb{E}[\mathbf{E}_l^{(i)}] = \mathbf{0}$. We next find the upper bound for $\|\mathbf{E}_l^{(i)}\|_2$.

2). Upper Bound for $\|\mathbf{E}_l^{(i)}\|_2$.

$$\begin{aligned}
\|\mathbf{E}_{l,i}\|_2 &= \frac{2\alpha_i}{L}\left\|\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right) - \mathbb{E}\left[\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right]\right\|_2 && \text{(62a)}\\
&\le \frac{2\alpha_i}{L}\left(\left\|\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right\|_2 + \left\|\mathbb{E}\left[\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right]\right\|_2\right) \qquad \text{(triangle inequality)} && \text{(62b)}\\
&\le \frac{2\alpha_i}{L}\left(\left\|\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right\|_2 + \mathbb{E}\left[\left\|\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right\|_2\right]\right) \qquad \text{(Jensen's inequality)} && \text{(62c)}\\
&\le \frac{2a}{L}\left(2N + 2N\right) && \text{(62d)}\\
&= \frac{2a}{L}4N && \text{(62e)}
\end{aligned}$$

where $a = \sqrt{\sum_{i=1}^{m}\alpha_i^2}$ and

$$\begin{aligned}
\boldsymbol{c}_l^{(i)} &= \left[\cos\left(2\pi\mathbf{w}_l^{(i)\top}\mathbf{x}_1\right), \ldots, \cos\left(2\pi\mathbf{w}_l^{(i)\top}\mathbf{x}_N\right)\right]^\top \in \mathbb{R}^{N\times 1}, && \text{(63a)}\\
\boldsymbol{s}_l^{(i)} &= \left[\sin\left(2\pi\mathbf{w}_l^{(i)\top}\mathbf{x}_1\right), \ldots, \sin\left(2\pi\mathbf{w}_l^{(i)\top}\mathbf{x}_N\right)\right]^\top \in \mathbb{R}^{N\times 1}, && \text{(63b)}\\
\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right) &= \boldsymbol{c}_l^{(i)}\boldsymbol{c}_l^{(i)\top} + \boldsymbol{s}_l^{(i)}\boldsymbol{s}_l^{(i)\top}, && \text{(63c)}
\end{aligned}$$

and the last inequality in Eq. (62), we use the fact that

$$\left\|\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right\|_2 = \sup_{\|\boldsymbol{v}\|_2^2=1}\boldsymbol{v}^\top\left(\boldsymbol{c}_l^{(i)}\boldsymbol{c}_l^{(i)\top} + \boldsymbol{s}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\boldsymbol{v} \le 2N.$$

Next, we are going to bound the variance, $\left\|\sum_{i=1}^{m}\sum_{l=1}^{L/2}\mathbb{E}[(\mathbf{E}_l^{(i)})^2]\right\|_2$.

3). Upper Bound for the Variance, $\left\|\sum_{i=1}^{m}\sum_{l=1}^{L/2}\mathbb{E}[(\mathbf{E}_l^{(i)})^2]\right\|_2$. We first have the following bound:

$$\begin{aligned}
\frac{L^2}{4\alpha_i^2}\mathbb{E}\left[\left(\mathbf{E}_l^{(i)}\right)^2\right] &= \mathbb{E}\left[\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)^2\right] - \left(\mathbb{E}\left[\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right]\right)^2 && \text{(64a)}\\
&\preccurlyeq \mathbb{E}\left[\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)^2\right] && \text{(64b)}\\
&= \mathbb{E}\left[\left(\boldsymbol{c}_l^{(i)\top}\boldsymbol{c}_l^{(i)}\right)\boldsymbol{c}_l^{(i)}\boldsymbol{c}_l^{(i)\top} + \left(\boldsymbol{s}_l^{(i)\top}\boldsymbol{s}_l^{(i)}\right)\boldsymbol{s}_l^{(i)}\boldsymbol{s}_l^{(i)\top} + \left(\boldsymbol{s}_l^{(i)\top}\boldsymbol{c}_l^{(i)}\right)\left(\boldsymbol{s}_l^{(i)}\boldsymbol{c}_l^{(i)\top} + \boldsymbol{c}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\right] && \text{(64c)}\\
&\preccurlyeq N\mathbb{E}\left[\boldsymbol{c}_l^{(i)}\boldsymbol{c}_l^{(i)\top} + \boldsymbol{s}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right] + \mathbb{E}\left[\left(\boldsymbol{s}_l^{(i)\top}\boldsymbol{c}_l^{(i)}\right)\left(\boldsymbol{s}_l^{(i)}\boldsymbol{c}_l^{(i)\top} + \boldsymbol{c}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\right] && \text{(64d)}\\
&= N\mathbb{E}\left[\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right] + \mathbb{E}\left[\left(\boldsymbol{s}_l^{(i)\top}\boldsymbol{c}_l^{(i)}\right)\left(\boldsymbol{s}_l^{(i)}\boldsymbol{c}_l^{(i)\top} + \boldsymbol{c}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\right] && \text{(64e)}
\end{aligned}$$

where the notation $\mathbf{A} \preccurlyeq \mathbf{B}$ denotes that $\mathbf{B} - \mathbf{A}$ is a positive semi definite (PSD) matrix, and the inequality in Eq. (64b) holds due to the fact that $\left(\mathbb{E}\left[\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right]\right)^2$ is a PSD matrix. The inequality in Eq. (64d) holds because

$$\begin{aligned}
&N\mathbb{E}\left[\boldsymbol{c}_l^{(i)}\boldsymbol{c}_l^{(i)\top} + \boldsymbol{s}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right] - \mathbb{E}\left[\left(\boldsymbol{c}_l^{(i)\top}\boldsymbol{c}_l^{(i)}\right)\boldsymbol{c}_l^{(i)}\boldsymbol{c}_l^{(i)\top} + \left(\boldsymbol{s}_l^{(i)\top}\boldsymbol{s}_l^{(i)}\right)\boldsymbol{s}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right]\\
&= \mathbb{E}\left[\left(\boldsymbol{s}_l^{(i)\top}\boldsymbol{s}_l^{(i)}\right)\boldsymbol{c}_l^{(i)}\boldsymbol{c}_l^{(i)\top} + \left(\boldsymbol{c}_l^{(i)\top}\boldsymbol{c}_l^{(i)}\right)\boldsymbol{s}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right] \quad \ldots \quad \left[\text{due to } \left(\boldsymbol{c}_l^{(i)\top}\boldsymbol{c}_l^{(i)} + \boldsymbol{s}_l^{(i)\top}\boldsymbol{s}_l^{(i)}\right) = N\right]
\end{aligned} \tag{65}$$

is a PSD matrix.

Then we are able to bound the variance, $\left\|\sum_{i=1}^{m}\sum_{l=1}^{L/2}\mathbb{E}[(\mathbf{E}_l^{(i)})^2]\right\|_2$, as

$$
\begin{aligned}
&\left\|\sum_{i=1}^{m}\sum_{l=1}^{L/2}\mathbb{E}[(\mathbf{E}_l^{(i)})^2]\right\|_2 \\
&\leq \left\|\sum_{i=1}^{m}\sum_{l=1}^{L/2}\frac{4\alpha_i^2}{L^2}\left(N\mathbb{E}\left[\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right]+\mathbb{E}\left[\left(\boldsymbol{s}_l^{(i)\top}\boldsymbol{c}_l^{(i)}\right)\left(\boldsymbol{s}_l^{(i)}\boldsymbol{c}_l^{(i)\top}+\boldsymbol{c}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\right]\right)\right\|_2 \\
&\leq \frac{2a}{L}\left\|\sum_{i=1}^{m}\alpha_i\left(N\mathbb{E}\left[\mathrm{Re}\left(\mathbf{Z}_l^{(i)}\mathbf{Z}_l^{(i)*}\right)\right]+\mathbb{E}\left[\left(\boldsymbol{s}_l^{(i)\top}\boldsymbol{c}_l^{(i)}\right)\left(\boldsymbol{s}_l^{(i)}\boldsymbol{c}_l^{(i)\top}+\boldsymbol{c}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\right]\right)\right\|_2 \\
&\leq \frac{2a}{L}\left(N\left\|\mathbf{K}_{\mathrm{sm}}\right\|_2+\sum_{i=1}^{m}\alpha_i\left\|\mathbb{E}\left[\left(\boldsymbol{s}_l^{(i)\top}\boldsymbol{c}_l^{(i)}\right)\left(\boldsymbol{s}_l^{(i)}\boldsymbol{c}_l^{(i)\top}+\boldsymbol{c}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\right]\right\|_2\right) \quad \text{(triangle inequality)} \\
&\leq \frac{2a}{L}\left(N\left\|\mathbf{K}_{\mathrm{sm}}\right\|_2+\sum_{i=1}^{m}\alpha_i\mathbb{E}\left[\left\|\left(\boldsymbol{s}_l^{(i)\top}\boldsymbol{c}_l^{(i)}\right)\left(\boldsymbol{s}_l^{(i)}\boldsymbol{c}_l^{(i)\top}+\boldsymbol{c}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\right\|_2\right]\right) \quad \text{(Jensen's inequality)} \\
&\leq \frac{2a}{L}\left(N\left\|\mathbf{K}_{\mathrm{sm}}\right\|_2+\frac{N}{2}\sum_{i=1}^{m}\alpha_i\mathbb{E}\left[\left\|\left(\boldsymbol{s}_l^{(i)}\boldsymbol{c}_l^{(i)\top}+\boldsymbol{c}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\right\|_2\right]\right) \quad \left(|\boldsymbol{s}_l^{(i)\top}\boldsymbol{c}_l^{(i)}|\leq\frac{N}{2}\right) \\
&\leq \frac{2aN}{L}\left(\left\|\mathbf{K}_{\mathrm{sm}}\right\|_2+\frac{N}{2}a\sqrt{m}\right)
\end{aligned}
\tag{66}
$$

where the last inequality is because that

$$
\mathbb{E}\left[\left\|\left(\boldsymbol{s}_l^{(i)}\boldsymbol{c}_l^{(i)\top}+\boldsymbol{c}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\right\|_2\right]=\sup_{\|\boldsymbol{v}\|_2^2=1}\mathbb{E}\left[\left\|\boldsymbol{v}^\top\left(\boldsymbol{s}_l^{(i)}\boldsymbol{c}_l^{(i)\top}+\boldsymbol{c}_l^{(i)}\boldsymbol{s}_l^{(i)\top}\right)\boldsymbol{v}\right\|_2\right]\leq N, \tag{67}
$$

and $\sum_{i=1}^{m}\alpha_i\leq a\sqrt{m}$ by the Cauchy–Schwarz inequality.

4). Final Result. We next can apply the derived upper bounds, Eqs. (62) and (66), to the H and $v(\boldsymbol{Y})$ in Lemma C.1,

$$
P\left(\left\|\hat{\mathbf{K}}_{\mathrm{sm}}-\mathbf{K}_{\mathrm{sm}}\right\|_2\geq\epsilon\right)\leq N\exp\left(\frac{-3\epsilon^2L}{2Na\left(6\left\|\mathbf{K}_{\mathrm{sm}}\right\|_2+3Na\sqrt{m}+8\epsilon\right)}\right) \tag{68}
$$

which completes the proof of Theorem 4.3 □

D. Extended Related Work

VAEs. As a facet of model collapse, the posterior collapse in variational autoencoders (VAEs) occurs when the variational posterior distribution of the latent variables approaches to the prior, resulting in a failure to exploit the valuable knowledge embedded in the observed data. Numerous approaches have been proposed to tackle this issue, with the most commonly embraced heuristic solution being the annealing of the KL term in the ELBO objective (Bowman et al., 2016; Sønderby et al., 2016). Specifically, Gulrajani et al. (2016) suggest that posterior collapse is induced by the high-capacity decoder, which can map any noise vector to the desired target $\mathbf{X}$. Motivating by this hypothesis, Gulrajani et al. (2016); Yang et al. (2017) propose reducing the capacity of the decoder for better representations, albeit at the cost of a reduction in generative capability. Another line of works, such as (Lucas et al., 2019; Wang & Liu, 2022; Wang et al., 2021), claims that posterior collapse is partially attributed to the suboptimal selection of likelihood variances, aligning with our findings in the context of the Bayesian non-parametric GPLVM. Nevertheless, despite the alignment of these works addressing posterior collapse with our findings, the primary objective in VAEs is to improve generative capacity, deviating from our emphasis, which lies in recovering compact and informative latent representations.

GPLVMs. This paper focuses on the GPLVMs (Lawrence, 2005), which apply GP for modeling the nonlinear function in LVM, obviating the need to optimize substantial neural network parameters while alleviating overfitting and generalization

Table 3: A summary of relevant LVMs, where N and M denote # observations and the observation dimension, respectively, while U, m, L represent # inducing points, # mixture components in SM kernel, and the dimension of random features, respectively.

Model	Scalable model	Advanced kernel	Probabilistic mapping	Bayesian inference of latent variables	Computational complexity	# parameters	Reference
VAE	✓	-	✗	✓	-	-	Kingma & Welling (2019)
NBVAE	✓	-	✗	✓	-	-	Zhao et al. (2020)
DCA	✓	-	✗	✓	-	-	Eraslan et al. (2019)
CVQ-VAE	✓	-	✗	✓	-	-	Zheng & Vedaldi (2023)
GPLVM	✗	✗	✓	✗	$\mathcal{O}(N^3)$	$N(N+Q)+C$	Lawrence (2005)
BGPLVM	✓	✗	✓	✓	$\mathcal{O}(NU^2)$	$Q(1+U+N+NQ)+C$	Titsias & Lawrence (2010)
GPLVM-SVI	✓	✗	✓	✓	$\mathcal{O}(MU^3)$	$U(M+MU+Q)+2NQ+C$	Lalchand et al. (2022)
RFLVM	✗	✓	✓	✗	$\mathcal{O}(NM^2L)$	$NQ+L(Q+M+\frac{Q^2}{2})+2M+C$	Zhang et al. (2023)
*advised*RFLVM	✓	✓	✓	✓	$\mathcal{O}(N(mL)^2)$	$Q(N+NQ+2m)+m+C$	**This work**

issues (Wilson & Izmailov, 2020). The seminal work of GPLVM was proposed by Lawrence (2005). Subsequently, Titsias & Lawrence (2010) introduced the Bayesian formulation of the GPLVM, which variationally integrated out latent variables. However, this model exhibits computational efficiency only with specific preliminary kernel functions, such as the radial basis function (RBF) kernel (Rasmussen & Williams, 2006), imposing significant constraints on the model capacity of the GPLVM and leading to model collapse. Recent endeavors have focused on enhancing the scalability and flexibility of the GPLVM (Lalchand et al., 2022; de Souza et al., 2021), as well as ensuring compatibility with various likelihoods (Ramchandran et al., 2021). Despite the relevance of these endeavors, the inference of these models relies on inducing points-based sparse GP (Titsias, 2009). This necessitates optimizing additional inducing points, leading to increased computational burden and the risk of getting stuck in suboptimal solutions. Consequently, despite the enhanced model capability, these models often face challenges in achieving their theoretical potential to address model collapse.

E. Experiment Details

E.1. Data Descriptions and Preprocessing

We first describe the detailed parameter settings for the two synthetic S-shaped datasets used in § 6.2. The datasets are generated from a GPLVM with different kernel configurations, which are listed below:

- Dataset with **RBF** kernel:
$$k_{\text{rbf}}(\mathbf{x}, \mathbf{x}') = \ell_o \exp\left(-\frac{(\mathbf{x}-\mathbf{x}')^2}{2\ell_l^2}\right), \tag{69}$$
with outputscale $\ell_o = 1$ and lengthscale $\ell_l = 1$.
- Dataset with a hybrid (**RBF+periodic**) kernel:
$$k_{\text{hybrid}}(\mathbf{x}, \mathbf{x}') = k_{\text{rbf}}(\mathbf{x}, \mathbf{x}') + k_{\text{periodic}}(\mathbf{x}, \mathbf{x}'), \tag{70a}$$
$$k_{\text{rbf}}(\mathbf{x}, \mathbf{x}') = \ell_o \exp\left(-\frac{(\mathbf{x}-\mathbf{x}')^2}{2\ell_l^2}\right), \qquad \text{with } \ell_o = 0.5, \ell_l = 1; \tag{70b}$$
$$k_{\text{periodic}}(\mathbf{x}, \mathbf{x}') = \ell_o \exp\left(-\frac{2\sin^2\left(\frac{\mathbf{x}-\mathbf{x}'}{p}\right)}{\ell_l^2}\right), \qquad \text{with } \ell_o = 0.5, \ell_l = 1, p = 4.5. \tag{70c}$$

Next, we offer a comprehensive introduction to real-world datasets and downsample large-scale datasets to a smaller size to accommodate the high computational complexity in RFLVM (Gundersen et al., 2021).

- BRIDGES: We recorded the daily count of bicycles crossing each of the four East River bridges in New York City[5]. To assign labels, we categorized the data into weekday versus weekend, treating them as binary labels due to the absence of explicit labels in the dataset. This categorization was made based on the understanding that weekdays and weekends are inherently linked to variations in bicycle counts.

[5] https://data.cityofnewyork.us/Transportation/Bicycle-Counts-for-East-River-Bridges/gua4-p9wg

- CIFAR-10: To create a final dataset of size 2000, we subsampled 400 images from each class within [airplane, automobile, bird, cat, deer]. These images were further resized from 32×32 pixels to 20×20 pixels and converted to grayscale. Test performance of different models on the full dataset can be found in §. E.4.4.
- MNIST: The dataset size was reduced by randomly selecting 1000 images. Test performance of different models on the full dataset can be found in §. E.4.4.
- MONTREAL: We analyze the daily count of cyclists on eight bicycle lanes in Montreal[6]. Given the absence of explicit labels, we employed the four seasons as labels, as seasonality is correlated with bicycle counts.
- NEWSGROUPS The 20 Newsgroups Dataset[7] was employed, with classes limited to *comp.sys.mac.hardware*, *sci.med*, and *alt.atheism*. The vocabulary was constrained to words with document frequencies falling within the range of $10 - 90\%$.
- YALE: The Yale Faces Dataset[8] was employed in our study, with subject IDs utilized as labels.
- BRENDAN: This dataset comprises 2000 images, each with a size of 20×28 pixels, depicting the face of Brendan[9].

E.2. Benchmark Methods Descriptions

- **PCA, LDA, Isomap:** PCA (Wold et al., 1987), LDA (Blei et al., 2003), and Isomap (Balasubramanian & Schwartz, 2002) were implemented utilizing the `sklearn.decomposition` module within the `scikit-learn` library (Buitinck et al., 2013).
- **HPF:** The implementation of HPF (Gopalan et al., 2015) is based on the `hpfrec` library[10].
- **BGPLVM:** We utilized the `BayesianGPLVMMiniBatch` implementation in the `GPy` library[11], which is an inducing points-based method (Titsias, 2009).
- **GPLVM-SVI:** We used the official source code based on `GPyTorch`[12]. We also extended the GPLVM-SVI (Lalchand et al., 2022) to accommodate the SM kernel function, but this modification could result in further performance degradation.
- **VAE:** The implementation of the VAE (Kingma & Welling, 2013) was built upon the example code provided by the `pytorch` library[13].
- **NBVAE, DCA, CVQ-VAE, RFLVM:** All implementations for those algorithms adhere to the corresponding official code libraries available online [14] [15] [16] [17].

[6] http://donnees.ville.montreal.qc.ca/dataset/f170fecc-18db-44bc-b4fe-5b0b6d2c7297/resource/64c26fd3-0bdf-45f8-92c6-715a9c852a7b
[7] http://qwone.com/~jason/20Newsgroups/
[8] http://vision.ucsd.edu/content/yale-face-database
[9] https://cs.nyu.edu/~roweis/data/frey_rawface.mat
[10] https://github.com/david-cortes/hpfrec
[11] http://github.com/SheffieldML/GPy
[12] https://github.com/vr308/Generalised-GPLVM
[13] https://github.com/pytorch/examples/blob/main/vae/main.py
[14] https://github.com/ethanhezhao/NBVAE
[15] https://github.com/theislab/dca
[16] https://github.com/lyndonzheng/CVQ-VAE
[17] https://github.com/gwgundersen/rflvm

E.3. Default Hyperparameter Configurations

Table 4: Default hyperparameter settings.

PARAMETER	VALUE
# MIXTURE DENSITIES IN SM KERNEL (m)	2
DIM. OF RANDOM FEATURE (L)	50
DIM. OF LATENT SPACE (Q)	2
OPTIMIZER	ADAM (KINGMA & BA, 2014)
LEARNING RATE	0.005
BETA	$(0.9, 0.99)$
# ITERATIONS	10000

Tab. 4 displays the default hyperparameter settings employed by *advised*RFLVM. These hyperparameter settings are employed in the majority of experiments, with the exception of the experiment corresponding to the left side of Fig. 2. In this case, the dimensionality of the latent space is configured to 50 to intuitively illustrate the variability in the number of zero-columns within the latent variables.

E.4. Additional Results

E.4.1. S-SHAPED LATENT MANIFOLD ESTIMATION

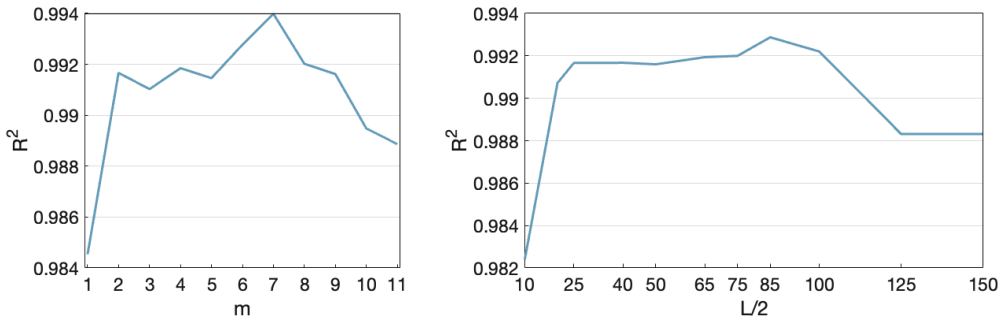
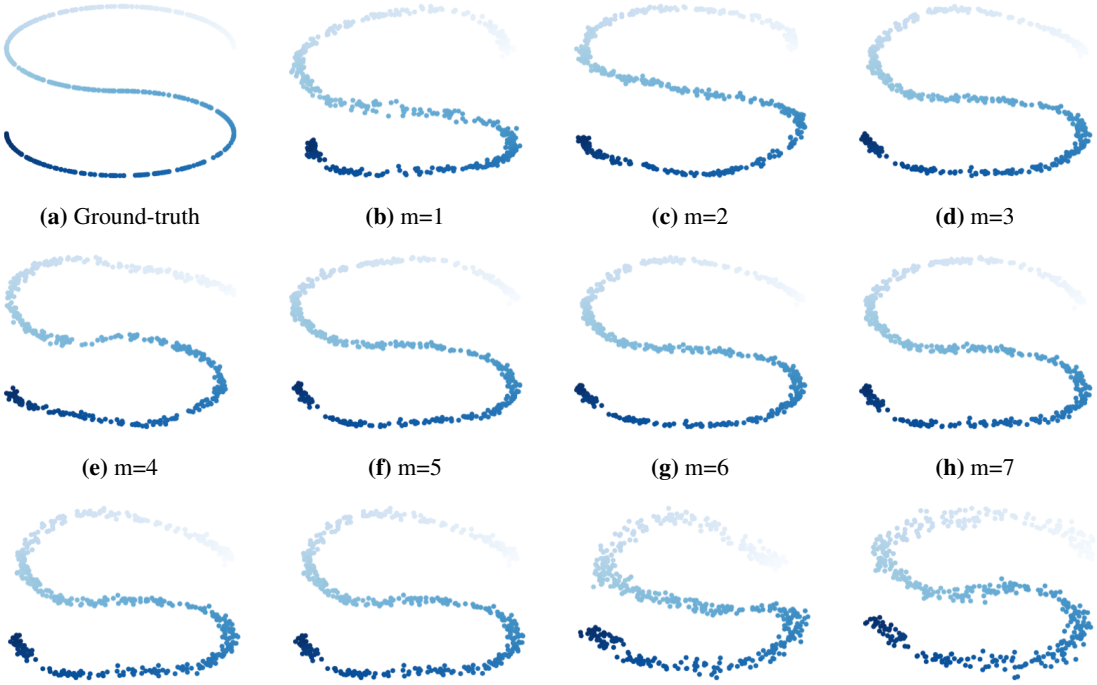


Figure 4: (Left) R^2 against the number of mixture densities in SM kernel (m). **(Right)** R^2 versus the dimensionality of the random feature $(L/2)$.

To validate the rationale behind our parameter selection, this section presents an evaluation of *advised*RFLVM, showcasing its performance in manifold visualization and R^2 scores across various values of m and $L/2$. Fig. 4 depicts the *advised*RFLVM performance in terms of R^2 scores. Additionally, visualizations of the latent manifold recovered by *advised*RFLVM are provided in Fig. 5 and Fig. 6. The results affirm that opting for $m = 2$ and $L/2 = 50$ ensures the lowest computational complexity while maintaining comparable performance.

E.4.2. MISSING DATA IMPUTATION

To intuitively showcase the capability of *advised*RFLVM in the task of missing data imputation, visualizations of the reconstructed observed data are presented in Fig. 7 and Fig. 8, underscoring its superior ability to restore missing pixels.

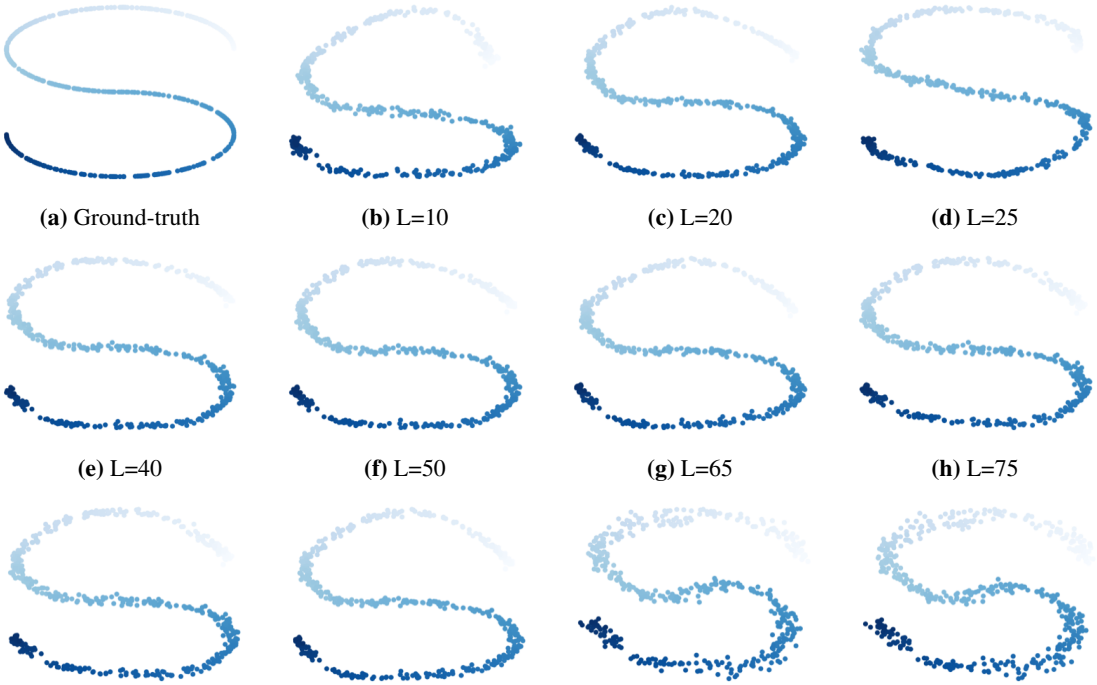


(a) Ground-truth **(b)** m=1 **(c)** m=2 **(d)** m=3

(e) m=4 **(f)** m=5 **(g)** m=6 **(h)** m=7

(i) m=8 **(j)** m=9 **(k)** m=10 **(l)** m=11

Figure 5: Latent manifold learning results with $L/2 = 25$ and different m

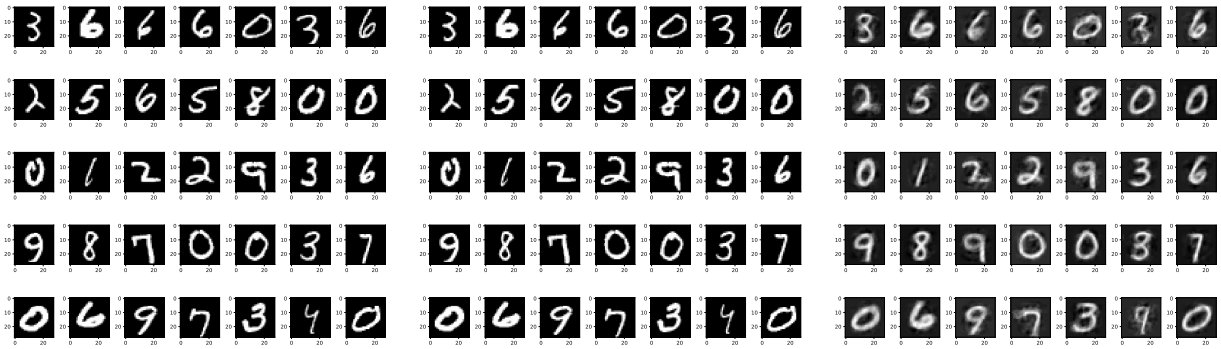


(a) Ground-truth **(b)** L=10 **(c)** L=20 **(d)** L=25

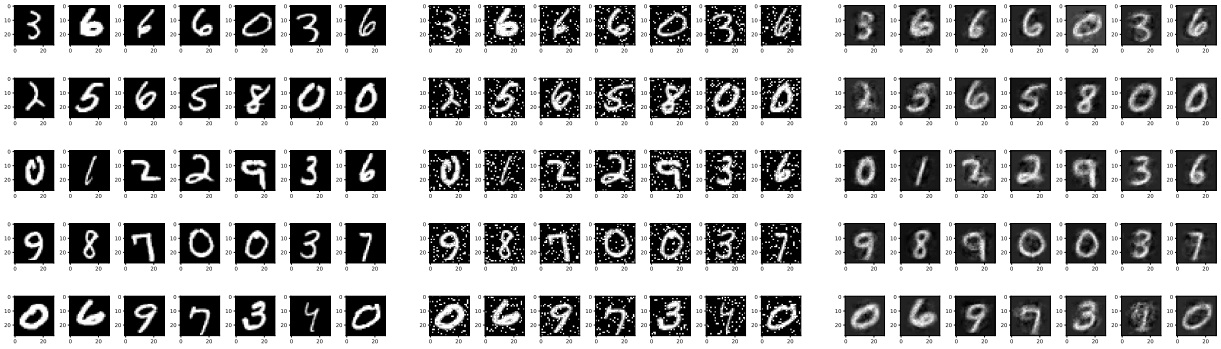
(e) L=40 **(f)** L=50 **(g)** L=65 **(h)** L=75

(i) L=85 **(j)** L=100 **(k)** L=125 **(l)** L=150

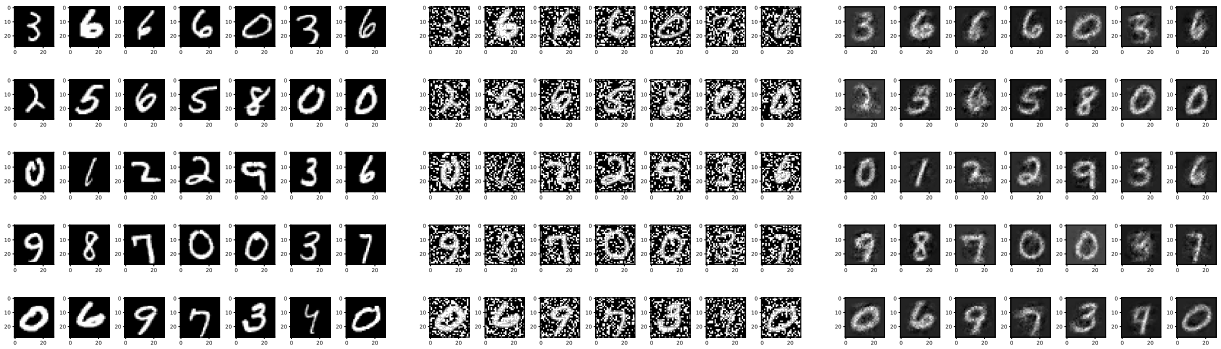
Figure 6: Latent manifold learning results with $m = 2$ and different L.



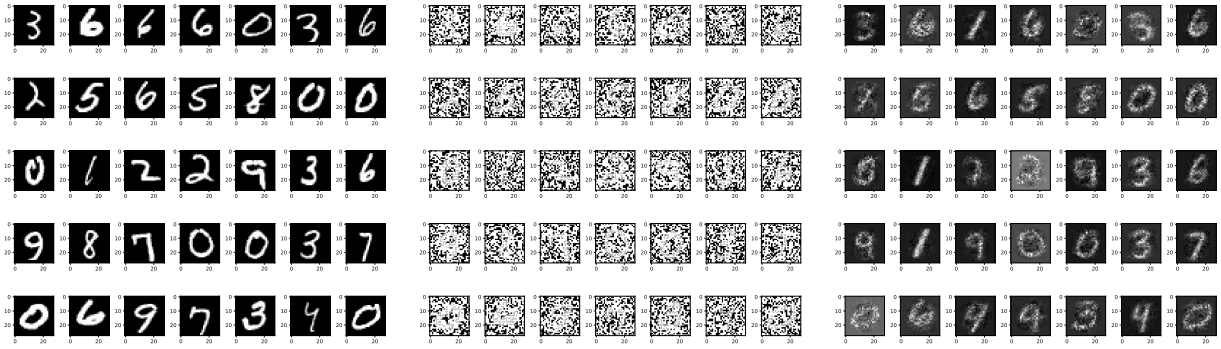
(a) MNIST reconstruction task with 0% missing pixels.



(b) MNIST reconstruction task with 10% missing pixels.

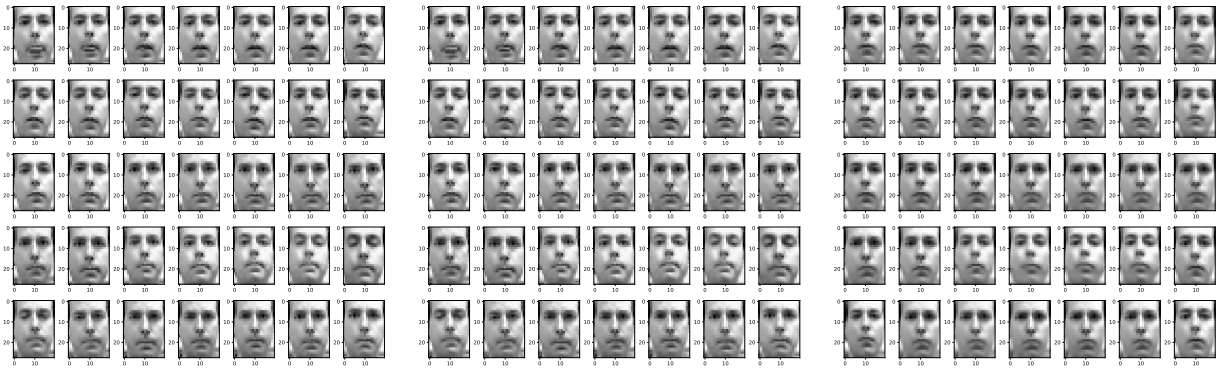


(c) MNIST reconstruction task with 30% missing pixels.



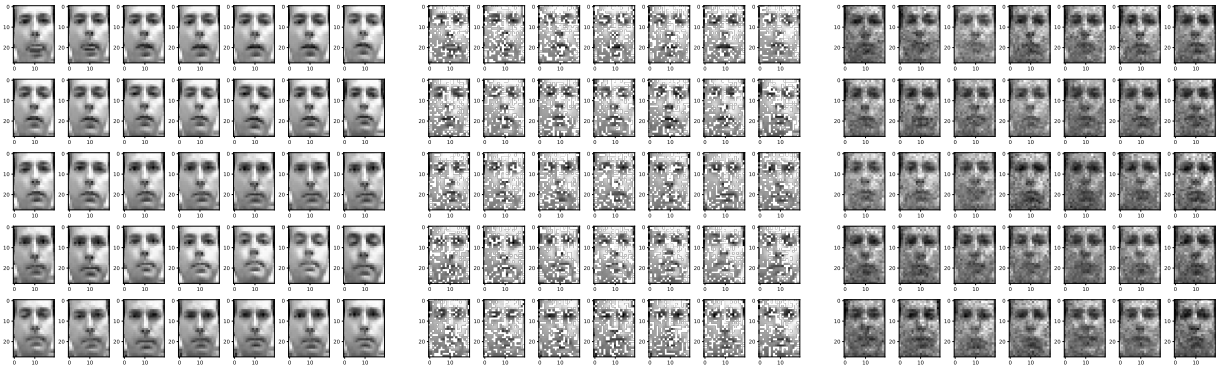
(d) MNIST reconstruction task with 60% missing pixels.

Figure 7: MNIST reconstruction task with missing pixels. From left to right: Ground truth, training images, reconstructions

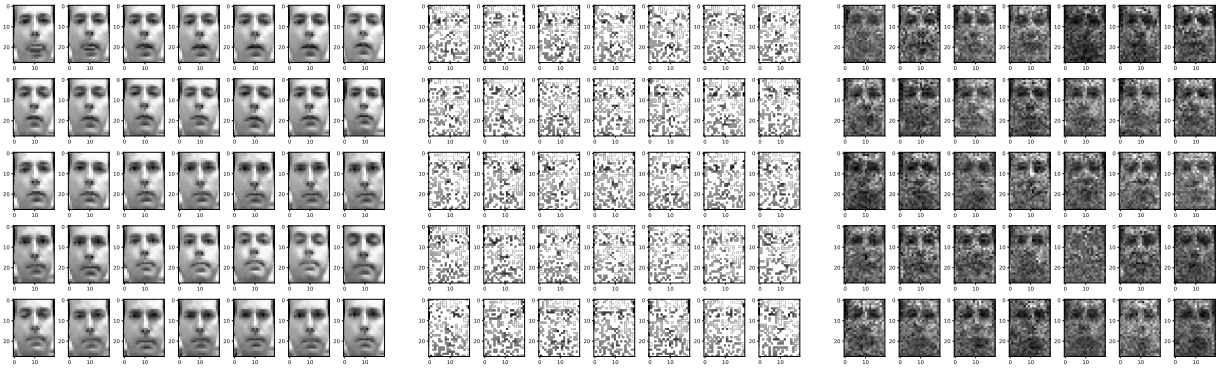


(a) Brendan faces reconstruction task with 0% missing pixels.

(b) Brendan faces reconstruction task with 10% missing pixels.



(c) Brendan faces reconstruction task with 30% missing pixels.



(d) Brendan faces reconstruction task with 60% missing pixels.

Figure 8: Brendan faces reconstruction task with missing pixels. From left to right: Ground truth, training images, reconstructions

Table 5: KNN classification accuracy using different numbers of nearest neighbors (K values). We ran this classification using 5-fold cross validation.

METHODS	VAE										*advised*RFLVM									
K-VALUE	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
BRIDGES	0.780	0.794	0.766	0.789	0.794	0.799	0.804	0.808	0.780	0.776	0.846	0.846	**0.902**	**0.902**	**0.907**	**0.888**	**0.893**	**0.898**	0.879	**0.903**
CIFAR	0.256	0.260	0.266	0.274	0.280	0.282	0.291	0.296	0.293	0.300	**0.300**	**0.310**	**0.309**	**0.340**	**0.335**	**0.342**	**0.350**	**0.357**	**0.365**	**0.358**
MNIST	0.631	0.614	0.657	0.646	0.677	0.670	0.674	0.671	0.673	0.669	**0.801**	**0.780**	**0.819**	**0.824**	**0.823**	**0.813**	**0.812**	**0.800**	**0.802**	**0.800**
MONTREAL	0.649	0.655	0.683	0.662	0.699	0.705	0.718	0.718	0.696	0.712	**0.799**	0.759	0.796	0.802	**0.815**	0.787	0.777	0.755	0.768	0.759
YALE	0.667	0.667	0.672	0.667	0.636	0.630	0.600	0.576	0.558	0.552	**0.757**	**0.703**	**0.721**	**0.745**	**0.727**	**0.691**	**0.685**	**0.673**	**0.655**	**0.642**
NEWSGROUPS	0.381	0.389	0.384	0.397	0.402	0.409	0.406	0.410	0.399	0.404	0.401	**0.403**	0.399	0.412	**0.419**	0.408	0.414	0.414	**0.426**	0.424
METHODS	BGPLVM										RFLVM									
K-VALUE	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
BRIDGES	0.836	0.808	0.813	0.794	0.818	0.808	0.837	0.832	0.832	0.813	**0.860**	**0.859**	0.869	0.892	0.869	0.869	0.883	0.888	**0.883**	0.887
CIFAR	0.262	0.278	0.279	0.293	0.291	0.295	0.282	0.290	0.288	0.294	0.270	0.288	0.288	0.288	0.306	0.310	0.320	0.326	0.331	0.333
MNIST	0.573	0.585	0.611	0.622	0.627	0.636	0.640	0.645	0.652	0.648	0.592	0.567	0.591	0.603	0.634	0.624	0.638	0.634	0.633	0.633
MONTREAL	0.752	0.771	0.759	0.771	0.787	0.778	0.800	0.800	0.793	0.787	0.778	**0.818**	**0.819**	**0.844**	0.809	**0.831**	**0.806**	**0.815**	**0.806**	**0.790**
YALE	0.558	0.515	0.545	0.558	0.527	0.503	0.503	0.467	0.485	0.461	0.576	0.515	0.612	0.564	0.564	0.576	0.576	0.558	0.582	0.576
NEWSGROUPS	0.388	0.374	0.406	0.397	0.392	0.395	0.404	0.397	0.396	0.403	**0.404**	0.394	**0.411**	**0.425**	0.412	**0.413**	**0.417**	**0.424**	0.416	**0.426**

Table 6: KNN classification accuracy using different numbers of nearest neighbors (K values) on larger datasets. We ran this classification using 5-fold cross validation.

METHODS	VAE										*advised*RFLVM									
K-VALUE	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
F-CIFAR	0.157	0.151	0.157	0.166	0.174	0.178	0.180	0.187	0.188	0.190	**0.172**	**0.161**	**0.177**	**0.181**	**0.194**	**0.199**	**0.203**	**0.209**	**0.213**	**0.214**
FD-CIFAR	0.263	0.266	0.279	0.285	0.293	0.297	0.302	0.304	0.309	0.312	**0.321**	**0.323**	**0.344**	**0.359**	**0.368**	**0.370**	**0.377**	**0.384**	**0.390**	**0.391**
F-MNIST	0.728	0.728	0.756	0.766	0.774	0.775	0.778	0.782	0.782	0.783	**0.794**	**0.796**	**0.831**	**0.838**	**0.845**	**0.847**	**0.850**	**0.852**	**0.851**	**0.852**
METHODS	BGPLVM										Isomap									
K-VALUE	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
F-CIFAR	0.138	0.132	0.140	0.145	0.154	0.156	0.159	0.161	0.162	0.163	0.144	0.142	0.147	0.157	0.159	0.163	0.165	0.168	0.170	0.173
FD-CIFAR	0.250	0.260	0.258	0.264	0.279	0.277	0.281	0.285	0.287	0.287	0.264	0.270	0.279	0.287	0.291	0.292	0.296	0.301	0.305	0.305
F-MNIST	0.414	0.420	0.433	0.449	0.455	0.464	0.466	0.470	0.473	0.474	0.456	0.468	0.493	0.504	0.514	0.524	0.529	0.534	0.535	0.540
METHODS	NBVAE										CVQ-VAE									
K-VALUE	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
F-CIFAR	0.134	0.137	0.140	0.147	0.152	0.157	0.156	0.155	0.161	0.162	0.101	0.098	0.099	0.101	0.102	0.102	0.101	0.100	0.098	0.096
FD-CIFAR	0.252	0.248	0.255	0.264	0.273	0.277	0.282	0.287	0.291	0.292	0.203	0.201	0.200	0.199	0.201	0.200	0.201	0.199	0.197	0.200
F-MNIST	0.502	0.502	0.533	0.548	0.557	0.566	0.571	0.577	0.579	0.582	0.104	0.107	0.107	0.104	0.105	0.106	0.102	0.103	0.103	0.103

E.4.3. KNN CLASSIFICATION ACCURACY WITH VARYING K

We have presented the KNN results with ten different choices of K in Tab. 5, wherein the setting of $K = 1$ aligns with the configuration employed in (Gundersen et al., 2021). The simulation results consistently demonstrate the superiority of our method over the benchmarks regardless of the K values across most datasets. In those exception cases, *advised*RFLVM still achieves very comparable performance with RFLVM on some relatively simple datasets, e.g., BRIDGES, MONTREAL, and NEWSGROUP datasets.

E.4.4. LARGER DATASETS EXTENSION

To ensure equitable evaluation of deep learning methods, such as various VAE variants, we conducted comprehensive comparisons on larger datasets, including the full MNIST and CIFAR datasets. The results are summarized in Table 6, where F-CIFAR and F-MNIST represent the full CIFAR and MNIST datasets, respectively, and FD-CIFAR denotes the full CIFAR dataset with each image downsampled to 20×20 pixels. Our empirical results demonstrate significant performance improvement for both VAE and our *advised*RFLVM when applied to larger datasets. Notably, *advised*RFLVM consistently outperforms the other benchmarks across datasets of varying sizes, highlighting its superiority over state-of-the-art variants irrespective of the dataset size.