Nonparametric Automatic Differentiation Variational Inference with Spline Approximation

Yuda Shao
University of Virginia

Shan Yu[†]
University of Virginia

Tianshu Feng[†]
George Mason University

Abstract

Automatic Differentiation Variational Inference (ADVI) is efficient in learning probabilistic models. Classic ADVI relies on the parametric approach to approximate the posterior. In this paper, we develop a spline-based nonparametric approximation approach that enables flexible posterior approximation for distributions with complicated structures, such as skewness, multimodality, and bounded support. Compared with widely-used nonparametric variational inference methods, the proposed method is easy to implement and adaptive to various data structures. By adopting the spline approximation, we derive a lower bound of the importance weighted autoencoder and establish the asymptotic consistency. Experiments demonstrate the efficiency of the proposed method in approximating complex posterior distributions and improving the performance of generative models with incomplete data.

1 INTRODUCTION

Variational Inference (VI) is widely used in data representation (Kingma and Welling, 2013; Zhang et al., 2018), graphical models (Wainwright et al., 2008), among others. VI approximates intractable distributions by minimizing the divergence between the true posterior and a chosen distribution family, aiming to identify an optimal distribution within this family. Unlike methods like Markov chain Monte Carlo (MCMC) sampling, VI is recognized for its computational efficiency and explicit distribution form (Blei et al., 2017).

[†]Corresponding authors.

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

Contemporary VI-based methods such as variational autoencoder (VAE) (Kingma and Welling, 2013) have garnered interest for learning representations of complex, high-dimensional data across fields like bioinformatics (Kopf et al., 2021), geoscience (Chen et al., 2022), and finance (Bergeron et al., 2022).

Automatic Differentiation Variational Inference (ADVI) (Kucukelbir et al., 2017) is a popular approach to derive variational inference algorithms for complex probabilistic models. Classic ADVI methods often adopt a parametric approach, approximating intractable posterior distributions with distributions from a specific probability distribution family (e.g., Gaussian distribution). However, it is limited to distributions allowing the reparametrization trick to calculate and backpropagate the gradient of the joint likelihood. Additionally, misspecified parametric assumptions can impair ADVI's efficacy to handle multimodal or skewed posteriors.

Recent studies show that more flexible posterior approximations usually result in better performance (Han et al., 2016; Kobyzev et al., 2020). In this paper, we aim to design a new type of variational inference based on spline approximation, named Spline Automatic Differentiation Variational Inference (S-ADVI), to improve the flexibility of posterior approximation while being interpretable. Spline approximation is an effective nonparametric tool for density estimation (Gu and Qiu, 1993). Theoretically, an arbitrary smooth density function can be well-approximated via weighted summation of a given sequence of spline bases. The shapes of spline bases are pre-specified and fixed, and the shapes of posterior distributions can be uniquely represented via the vector of spline coefficients. This property allows the assessment of the structure of posterior distributions and the interpretation of the latent representations. Consequently, the proposed S-ADVI achieves a balance of flexibility and parsimony.

The proposed S-ADVI holds several merits over existing nonparametric methods. First, a major limitation of nonparametric variational methods is the difficulty in providing theoretical guidance to recover the true posterior distribution. This paper theoretically inves-

tigates the asymptotic properties of the importance weighted autoencoder (IWAE), as well as the Kullback–Leibler (KL) divergence of spline approximations from the true posterior. Second, contrasted with other ADVI-based methods requiring pre-specified transformation to approximate distributions with bounded support, our approach simultaneously estimates the distribution support boundary. This adaptive boundary implementation enhances the accuracy and robustness of the model, ensuring superior performance. Last, the streamlined structure of the proposed method facilitates straightforward implementation and adaptability to various data structures.

In summary, our major contributions are: (i) We design a novel nonparametric variational inference framework, S-ADVI, based on spline approximation to improve the flexibility of posterior approximation; (ii) Theoretical properties are established on the lower bound of IWAE and variational approximation errors of the proposed S-ADVI method; (iii) S-ADVI represents posterior distributions with deterministic vectors, allowing the assessment of shapes of posterior distributions and the interpretation of latent representation.

2 BACKGROUND

2.1 Variational Inferences

Let $\boldsymbol{x}$ be the observed variables and $\boldsymbol{z}$ be the latent variables. We consider the joint distribution $p_\theta(\boldsymbol{x}, \boldsymbol{z})$ for some parameter θ, the generative model defined over the variables (Kingma and Welling, 2013). Learning θ typically requires the maximization of the marginal distribution of $\boldsymbol{x}$: $p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$, where $p(\boldsymbol{z})$ is the prior of $\boldsymbol{z}$, and $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is the conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{z}$.

Generally, the marginal likelihood function is intractable for flexible generative models. In VI, one common solution is to approximate the posterior $p(\boldsymbol{z}|\boldsymbol{x})$ using a variational distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ with ϕ being a collection of unknown parameters depending on the observed data $\boldsymbol{x}$. The problem then is transformed into maximizing the evidence lower bound (ELBO) $\mathcal{L}_{\text{ELBO}}\{\phi(\boldsymbol{x})\}$ (Blei et al., 2017):

$$\log p_\theta(\boldsymbol{x}) \geq \mathcal{L}_{\text{ELBO}}\{\phi(\boldsymbol{x})\},$$
$$\text{where } \mathcal{L}_{\text{ELBO}}\{\phi(\boldsymbol{x})\} \triangleq \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right].$$

A tighter bound derived from importance weighting, namely importance-weighted autoencoder (IWAE) (Burda et al., 2015), is then proposed with a strictly tighter log-likelihood lower bound than ELBO. To calculate an importance-weighted estimate of the log-likelihood, T independent samples are drawn from the posterior $\{\boldsymbol{z}_t\}_{t=1}^T \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})$, and a lower bound is then calculated as the log of the average of the ratio of the joint distribution and posterior for each sample:

$$\mathcal{L}_{\text{IWAE}}\{\phi(\boldsymbol{x})\} \tag{1}$$
$$\triangleq \mathbb{E}_{\{\boldsymbol{z}_t\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})\}_{t=1}^T}\left[\log \frac{1}{T}\sum_{t=1}^T \frac{p_\theta\left(\boldsymbol{x}|\boldsymbol{z}_t\right)p\left(\boldsymbol{z}_t\right)}{q_\phi\left(\boldsymbol{z}_t|\boldsymbol{x}\right)}\right].$$

While tighter bound may not always be the best option (Rainforth et al., 2018), previous works suggest that multiple samples from the posterior help IWAE to be well-adapted to multimodal distributions and to approximate complex posteriors (Burda et al., 2015; Morningstar et al., 2021). In this paper, we adopt IWAE as the objective function for its overall better properties.

2.2 Spline Approximation

Spline approximation provides a method for approximating complex curves with a modest set of parameters, thereby achieving computational efficiency and having been widely used in machine learning and statistical learning areas, including generalized additive models (Hastie, 2017), functional data analysis (Wang et al., 2016), longitudinal data analysis (Anderson and Jones, 1995), neural networks (Balestriero and Baraniuk, 2018; Fakhoury et al., 2022), and point process intensity estimation (Loaiza-Ganem and Cunningham, 2019).

Let $\boldsymbol{\upsilon}$ be a partition of the interval $\mathcal{T} = [\upsilon_0, \upsilon_{H+1}]$ with H interior knots, where $\boldsymbol{\upsilon} = \{\upsilon_0 < \upsilon_1 < \cdots < \upsilon_H < \upsilon_{H+1}\}$. Any spline function $s(z)$ within the spline space $\mathcal{U}$ of order $\varrho + 1$ satisfies that: 1) the function $s(z)$ is a polynomial function with ϱ-degree (or less) on intervals $[\upsilon_h, \upsilon_{h+1})$, $h = 0, \ldots, H$ and $[\upsilon_H, \upsilon_{H+1}]$; 2) it has $\varrho - 1$ continuous derivatives over the entire region $\mathcal{T}$. Consider $\{B_{1,\mathcal{T}}(z), \ldots, B_{K,\mathcal{T}}(z)\}^\top$ as a vector of the spline basis functions with degree ϱ and partition $\boldsymbol{\upsilon}$, where $K = H + \varrho + 1$. For the sake of notation simplicity, for the rest of the paper, we define the normalized spline basis by $b_{k,\mathcal{T}}(z) \triangleq B_{k,\mathcal{T}}(z)/a_{k,\mathcal{T}}$, where $a_{k,\mathcal{T}} = \int_{\mathcal{T}} B_{k,\mathcal{T}}(z)dz$. It implies that $\int_{\mathcal{T}} b_{k,\mathcal{T}}(z)dz = 1$. Let $\boldsymbol{b}_{\mathcal{T}}(z) = \{b_{1,\mathcal{T}}(z), \ldots, b_{K,\mathcal{T}}(z)\}^\top$. All the spline basis functions $b_{k,\mathcal{T}}(z)$ are nonnegative; therefore, they are all valid probability density functions. For any polynomial spline $s(z)$, it can be uniquely represented via a linear combination of spline basis functions, that is, $s(z) = \sum_{k=1}^K \gamma_k b_{k,\mathcal{T}}(z)$.

Define $\mathcal{H}^{(\varrho)}(\mathcal{T})$ as the space of functions ψ on $\mathcal{T}$ whose ν-th derivative exists and satisfies a Lipschitz condition of order $\delta : \left|\psi^{(\nu)}(z) - \psi^{(\nu)}(z')\right| \leq C_v \left|z - z'\right|^\delta$, for $z, z' \in \mathcal{T}$ and $\varrho = \nu + \delta$. The following Lemma 2.1 can quantify the approximation power of polynomial splines

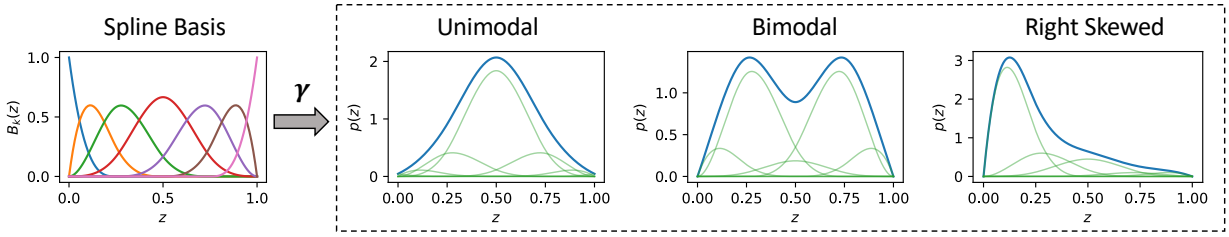


Figure 1: Illustration Of The Density Functions Based On Different Linear Combinations Of Spline Basis Functions

for functions within $\mathcal{H}^{(\varrho)}(\mathcal{T})$, suggesting smooth functions can be well approximated when the knot number increases to infinity.

Lemma 2.1 (Schumaker (2007)). *For any function $\psi \in \mathcal{H}^{(\varrho)}(\mathcal{T})$, there exists a spline $\psi^* \in \mathcal{U}$, such that $\sup_{z\in\mathcal{T}} |\psi^*(z) - \psi(z)| \leq CH^{-(\varrho+1)}$ for some positive constant C.*

Remark 2.2. According to Lemma 2.1, for density function $p(z) \in \mathcal{H}^{(\varrho)}(\mathcal{T}^o)$, where $\mathcal{T}^o \subseteq \mathbb{R}$ could be either finite or infinite support, there exists $\widetilde{p}(z) \in \mathcal{H}^{(\varrho)}(\mathcal{T})$ such that is $\widetilde{p}(z)$ a valid density function and $\sup_{z\in\mathcal{T}^o} |\widetilde{p}(z) - p(z)| \leq CH^{-(\varrho+1)}$.

3 NONPARAMETRIC POSTERIOR APPROXIMATION WITH SPLINE

Remark 2.2 emphasizes the capability of spline functions to approximate complex distributions provided sufficient interior knots. To enhance the approximation of posteriors with arbitrary shapes, we introduce Spline Automatic Differentiation Variational Inference (S-ADVI). This nonparametric approach aims to represent posteriors as spline functions, allowing for more flexible and accurate modeling.

3.1 Spline Automatic Differentiation Variational Inference (S-ADVI)

Following the framework of nonparametric Bayesian inference, we assume the true posterior $p(z|\boldsymbol{x})$ is within an infinite dimensional space such that $p(z|\boldsymbol{x}) \in \mathcal{H}^{(\varrho)}(\mathcal{T}^o)$. Therefore, according to Lemma 2.1 and Remark 2.2, the posterior distribution of a latent variable z, $p(z|\boldsymbol{x})$ can be well approximated by a spline function: $q_\phi(z|\boldsymbol{x}) = \sum_{k=1}^K \gamma_k(\boldsymbol{x}) b_{k,\mathcal{T}}(z) = \boldsymbol{b}(z)^\top \boldsymbol{\gamma}(\boldsymbol{x})$. See Figure 1 to illustrate the density functions based on the linear combinations of spline basis functions. By the definition of normalized spline basis in Section 2.2, $\int_{\mathcal{T}} b_{k,\mathcal{T}}(z)dz = 1$, for $k = 1, \ldots, K$. Therefore, to ensure that $\int_{\mathcal{T}} q_\phi(z|\boldsymbol{x})dz = 1$ and $q_\phi(z|\boldsymbol{x}) > 0$ for $z \in \mathcal{T}$, the spline coefficients must satisfy $\gamma_k(\boldsymbol{x}) \geq 0$ and $\sum_{k=1}^K \gamma_k(\boldsymbol{x}) = 1$. For notation simplicity, we denote $b_k(z) = b_{k,[0,1]}(z)$ for the rest of paper. We consider the mean-field assumption and assume the latent variables to be independent of each other. For j-th latent variable, we use $\mu_j(\boldsymbol{x})$ and $\sigma_j(\boldsymbol{x})$ for location-scale transformations for latent variable z_j, such that $z_j = \mu_j(\boldsymbol{x}) + \sigma_j(\boldsymbol{x})\epsilon_j$, and $\epsilon_j \in [0, 1]$ is a random variable. The location-scale transformation allows for adaptive supports of posteriors on $[\mu_j(\boldsymbol{x}), \mu_j(\boldsymbol{x}) + \sigma_j(\boldsymbol{x})]$, where $\mu_j(\boldsymbol{x})$ and $\sigma_j(\boldsymbol{x})$ are unknown parameters to be estimated. The proposed posterior is determined by unknown parameters $\mu_j(\boldsymbol{x})$, $\sigma_j(\boldsymbol{x})$, and the spline coefficients $\{\gamma_{jk}(\boldsymbol{x}), k = 1, \ldots, K\}$, which capture the location, scale, and shape of the distribution. With pre-specified spline basis functions in the S-ADVI, we can use spline coefficients to represent the shape of approximated posteriors. Section 6.2 demonstrates using spline coefficients to investigate the relationship between the shape of posteriors and input features, enhancing the model interpretation. Let J be the total number of latent variables. Collectively, for the vector of latent variables $\boldsymbol{z} = \{z_1, \ldots, z_J\}$, the posterior $p(\boldsymbol{z}|\boldsymbol{x})$ can be approximated by

$$
\begin{aligned}
q_\phi(\boldsymbol{z}|\boldsymbol{x}) &= \prod_{j=1}^J \sigma_j^{-1}(\boldsymbol{x}) \cdot q_\phi(\boldsymbol{\epsilon}|\boldsymbol{x}) \qquad (2)\\
&= \prod_{j=1}^J \frac{1}{\sigma_j(\boldsymbol{x})} \cdot \sum_{k=1}^K \gamma_{jk}(\boldsymbol{x}) b_k\left(\frac{z_j - \mu_j(\boldsymbol{x})}{\sigma_j(\boldsymbol{x})}\right),
\end{aligned}
$$

where $\boldsymbol{\epsilon} = \{\epsilon_1, \ldots, \epsilon_J\}$ is the vector of latent variables before the location-scale transformation. In the S-ADVI, the parameters of approximation family are $\phi = \{\mu_j(\boldsymbol{x}), \sigma_j(\boldsymbol{x}), \gamma_{jk}(\boldsymbol{x}), k = 1, \ldots, K, j = 1, \ldots, J\}$. The objective of the proposed S-ADVI is to maximize the IWAE defined in (1), where the term $q_\phi(\boldsymbol{z}_t|\boldsymbol{x})$ is given in (2). The spline degree ϱ, and the number and values of interior knots are hyperparameters to be specified. In the numerical studies, we choose the cubic spline ($\varrho = 3$) with equal-space knots, which is commonly used in nonparametric model estimation (Hastie, 2017; Yu et al., 2020). The influence of the number of interior knots is evaluated in the experiments in Section 6.2.

Remark 3.1. It is possible to get correlated $\boldsymbol{z}$ by taking a linear transformation $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\Sigma}\boldsymbol{\epsilon}$, where $\boldsymbol{\Sigma}$ is an unknown covariance matrix and the random variables in $\boldsymbol{\epsilon}$ are independent of each other. For complex, nonlinear

dependency between components, multivariate posterior estimation is viable through multivariate spline approximation to capture relationships between latent variables, which is considered as a future study.

3.2 Model Estimation

The spline posterior approximation can be regarded as a mixture of density functions based on spline bases. The stratified ELBO (SELBO)/IWAE (SIWAE) (Morningstar et al., 2021; Roeder et al., 2017) are common methods to optimize ELBO/IWAE with mixture densities. However, in S-ADVI, directly applying the stratified techniques is challenging, which involves the summation of products of all the combinations of spline coefficients and latent variables. When the number of latent variables is large, the stratified methods can be computationally expensive.

We tackle the above-mentioned challenge via the concrete distribution (Maddison et al., 2017), which can be used as an approximation to the categorical distribution. The concrete distribution has two parameters, $\boldsymbol{\alpha} \in (\mathbb{R}^+)^K$ and Λ, where $\sum_k \alpha_k = 1$. When $\Lambda \to 0$, the concrete distribution approaches the categorical distribution with the event probability vector being $\boldsymbol{\alpha}$. However, if Λ is fixed low, the concrete approximation cannot explore different combinations of spline bases, leading to poor model estimation. We use an annealing approach (Abid et al., 2019), where we start model training with a high $\Lambda = \Lambda_0$, and gradually reduce Λ after each epoch. In this paper, we use $\Lambda(c) = \Lambda_1 + (\Lambda_0 - \Lambda_1)e^{-c/\eta}$ to smoothly reduce the temperature, where Λ_1 is the final temperature, c denotes the current epoch number, and η controls the speed of decay. As shown in the experiments (Section S.3.4 in supplementary material), S-ADVI is not sensitive to the choice of annealing functions.

To this end, we summarize the Stochastic Backpropagation (Rezende et al., 2014) for estimating S-ADVI.

Generating random samples from mixture models. One key component of the proposed S-ADVI method is to generate random samples from a mixture model with distribution $q_\phi(z_j|\boldsymbol{x}) = \sum_{k=1}^K \gamma_{jk}(\boldsymbol{x})\widetilde{b}_{k,\mathcal{T}}(z_j;\boldsymbol{x})$, where $\widetilde{b}_{k,\mathcal{T}}(z_j;\boldsymbol{x}) = \sigma_j(\boldsymbol{x})^{-1} b_k\left\{\sigma_j^{-1}(\boldsymbol{x})[z_j - \mu_j(\boldsymbol{x})]\right\}$. A hierarchical approach to generate random samples from mixture models involves two steps: generating random samples for distributions $\widetilde{b}_{k,\mathcal{T}}(z_j;\boldsymbol{x})$ and randomly selecting one sample with probability $\gamma_{jk}(\boldsymbol{x})$ for each z_j. However, it is not straightforward to sample from $\widetilde{b}_{k,\mathcal{T}}(z_j;\boldsymbol{x})$ and apply the reparameterization trick to the categorical distribution. Utilizing the pre-specified spline bases, with concrete approximation, at each iteration, we consider the following procedures:

1. Use the Metropolis-Hastings algorithm to generate a sequence of random samples from the distribution $b_k(\epsilon_j)$ and then randomly pick w_{jk} from generated samples.
2. Generate random sample $\boldsymbol{u}_j$ from a concrete distribution with $\Lambda = \Lambda(c)$ and $\alpha_{jk} = \gamma_{jk}(\boldsymbol{x})$.
3. Define $\epsilon_j = \sum_{k=1}^K u_{jk} w_{jk}$. The property of concrete distribution guarantees that when $\Lambda(c) \to \Lambda_1$ as c increases, the procedure well approximates the discrete hierarchical sampling process.

Backpropagation with reparameterization trick. We aim to differentiate the objective function w.r.t. the parameters ϕ via a Monte Carlo approximation. The Monte Carlo approximation is based on the random samples generated from the mixture of spline basis functions. To obtain the differentiation, we consider the following reparameterization trick for our objective function $\mathcal{L}_{\text{IWAE}}\{\phi(\boldsymbol{x})\}$:

$$\mathbb{E}_{\{\boldsymbol{\epsilon}_t\}_{t=1}^T}\left[\log \frac{1}{T}\sum_{t=1}^T \frac{p_\theta\left\{\boldsymbol{x}, \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{\epsilon}_t\right\}}{\prod_{j=1}^J \left\{\sum_{k=1}^K \gamma_{jk}(\boldsymbol{x}) b_k\left(\epsilon_{jt}\right)\right\}}\right] + \sum_{j=1}^J \log \sigma_j(\boldsymbol{x}), \tag{3}$$

where $\boldsymbol{\epsilon}_t = \{\epsilon_{1t}, \ldots, \epsilon_{Jt}\}^\top$ are variables generated from $\sum_{k=1}^K \gamma_{jk}(\boldsymbol{x}) b_k(\epsilon_{jt})$, $\boldsymbol{\mu}(\boldsymbol{x}) = \{\mu_j(\boldsymbol{x}), j = 1, \ldots, J\}$ and $\boldsymbol{\sigma}(\boldsymbol{x}) = \{\sigma_j(\boldsymbol{x}), j = 1, \ldots, J\}$ are vectors of location and scale parameters. The $\log \sigma_j(\boldsymbol{x}), j = 1, \ldots, J$ terms in (3) prevents the model from degenerating into a deterministic model. Derivative of (3) can be found in Section S.2.2 of the supplementary material.

Penalized spline. In nonparametric smoothing, penalized spline captures intricate data patterns with regularization (roughness penalty) to prevent overfitting and manage the complexity of the fitted function (Wood, 2003). Here, we consider a roughness penalty for a spline function $s(\cdot)$, defined as $\mathcal{E}(s) = \int_{\mathcal{T}}\{s''(t)\}^2 dt$, to control the complexity of the fitted curve and avoid overfitting. According to properties of spline polynomials, $\mathcal{E}(s) = \boldsymbol{\gamma}^\top \mathbf{P} \boldsymbol{\gamma}$, where the matrix $\mathbf{P}$ is a K by K positive definite matrix, see Section S.2.1 of the supplementary material for the detailed definition. Then, the objective function becomes $\mathcal{L}^P_{\text{IWAE}}\{\phi(\boldsymbol{x})\} = \mathcal{L}_{\text{IWAE}}\{\phi(\boldsymbol{x})\} + \lambda \boldsymbol{\gamma}^\top \mathbf{P} \boldsymbol{\gamma}$.

4 PROPERTIES OF S-ADVI

While existing works have demonstrated that the spline approximation has an upper bound on the approximation error for any function within the functional space $\mathcal{H}^{(\varrho)}(\mathcal{T}^o)$ (Lemma 2.1), it is worth examining the approximation power of the proposed S-ADVI based on

spline approximation. In this section, we start with the bound of IWAE (Theorem 4.1), which implies the bound of KL divergence of the spline density approximation from the true posterior (Theorem 4.2). Then, Theorem 4.3 quantifies the posterior approximation error between the S-ADVI estimator and true posterior.

We first state the necessary assumptions to facilitate our theoretical studies.

(A1) The prior $p(\boldsymbol{z})$ and the likelihood function $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ are bounded over the support regions.

(A2) The true posteriors can be fully factorized, that is, $p(\boldsymbol{z}|\boldsymbol{x}) = \prod_{j=1}^{J} p(z_j|\boldsymbol{x})$. For $j = 1, \ldots, J$, the posterior $p(z_j|\boldsymbol{x}) \in \mathcal{H}^{(\varrho)}(\mathcal{T}^o)$. There exists some region $\mathcal{T}^* \subset \mathcal{T}^o$ such that $\int_{\mathcal{T}^o - \mathcal{T}^*} p(z_j|\boldsymbol{x})dz_j < \epsilon$ for some $\epsilon > 0$. In addition, the posterior $p(z_j|\boldsymbol{x})$ are bounded by some constant over $\mathcal{T}^*$.

(A3) There exist constants C_1 and C_2 such that $C_1 H^{-1} \leq \upsilon_h - \upsilon_{h-1} \leq C_2 H^{-1}$ for $1 \leq h \leq H$.

Remark 4.1. Assumption (A1) is a mild assumption on the prior $p(\boldsymbol{z})$ and the likelihood function, which can be easily satisfied. Assumptions (A2) – (A3) are typical assumptions under the framework of spline approximation (Wang and Yang, 2009; Yu et al., 2020). Assumption (A3) assumes that the true posteriors can be fully factorized. For more general cases, the true posterior can be factorized into $\prod_{m=1}^{M} p(\boldsymbol{z}_{s_m}|\boldsymbol{x})$, where s_m is an index set of the latent variables and $\cup_{m=1}^{M} s_m = \{1, \ldots, J\}$. Applying the functional ANOVA results in (Stone, 1994), we can derive the L_2 approximate rate is $H^{-(\varrho^*+1)}$, where the ϱ^* is a suitably lower bound to the smoothness of the components $p(\boldsymbol{z}_{s_m}|\boldsymbol{x}), m = 1, \ldots, M$. The above approximation result is a general form of multivariate posterior approximation, allowing complex interactions between latent variables.

Lemma 4.1 shows that the proposed S-ADVI allows us to quantify the lower bound of IWAE.

Lemma 4.1. *Under Assumptions (A1) – (A3), the optimal IWAE is bounded by* $\log p_\theta(\boldsymbol{x}) - CJH^{-(\varrho+1)} - J\epsilon$, *where C is some positive constant.*

Theorem 4.2 quantifies the variational approximation error with respect to the class defined in (2). See Section S.1.1 in the supplementary material for the detailed proof.

Theorem 4.2. *Under Assumptions (A1) – (A3), the difference between the true posterior and the spline estimator is bounded by the order of $H^{-(\varrho+1)}$, that is, there exists a constant C, such that $D_{KL}\{q_{\phi(\boldsymbol{x})}(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})\} \leq CJ(H^{-(\varrho+1)} + \epsilon)$.*

We consider the posterior approximation based on the observed data $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$. When the observed data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ are close enough, the corresponding posteriors $p(\boldsymbol{z}|\boldsymbol{x}_i)$ and $p(\boldsymbol{z}|\boldsymbol{x}_{i'})$ are close. For any given posterior $p(\boldsymbol{z}|\boldsymbol{x})$ satisfying Assumption (A2), there exists a density function based on spline approximation $\prod_{j=1}^{J}\sum_{k=1}^{K}\gamma^*_{jk}(\boldsymbol{x})b_{k,\mathcal{T}_j^*}(z_j)$ with $\mathcal{T}_j^* = [\mu_j^*(\boldsymbol{x}), \mu_j^*(\boldsymbol{x}) + \sigma_j^*(\boldsymbol{x})]$ close to $p(\boldsymbol{z}|\boldsymbol{x})$ with differences bounded by $JH^{-(\varrho+1)}$. Under some mild assumptions, $\gamma^*_{jk}(\boldsymbol{x})$, $k = 1, \ldots, K$, $\mu_j^*(\boldsymbol{x})$, and $\sigma_j^*(\boldsymbol{x})$ can be well approximated by nonparametric regression, such as the deep neural network. Specifically, the objective function can be formulated as $\sum_{i=1}^{n}\mathcal{L}_{\text{IWAE}}\{\phi(\boldsymbol{x}_i)\}$ and $\widehat{\phi}(\boldsymbol{x}) = \arg\max_\phi \sum_{i=1}^{n}\mathcal{L}_{\text{IWAE}}\{\phi(\boldsymbol{x}_i)\}$, where $\phi(\boldsymbol{x}_i)$ is the collection of parameters of the S-ADVI estimators.

Theorem 4.3 quantifies posterior approximation error generated from the nonparametric smoothing and theoretical differences between the S-ADVI estimator and true posterior. See Section S.1.2 in the supplementary material for the detailed proof.

Theorem 4.3. *Under Assumptions (A1) – (A3), the average KL divergence of the spline estimator from the true posterior satisfies has*

$$\lim_{n\to\infty} Pr\Big[n^{-1}\textstyle\sum_{i=1}^{n} D_{KL}\{q_{\widehat{\phi}(\boldsymbol{x}_i)}(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x}_i)\} \leq CJ(H^{-2(\varrho+1)} + \epsilon^2 + H^2\Delta^2)\Big] = 1,$$

where C is a positive constant and Δ is the L_2 estimation error of nonparametric regression for $\mu_j(\boldsymbol{x})$, $\sigma_j(\boldsymbol{x})$, and $\gamma_{jk}(\boldsymbol{x}), k = 1, \ldots, K, j = 1, \ldots, J$.

Remark 4.4. Theorem 4.2 suggests increasing the number of interior knots H can reduce the S-ADVI approximation errors. However, when applied to real data analysis, according to the results in Theorem 4.3, choosing the optimal number of interior knots balances approximation bias and estimation variance. In addition, the roughness penalty in penalized spline is used to avoid overfitting.

Remark 4.5. (*Example of the convergence rate* Δ) We assume that all the components are compositions of several functions. Suppose that $\phi(\boldsymbol{x}) = g_q \circ g_{q-1} \circ \cdots \circ g_1 \circ g_0$, where $g_i : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}}$. Denote by $g_{ij}, j = 1, \ldots, d_{i+1}$ the components of g_i. Let t_i be the maximal number of variables on which each of the g_{ij} depends, and each g_{ij} is a t_i-variate function. Then, the convergence rate is $\Delta = \max_{i=0,\ldots,q} n^{-(2\beta_i)/(2\beta_i+t_i)}$, where $\beta_i, i = 0, \ldots, q$ are degrees of Hölder smoothness conditions of functions g_{ij} (Schmidt-Hieber, 2020).

5 RELATED WORKS

Variational inference has traditionally relied on parametric approximations, but efforts to enhance flexibility have led to various approaches. Gaussian mixture approximation offers a flexible posterior approximation but is susceptible to issues like posterior collapse

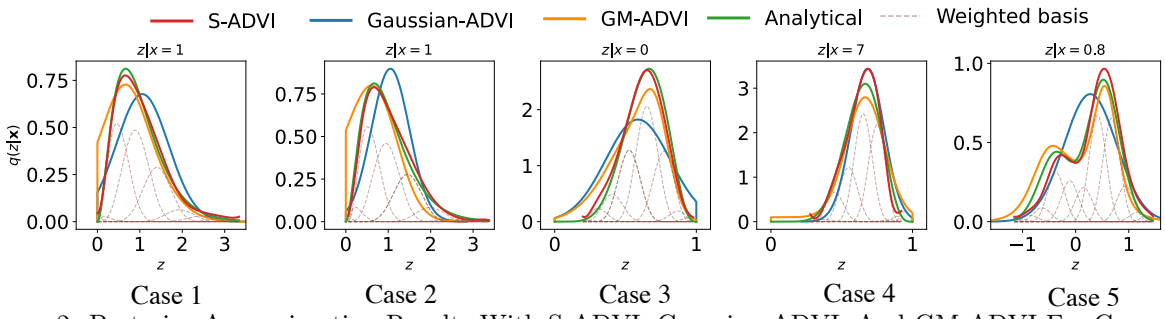


Figure 2: Posterior Approximation Results With S-ADVI, Gaussian-ADVI, And GM-ADVI For Cases 1–5

(Gershman et al., 2012). Normalizing flow, employing invertible functions to model posterior distributions, provides an alternative flexible approach (Rezende and Mohamed, 2015; Rezende et al., 2020; Wu et al., 2020). Neural Spline Flows utilizes monotonic and element-wise rational-quadratic spline as building blocks of normalizing flow (Durkan et al., 2019). Boosting variational inference, a mixture-based approximation, offers flexibility but poses implementation and interpretation challenges (Kobyzev et al., 2020; Locatello et al., 2018). On the other hand, the estimation of the density of the non-parametric kernel resembles a Gaussian mixture with numerous components (Gershman et al., 2012).

Implicit processes represent another avenue for facilitating flexible inferences utilized across Bayesian neural networks, neural samplers, and data generation frameworks. The enhancement of priors and posteriors through approximate inference techniques is well-documented (Ma et al., 2019; Molchanov et al., 2019; Ortega et al., 2022; Shi et al., 2017; Takahashi et al., 2019). A notable method for training implicit models involves the nonparametric approximation of log density, known as the score estimator, which has been explored in recent studies (Li and Turner, 2017; Shi et al., 2018; Sriperumbudur et al., 2017). Furthermore, a comprehensive examination and convergence analysis of existing score estimators have been presented, offering a unified perspective on this methodology (Zhou et al., 2020). Despite these advancements, estimating implicit posteriors, especially in models characterized by high-dimensional latent variables, remains a significant challenge (Rodríguez-Santana et al., 2022).

Research on variational approximations has explored theoretical guarantees, convergence, optimization techniques, and model-specific analyses. Frequentist consistency has been considered (Wang and Blei, 2019; Zhang and Gao, 2020), as well as overparameterized Bayesian Neural Networks (Huix et al., 2022). Notably, existing theoretical studies have primarily focused on parametric distribution families.

6 RESULTS

In this section, we demonstrate the proposed method with experiments on both simulated and real datasets. All experiments are based on PyTorch 2.0 (Paszke et al., 2019) running on a Nvidia A100 80G GPU [1].

6.1 Posterior Approximation

We consider the following five simulation cases to demonstrate the proposed method (S-ADVI) in approximating the posterior distribution of $z|x$:

1. $z \sim \text{Gamma}(2,2)$, $x|z \sim \text{Exponential}(z)$;
2. $z \sim \text{Gamma}(2,2)$, $x|z \sim \text{Poisson}(z)$;
3. $z \sim \text{Beta}(7,3)$, $x|z \sim \text{Bernoulli}(z)$;
4. $z \sim \text{Beta}(2,2)$, $x|z \sim \text{Binomial}(10,z)$;
5. $z \sim 0.5N(-0.5,0.1) + 0.5N(0.5,0.1)$, $x|z \sim N(z,1)$.

We generate 1024 samples for all cases, training models in batches of 32 across 40 epochs over 20 runs. A two-layer multilayer perceptron (MLP) with 20 hidden units per layer is used to estimate unknown parameters. For S-ADVI, we set the interior knots (H) to 6 for Cases 1-4, and 9 for the final case due to its complex multimodal structure. Performance is evaluated using root integrated squared error (RISE), defined as $[\int [q(z|x) - p(z|x)]^2 dz]^{1/2}$, comparing our method against Gaussian-ADVI (approximating $p(z|x)$ with a Gaussian distribution) and Gaussian Mixture ADVI (GM-ADVI, Morningstar et al. (2021)) based on stratified sampling. For likelihoods requiring bounded support of $z|x$, truncated distributions of $z|x$ are considered for Gaussian-ADVI and GM-ADVI.

The results presented in Table 1 underscore the advantage of S-ADVI over Gaussian-ADVI and GM-ADVI, and Table S.1 in the supplementary materials shows additional comparison results with methods based on normalizing flows. Visual representations in Figure 2 and Figure S.1 (supplementary material) depict the approximated posteriors compared to the true posterior

[1] Example codes are available at: `https://github.com/TianshuFeng/SADVI_AISTATS2024`

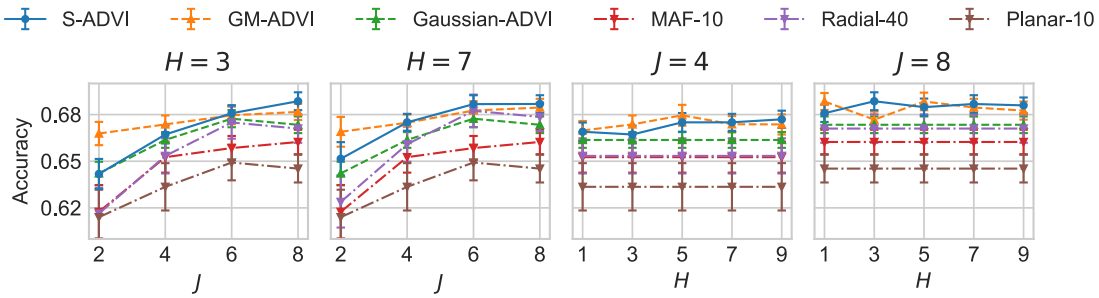


Figure 3: Performance Comparison For Single Column Classification For The FMNIST Dataset With A Range Of Latent Variables J And Interior Knots H (Error Bars Denote The Standard Error)

Table 1: Mean And Standard Deviation (in Bracket) Of Root Integrated Squared Error In Posterior Approximation

Method	Case 1	Case 2	Case 3	Case 4	Case 5
Gaussian-ADVI	0.408 (0.274)	0.239 (0.044)	0.630 (0.239)	0.631 (0.386)	0.243 (0.014)
GM-ADVI	0.353 (0.152)	0.211 (0.026)	0.403 (0.115)	0.395 (0.107)	0.137 (0.020)
S-ADVI	**0.086** (0.046)	**0.054** (0.020)	**0.211** (0.070)	**0.310** (0.080)	**0.097** (0.025)

functions, showcasing S-ADVI's ability to approximate the true posterior distribution. Across Cases 1 to 5, S-ADVI consistently outperforms other methods by leveraging spline functions for posterior approximation. Specifically, for Cases 1 through 3 and Case 5, where the true posterior distribution exhibits significant skewness or complex multimodal nature, Gaussian-ADVI effectively estimates the posterior locations but struggles to capture their shapes. In Cases 1 through 3, characterized by strong skewness and bounded supports in the true posterior distribution, GM-ADVI exhibits subpar performance. While GM-ADVI captures the skewness of the posteriors, it fails to estimate the boundaries, showing a high estimation error when z is close to the extreme points. In contrast, our proposed methodology excels, particularly when dealing with distributions that are skewed and with bounded latent variable supports. Additionally, Figure S.1 shows that normalizing flows effectively capture the posterior distribution's shape. In complex scenarios like the multimodal distributions of Case 5, normalizing flows outperform the GM-ADVI and Gaussian-ADVI through invertible density transformations. However, normalizing flows generates unexpected irregularities, such as wiggles in Cases 1 and 2. One possible reason is that, in contrast to the complex invertible density transformations of normalizing flows, S-ADVI provides a more effective way to approximate posterior distributions.

6.2 Real Data Applications

Single Column Classification. We conduct classification tasks on the Fashion-MNIST (FMNIST) (Xiao et al., 2017) datasets to evaluate the performance of S-ADVI under different parameters. The model used

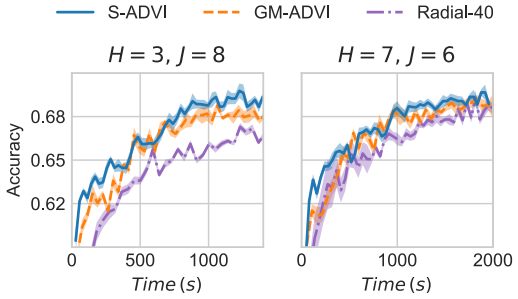


Figure 4: Computational Budget Comparison For Single Column Classification For The FMNIST Dataset With A Range Of Latent Variables J And Interior Knots H (Shadows Represent The Standard Error)

for classification is Variational Information Bottleneck (VIB) (Alemi et al., 2017), which shares a similar structure as VAE with decoder replaced by a classifier. We use a simple multinomial logistic regression as the classifier and the isotropic Gaussian distribution prior to encouraging the encoder to capture the underlying information present in the images. To showcase the flexible spline approximation, we limit the input samples only to include the center column of training images. An illustration example of the input sample can be found in Figure S.4 in the supplementary material.

We assess S-ADVI's performance with $H = 1, 3, 5, 7, 9$, $J = 2, 4, 6, 8$ and $T = 10$ in IWAE, setting $v_0 = 0$ and $v_{H+1} = 1$ with equal spacing for v_h where $h = 1, \ldots, H$. For comparison, we consider VIB with GM-ADVI and Gaussian-ADVI, matching the number of Gaussian components in GM-ADVI to the spline bases in S-ADVI ($H + 4$). Three normalizing flow-based methods, Planar and Radial (10 and 40 flows), and masked autoregressive flow with 10 flows (MAF-10)

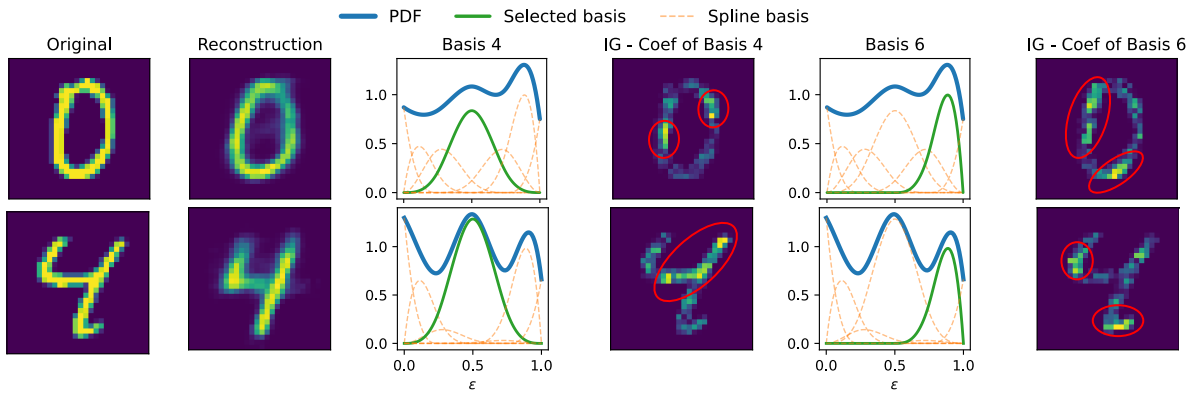


Figure 5: Analysis Of The Approximated Shape Of Posterior From S-VAE, Where Brighter Pixels Correspond To Higher Absolute Attributes, And Regions Containing Highly Attributed Pixels Are Highlighted In Red Circles

are also compared. Prediction scores are calculated by averaging model outputs from 100 samples drawn from $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. Experiments are repeated 10 times, and we reporting the mean and standard error of accuracy on the test set.

Figure 3 shows the performance comparison on the FMNIST dataset for the top six methods across varying J and H for S-ADVI ($H+4$ basis for GM-ADVI). Results indicate S-ADVI's performance generally improves with increased latent variables and interior knots, aligning with theoretical outcomes. S-ADVI benefits from additional interior knots when the number of latent variables is relatively small. With sufficient latent variables, extra interior knots is less impactful and can lead to overfitting. GM-ADVI's flexible Gaussian components provide an edge over S-ADVI with limited J and H. However, adding interior knots or latent variables can help S-ADVI to match and outperform GM-ADVI, which may result from the natural regularization of the latent variables with bounded support of spline approximation. Both S-ADVI and GM-ADVI outperform the normalizing flow methods, especially with limited latent variables.

In addition, we compare the computational budget of the top 3 performing methods: S-ADVI, GM-ADVI, and Radial-40. Results are shown in Figure 4. In general, S-ADVI converges faster than GM-ADVI and Radial-40, especially when the numbers of knots and latent variables are limited. One of the potential reasons could be the fewer parameters required by S-ADVI. When the number of latent variables increases, three methods achieve comparable convergence speeds. Annealing is important in the proposed method to ensure most of the combinations of coefficients are explored. Section S.3.4 of the supplementary file evaluates the influences of annealing functions on model performance. The performance of our proposed method is robust to the choice of annealing function.

Imaging Reconstructions. Our previous experiments suggest that the proposed S-ADVI methods can match and outperform other VI-based methods in terms of classification task, even when the data does not contain sufficient information. In this experiment, we further compare the S-ADVI and GM-ADVI methods in terms of reconstruction using MNIST (LeCun et al., 1998), FMNIST, and CIFAR-10 (Krizhevsky et al., 2009) datasets. Section S.3.5 in the supplementary material presents the implementation details and the numerical results. We find that when the number of latent variables is limited, GM-ADVI outperforms S-ADVI. When J increases, the performance of both models improves, and S-ADVI outperforms GM-ADVI. The number of bases H also influences model performance when H increases from 1 to 3, but further increasing H does not improve model performance much and can lead to overfitting. It is also interesting to note that the advantage of S-ADVI increases for more complicated datasets (e.g., MNIST vs CIFAR-10) with higher J and H.

Interpreting Distributions of Latent Variables. We evaluate the relationship between spline approximations' shape and input features by training a variational autoencoder with S-ADVI (S-VAE) on the MNIST benchmark dataset to reconstruct input images. Implementation details and overall results are provided in Section S.3.6 (supplementary material). We assess the shape of posterior distribution pre location-scale transformation, focusing on the density functions of $\epsilon_j|\boldsymbol{x}$ defined in (2), with support $q_\phi(\epsilon_j|\boldsymbol{x})$ in $[0,1]$ for all $j = 1,\ldots,J$, so that the shapes are comparable across samples.

Using Integrated Gradients (IG) (Sundararajan et al., 2017), we explore the link between input samples and approximated $q_\phi(\epsilon_j|\boldsymbol{x})$ shapes. For a sample $\boldsymbol{x}$ and its density function $\sum_{k=1}^{K}\gamma_{jk}(\boldsymbol{x})b_k(\epsilon_j)$, IG attributes $\gamma_{jk}(\boldsymbol{x})$ values to input sample features, here MNIST

pixels. We measure feature relative importance to $\gamma_{jk}(\boldsymbol{x})$ using IG's absolute attribute values. We set the baseline to a zero vector, representing a blank image.

Experiment results in Figure 5 show original samples and S-VAE reconstructions (Columns 1 and 2). We selected bases 4 and 6 of $q_\phi(\epsilon_4|\boldsymbol{x})$ with distinct modes for different digits (Figure S.9). Columns 3 and 5 illustrate selected and other weighted spline bases, while Columns 4 and 6 display IG results, with brighter pixels indicating higher attribute values. We notice that bases with higher coefficients represent larger image regions and capture key image characteristics. For instance, IG reveals that in the third image, basis 4 corresponds to digit 4's upper right structure, and basis 6 to the top left part of digit 0.

7 DISCUSSION

In this paper, we introduce a nonparametric ADVI framework that uses spline approximations to approximate posterior distributions and achieves a balance between flexibility, parsimony, and interpretability. We establish S-ADVI's posterior consistency in approximating complex distributions through the asymptotic properties of IWAE. Compared with classic ADVI methods, experiment results suggest S-ADVI's superior capacity to approximate distributions with bounded support and multimodality.

Despite its strengths, the proposed S-ADVI can be further improved. First, our methodology uses the mean-field approximation for model simplicity and computational efficiency, but it does not account for the underlying dependencies among latent variables. Second, we can further optimize the locations of interior knots, which are pre-specified in the current framework, based on dataset characteristics (Spiriti et al., 2013). In addition, incorporating techniques such as fused lasso could help in subgroup analysis on spline coefficients (Tibshirani et al., 2005). For the posterior inference problems in VI, latent variables can be divided into two categories: local latent variables for individual observations and global latent variables (Wang and Blei, 2019). This paper focuses primarily on the posterior inference of local latent variables, as discussed in the theoretical analysis and numerical evaluations. Investigating the theoretical properties of global latent variables represents promising future research. Finally, our spline-based posterior approximation approach opens up possibilities for modeling spatial-temporal data within the ADVI framework, which aligns with the current research trends towards more accurate prior approximations in variational autoencoders (Pang et al., 2020).

Acknowledgements

We would like to thank the reviewers for valuable feedback and suggestions.

References

Abid, A., Balin, M. F., and Zou, J. (2019), "Concrete autoencoders for differentiable feature selection and reconstruction," *arXiv preprint arXiv:1901.09346*.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. (2017), "Deep variational information bottleneck," in *International Conference on Learning Representations*.

Anderson, S. J. and Jones, R. H. (1995), "Smoothing splines for longitudinal data," *Statistics in Medicine*, 14, 1235–1248.

Balestriero, R. and Baraniuk, R. (2018), "A spline theory of deep Learning," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 374–383.

Bergeron, M., Fung, N., Hull, J., Poulos, Z., and Veneris, A. (2022), "Variational autoencoders: A hands-off approach to volatility," *The Journal of Financial Data Science*, 4, 125–138.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017), "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, 112, 859–877.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015), "Importance weighted autoencoders," *arXiv preprint arXiv:1509.00519*.

Chen, Y., Yang, Y., Pan, X., Meng, X., and Hu, J. (2022), "Spatiotemporal fusion network for land surface temperature based on a conditional variational autoencoder," *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1–13.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019), "Neural spline flows," in *Advances in Neural Information Processing Systems*, vol. 32.

Fakhoury, D., Fakhoury, E., and Speleers, H. (2022), "ExSpliNet: An interpretable and expressive spline-based neural network," *Neural Networks*, 152, 332–346.

Gershman, S. J., Hoffman, M. D., and Blei, D. M. (2012), "Nonparametric variational inference," in *Proceedings of the 29th International Conference on Machine Learning*, p. 235–242.

Gu, C. and Qiu, C. (1993), "Smoothing spline density estimation: Theory," *The Annals of Statistics*, 21, 217–234.

Han, S., Liao, X., Dunson, D., and Carin, L. (2016), "Variational gaussian copula inference," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, eds. Gretton, A. and Robert, C. C., Cadiz, Spain, vol. 51 of *Proceedings of Machine Learning Research*, pp. 829–838.

Hastie, T. J. (2017), "Generalized additive models," in *Statistical Models in S*, pp. 249–307.

Huix, T., Majewski, S., Durmus, A., Moulines, E., and Korba, A. (2022), "Variational inference of overparameterized bayesian neural networks: a theoretical and empirical study," *arXiv preprint arXiv:2207.03859*.

Kingma, D. P. and Welling, M. (2013), "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2020), "Normalizing flows: An introduction and review of current methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43, 3964–3979.

Kopf, A., Fortuin, V., Somnath, V. R., and Claassen, M. (2021), "Mixture-of-experts variational autoencoder for clustering and generating from similarity-based representations on single cell data," *PLOS Computational Biology*, 17, e1009086.

Krizhevsky, A., Hinton, G., et al. (2009), "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017), "Automatic differentiation variational inference," *Journal of Machine Learning Research*, 18, 1–45.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998), "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86, 2278–2324.

Li, Y. and Turner, R. E. (2017), "Gradient estimators for implicit models," *arXiv preprint arXiv:1705.07107*.

Loaiza-Ganem, G. and Cunningham, J. P. (2019), "Deep random splines for point process intensity estimation," .

Locatello, F., Khanna, R., Ghosh, J., and Ratsch, G. (2018), "Boosting variational inference: An optimization perspective," in *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, vol. 84, pp. 464–472.

Ma, C., Li, Y., and Hernández-Lobato, J. M. (2019), "Variational implicit processes," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 4222–4233.

Maddison, C., Mnih, A., and Teh, Y. (2017), "The concrete distribution: A continuous relaxation of discrete random variables," in *International Conference on Learning Representations*.

Molchanov, D., Kharitonov, V., Sobolev, A., and Vetrov, D. (2019), "Doubly semi-implicit variational inference," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, vol. 89, pp. 2593–2602.

Morningstar, W., Vikram, S., Ham, C., Gallagher, A., and Dillon, J. (2021), "Automatic differentiation variational inference with mixtures," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, vol. 130, pp. 3250–3258.

Ortega, L. A., Santana, S. R., and Hernández-Lobato, D. (2022), "Deep variational implicit processes," *arXiv preprint arXiv:2206.06720*.

Pang, B., Han, T., Nijkamp, E., Zhu, S.-C., and Wu, Y. N. (2020), "Learning Latent Space Energy-Based Prior Model," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 21994–22008.

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

Rainforth, T., Kosiorek, A., Le, T. A., Maddison, C., Igl, M., Wood, F., and Teh, Y. W. (2018), "Tighter variational bounds are not necessarily better," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 4277–4285.

Rezende, D. and Mohamed, S. (2015), "Variational inference with normalizing flows," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 1530–1538.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014), "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, pp. 1278–1286.

Rezende, D. J., Papamakarios, G., Racaniere, S., Albergo, M., Kanwar, G., Shanahan, P., and Cranmer, K. (2020), "Normalizing flows on tori and spheres," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, pp. 8083–8092.

Rodríguez-Santana, S., Zaldivar, B., and Hernandez-Lobato, D. (2022), "Function-space inference with sparse implicit processes," in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162, pp. 18723–18740.

Roeder, G., Wu, Y., and Duvenaud, D. K. (2017), "Sticking the landing: Simple, lower-variance gradient estimators for variational inference," vol. 30.

Schmidt-Hieber, J. (2020), "Nonparametric regression using deep neural networks with ReLU activation function," *The Annals of Statistics*, 48, 1875 – 1897.

Schumaker, L. (2007), *Spline functions: basic theory*, Cambridge University Press.

Shi, J., Sun, S., and Zhu, J. (2017), "Kernel implicit variational inference," *arXiv preprint arXiv:1705.10119*.

— (2018), "A spectral approach to gradient estimation for implicit distributions," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 4644–4653.

Spiriti, S., Eubank, R., Smith, P. W., and Young, D. (2013), "Knot selection for least-squares and penalized splines," *Journal of Statistical Computation and Simulation*, 83, 1020–1036.

Sriperumbudur, B., Fukumizu, K., Gretton, A., Hyvärinen, A., and Kumar, R. (2017), "Density estimation in infinite dimensional exponential families," *Journal of Machine Learning Research*, 18, 1–59.

Stone, C. J. (1994), "The use of polynomial splines and their tensor products in multivariate function estimation," *The Annals of Statistics*, 22, 118–171.

Sundararajan, M., Taly, A., and Yan, Q. (2017), "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 3319–3328.

Takahashi, H., Iwata, T., Yamanaka, Y., Yamada, M., and Yagi, S. (2019), "Variational autoencoder with implicit optimal priors," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5066–5073.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005), "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 91–108.

Wainwright, M. J., Jordan, M. I., et al. (2008), "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, 1, 1–305.

Wang, J.-L., Chiou, J.-M., and Müller, H.-G. (2016), "Functional data analysis," *Annual Review of Statistics and Its Application*, 3, 257–295.

Wang, L. and Yang, L. (2009), "Spline estimation of single-index models," *Statistica Sinica*, 765–783.

Wang, Y. and Blei, D. M. (2019), "Frequentist consistency of variational Bayes," *Journal of the American Statistical Association*, 114, 1147–1161.

Wood, S. N. (2003), "Thin plate regression splines," *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 65, 95–114.

Wu, H., Köhler, J., and Noe, F. (2020), "Stochastic normalizing flows," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 5933–5944.

Xiao, H., Rasul, K., and Vollgraf, R. (2017), "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*.

Yu, S., Wang, G., Wang, L., Liu, C., and Yang, L. (2020), "Estimation and inference for generalized geoadditive models," *Journal of the American Statistical Association*, 115, 761–774.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. (2018), "Advances in variational inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41, 2008–2026.

Zhang, F. and Gao, C. (2020), "Convergence rates of variational posterior distributions," *The Annals of Statistics*, 48, 2180 – 2207.

Zhou, Y., Shi, J., and Zhu, J. (2020), "Nonparametric score estimators," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, pp. 11513–11522.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes]

   (b) The license information of the assets, if applicable. [Not Applicable]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

   (d) Information about consent from data providers/curators. [Not Applicable]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials for "Nonparametric Automatic Differentiation Variational Inference with Spline Approximation"

S.1 ADDITIONAL LEMMAS AND PROOF DETAILS

S.1.1 Proof of Theorem 4.2

Lemma 4.1 shows that the proposed S-ADVI allows us to quantify the Lower Bound of IWAE.

Proof of Lemma 4.1. We start with a one-dimensional latent variable with a finite support $\mathcal{T}$. The objective function is $\mathcal{L}_{\mathrm{IWAE}}(\phi) = \mathbb{E}_{\{z_t\sim q_\phi(z|\boldsymbol{x})\}_{t=1}^T}\left[\log\frac{1}{T}\sum_{t=1}^T\frac{p_\theta(\boldsymbol{x}|z_t)p(z_t)}{q_\phi(z_t|\boldsymbol{x})}\right] = \int\log\frac{1}{T}\sum_{t=1}^T\frac{p_\theta(\boldsymbol{x}|z_t)p(z_t)}{q_\phi(z_t|\boldsymbol{x})}\cdot\prod_{t=1}^T p(z_t|\boldsymbol{x})dz_t$. According to Remark 2.2, for any posterior $p(z_t|\boldsymbol{x})\in\mathcal{H}^{(\varrho)}(\mathcal{T})$, there exists a spline function $s(z_t)=\sum_{k=1}^K\gamma_k b_k(z_t)$ such that $\sup_{z_t\in\mathcal{T}}|p(z_t|\boldsymbol{x})-s(z_t)|\le CH^{-(\varrho+1)}$. We denote the corresponding spline coefficients as ϕ^*, and the optimal spline function as $\widehat{s}(z_t)$ whose spline coefficients satisfy $\widehat{\phi}=\arg\max\mathcal{L}_{\mathrm{IWAE}}(\phi)$. Therefore, we have $\mathcal{L}_{\mathrm{IWAE}}(\widehat{\phi})\ge\mathcal{L}_{\mathrm{IWAE}}(\phi^*)$. We denote $\Delta(z_t)=p(z_t|\boldsymbol{x})-s^*(z_t)$ and $\sup_{z_t\in\mathcal{T}}|\Delta(z_t)|\le CH^{-(\varrho+1)}$. Then, we have

$$\begin{aligned}
\mathcal{L}_{\mathrm{IWAE}}(\widehat{\phi})\ge\mathcal{L}_{\mathrm{IWAE}}(\phi^*) &= \int\left[\log\frac{1}{T}\sum_{t=1}^T\frac{p_\theta\left(\boldsymbol{x}|z_t\right)p\left(z_t\right)}{p(z_t|\boldsymbol{x})-\Delta(z_t)}\right]\left[\prod_{t=1}^T p(z_t|\boldsymbol{x})-\Delta(z_t)\right]dz_t\\
&= \int\left\{\log\left[p_\theta(\boldsymbol{x})+\frac{1}{T}\sum_{t=1}^T\frac{\Delta(z_t)p_\theta\left(\boldsymbol{x}|z_t\right)p\left(z_t\right)}{\{p(z_t|\boldsymbol{x})-\Delta(z_t)\}p(z_t|\boldsymbol{x})}\right]\right\}\left[\prod_{t=1}^T p(z_t|\boldsymbol{x})-\Delta(z_t)\right]dz_t\\
&\ge \int\left\{\log\left[p_\theta(\boldsymbol{x})-C_1H^{-(\varrho+1)}\right]\right\}\left[\prod_{t=1}^T p(z_t|\boldsymbol{x})-C_2H^{-(\varrho+1)}\right]dz_t\ge\log p_\theta(\boldsymbol{x})-C_3H^{-(\varrho+1)},
\end{aligned}\tag{1}$$

which yields the results in Lemma 4.1.

Next, we consider latent variables with infinite support. For a given ϵ, consider a finite support $\mathcal{T}$ such that $\int_{\mathcal{T}}p\left(z|\boldsymbol{x}\right)dz\ge1-\epsilon$. The spline-based posterior has a finite support on $\mathcal{T}$. According to the definition of IWAE and the fact that the IWAE has a tighter bound than ELBO, we have

$$\begin{aligned}
\mathcal{L}_{\mathrm{IWAE}}(\phi) &= \mathbb{E}_{\{z_t\sim q_\phi(z|\boldsymbol{x})\}_{t=1}^T}\left[\log\frac{1}{T}\sum_{t=1}^T\frac{p_\theta\left(\boldsymbol{x}|z_t\right)p\left(z_t\right)}{q_\phi\left(z_t|\boldsymbol{x}\right)}\right]\\
&= \int_{\mathcal{T}}\left[\log\frac{1}{T}\sum_{t=1}^T\frac{p_\theta\left(\boldsymbol{x}|z_t\right)p\left(z_t\right)}{q_\phi(z_t|\boldsymbol{x})}\right]\left[\prod_{t=1}^T q_\phi(z_t|\boldsymbol{x})\right]dz_1dz_2\ldots dz_T\\
&\quad+\int_{\mathbb{R}/\mathcal{T}}\left[\log\frac{1}{T}\sum_{t=1}^T\frac{p_\theta\left(\boldsymbol{x}|z_t\right)p\left(z_t\right)}{q_\phi(z_t|\boldsymbol{x})}\right]\left[\prod_{t=1}^T q_\phi(z_t|\boldsymbol{x})\right]dz_1dz_2\ldots dz_T\\
&\ge \int_{\mathcal{T}}\left[\log\frac{p_\theta\left(\boldsymbol{x}|z\right)p\left(z\right)}{q_\phi(z|\boldsymbol{x})}\right]q_\phi(z|\boldsymbol{x})dz+\int_{\mathbb{R}/\mathcal{T}}\left[\log\frac{p_\theta\left(\boldsymbol{x}|z\right)p\left(z\right)}{q_\phi(z|\boldsymbol{x})}\right]q_\phi(z|\boldsymbol{x})dz=\int_{\mathcal{T}}\left[\log\frac{p_\theta\left(\boldsymbol{x}|z\right)p\left(z\right)}{q_\phi(z|\boldsymbol{x})}\right]q_\phi(z|\boldsymbol{x})dz\\
&= \log p_\theta(\boldsymbol{x})+\int_{\mathcal{T}}\left[\log\frac{p\left(z|\boldsymbol{x}\right)}{q_\phi(z|\boldsymbol{x})}\right]q_\phi(z|\boldsymbol{x})dz\ge\log p_\theta(\boldsymbol{x})+\int_{\mathcal{T}}\left[\log\frac{p^{\mathrm{T}}\left(z|\boldsymbol{x}\right)}{q_\phi(z|\boldsymbol{x})}+\log(1-\epsilon)\right]q_\phi(z|\boldsymbol{x})dz,
\end{aligned}$$

where $p^{\mathrm{T}}\left(z|\boldsymbol{x}\right)=\left(\int_{\mathcal{T}}p\left(z|\boldsymbol{x}\right)dz\right)^{-1}p\left(z|\boldsymbol{x}\right)$ is the density function for random variable $z|\boldsymbol{x}$ truncated on the interval $\mathcal{T}$. Notice that when $\mathcal{T}$ is properly chosen and we can apply the conclusion from (1)

$$\log p_\theta(\boldsymbol{x})+\int_{\mathcal{T}}\left[\log\frac{p^{\mathrm{T}}\left(z|\boldsymbol{x}\right)}{q_\phi(z|\boldsymbol{x})}+\log(1-\epsilon)\right]q_\phi(z|\boldsymbol{x})dz\ge\log p_\theta(\boldsymbol{x})-C_3H^{-(\varrho+1)}-C_3\epsilon,$$

which yields the results in Lemma 4.1.

It is straightforward to extend the above results to the multivariate case. If Assumption (A2) holds, then we have

$$\int_{\mathcal{T}_1\cdots\mathcal{T}_J}\left[\log\frac{p\left(\boldsymbol{z}|\boldsymbol{x}\right)}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right]q_\phi(\boldsymbol{z}|\boldsymbol{x})d\boldsymbol{z}=\int_{\mathcal{T}_1\cdots\mathcal{T}_J}\sum_{j=1}^{J}\left[\log\frac{p\left(z_j|\boldsymbol{x}\right)}{q_\phi(z_j|\boldsymbol{x})}\right]\left[\prod_{j=1}^{J}q_\phi(z_j|\boldsymbol{x})\right]dz_1\cdots dz_J$$
$$=\sum_{j=1}^{J}\int_{\mathcal{T}_j}\left[\log\frac{p\left(z_j|\boldsymbol{x}\right)}{q_\phi(z_j|\boldsymbol{x})}\right]q_\phi(z_j|\boldsymbol{x})dz_j.$$

Similar to the results in (1), we can obtain the lower bound of IWAE based on the S-ADVI is $\log p_\theta(\boldsymbol{x})-CJH^{-(\varrho+1)}-J\epsilon$. □

According to the results in Lemma 4.1, we can further quantify the variational approximation error with respect to the class defined in (2).

Proof of Theorem 4.2. According to Remark 2.2 and similar to the Proof of Lemma 4.1, there exists $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ such that $\sup_{\boldsymbol{z}}|p(\boldsymbol{z}|\boldsymbol{x})-q_\phi(\boldsymbol{z}|\boldsymbol{x})|\leq CJH^{-(\varrho+1)}$. When H goes to infinity, $JH^{-(\varrho+1)}$ goes to zero. One can also infer that $|p(\boldsymbol{z}|\boldsymbol{x})-q_{\widehat{\phi}}(\boldsymbol{z}|\boldsymbol{x})|\leq CJH^{-(\varrho+1)}$ almost everywhere on $\mathcal{T}_1\times\cdots\times\mathcal{T}_J$. Then, the following property holds that is,

$$\begin{aligned}
\mathcal{L}_{\text{IWAE}}\left(\widehat{\phi}\right)&=\mathbb{E}_{\left\{q_{\widehat{\phi}}(\boldsymbol{z}_t|\boldsymbol{x})\right\}_{t=1}^{T}}\left[\log\frac{1}{T}\sum_{t=1}^{T}\frac{p\left(\boldsymbol{z}_t|\boldsymbol{x}\right)}{q_{\widehat{\phi}}\left(\boldsymbol{z}_t|\boldsymbol{x}\right)}\right]+\log p_\theta(\boldsymbol{x})\\
&=\mathbb{E}_{\left\{q_{\widehat{\phi}}(\boldsymbol{z}_t|\boldsymbol{x})\right\}_{t=1}^{T}}\left\{\log\left[1+\frac{1}{T}\sum_{t=1}^{T}\frac{p\left(\boldsymbol{z}_t|\boldsymbol{x}\right)-q_{\widehat{\phi}}\left(\boldsymbol{z}_t|\boldsymbol{x}\right)}{q_{\widehat{\phi}}\left(\boldsymbol{z}_t|\boldsymbol{x}\right)}\right]\right\}+\log p_\theta(\boldsymbol{x})\\
&=\mathbb{E}_{\left\{q_{\widehat{\phi}}(\boldsymbol{z}_t|\boldsymbol{x})\right\}_{t=1}^{T}}\left\{\frac{1}{T}\sum_{t=1}^{T}\frac{p\left(\boldsymbol{z}_t|\boldsymbol{x}\right)-q_{\widehat{\phi}}\left(\boldsymbol{z}_t|\boldsymbol{x}\right)}{q_{\widehat{\phi}}\left(\boldsymbol{z}_t|\boldsymbol{x}\right)}+o\left[JH^{-(\varrho+1)}\right]\right\}+\log p_\theta(\boldsymbol{x})\\
&=\mathbb{E}_{q_{\widehat{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left\{\frac{p\left(\boldsymbol{z}|\boldsymbol{x}\right)-q_{\widehat{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}\right)}{q_{\widehat{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}\right)}+o\left[JH^{-(\varrho+1)}\right]\right\}+\log p_\theta(\boldsymbol{x})\\
&=\mathbb{E}_{q_{\widehat{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left\{\log\frac{p\left(\boldsymbol{z}|\boldsymbol{x}\right)}{q_{\widehat{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}\right)}+o\left[JH^{-(\varrho+1)}\right]\right\}+\log p_\theta(\boldsymbol{x})\\
&=-D_{\text{KL}}\left[q_{\widehat{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}\right)||p\left(\boldsymbol{z}|\boldsymbol{x}\right)\right]+\log p_\theta(\boldsymbol{x})+o\left[JH^{-(\varrho+1)}\right]
\end{aligned}$$

The result in Lemma 4.1 implies the KL divergence of the spline density approximation from the true posterior is in the order of $JH^{-(\varrho+1)}+J\epsilon$. □

S.1.2 Proof of Theorem 4.3

LEMMA **1** (Lemma 6.3 (Csiszár and Talata, 2006))**.** *If p and q are probability densities both supported on a bounded interval $\mathcal{T}$, then we have the KL divergence between probability densities satisfies that $D_{\text{KL}}(p||q)\leq\frac{1}{\inf_{x\in\mathcal{T}}q(x)}\|p-q\|_2^2$.*

Proof. Notice that

$$D_{\text{KL}}(p||q)=\int_{\mathcal{T}}p(x)\log\frac{p(x)}{q(x)}\mathrm{d}x\leq\int_{\mathcal{T}}p(x)\left\{\frac{p(x)}{q(x)}-1\right\}\mathrm{d}x=\int_{\mathcal{T}}\frac{\{p(x)-q(x)\}^2}{q(x)}\mathrm{d}x$$

from which the claim follows. □

Here we provide the proof sketch of Theorem 4.3. Our proof is based on the assumption when the observed data points $\boldsymbol{x}$ and $\boldsymbol{x}'$ are close enough, the corresponding posteriors $p(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z}|\boldsymbol{x}')$ are similar to each other. For

any given posterior $p(\boldsymbol{z}|\boldsymbol{x})$ under satisfying Assumption (A2), we can identify the density function based on spline approximation $\prod_{j=1}^{J}\sum_{k=1}^{K}\gamma_{jk}^{*}(\boldsymbol{x})b_{k,\mathcal{T}_j^*}(z_j)$ with $\mathcal{T}_j^* = [\mu_j^*(\boldsymbol{x}), \mu_j^*(\boldsymbol{x}) + \sigma_j^*(\boldsymbol{x})]$ close to $p(\boldsymbol{z}|\boldsymbol{x})$ with differences bounded by $JH^{-(\varrho+1)}$. Under some mild assumptions, $\gamma_{jk}^*(\boldsymbol{x})$, $k = 1, \ldots, K$, $\mu_j^*(\boldsymbol{x})$, and $\sigma_j^*(\boldsymbol{x})$ can be well approximated by nonparametric regression, such as the deep neural network. Combining the results in Theorem 4.2, we can further obtain the KL divergence between the proposed S-ADVI estimator and the true posterior.

Proof of Theorem 4.3. In the following, we denote that the unknown parameters as $\phi = \{\mu_j, \sigma_j, \gamma_{jk}, j = 1, \ldots, J, k = 1, \ldots, K\}$ and $\widehat{\phi}(\boldsymbol{x}) = \arg\max_{\phi}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{ELBO}}\{\phi(\boldsymbol{x}_i)\}$. We notice that $\sum_{i=1}^{n}\mathcal{L}_{\mathrm{ELBO}}\{\widehat{\phi}(\boldsymbol{x}_i)\} = \sum_{i=1}^{n}\log p_\theta(\boldsymbol{x}_i) - \sum_{i=1}^{n} D_{KL}\left\{q_{\widehat{\phi}(\boldsymbol{x}_i)}(\boldsymbol{z}) \mid\mid p(\boldsymbol{z}|\boldsymbol{x}_i)\right\}$, we can infer to the average of the KL divergence between $q_{\widehat{\phi}(\boldsymbol{x}_i)}(\boldsymbol{z})$ and $p(\boldsymbol{z}|\boldsymbol{x}_i)$ equaling to $n^{-1}\sum_{i=1}^{n}\log p_\theta(\boldsymbol{x}_i) - n^{-1}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{ELBO}}\{\widehat{\phi}(\boldsymbol{x}_i)\}$. In addition, for a given ξ, consider a finite support $\mathcal{T}_j^* = [\mu_j^*(\boldsymbol{x}), \mu_j^*(\boldsymbol{x}) + \sigma_j^*(\boldsymbol{x})]$ such that $\int_{\mathcal{T}_j^*} p\left(z_j|\boldsymbol{x}\right) dz_j \geq 1 - \xi$. For a specific posterior $p(\boldsymbol{z}|\boldsymbol{x})$, the optimal parameters are $\phi^*(\boldsymbol{x}) = \left\{\mu_j^*(\boldsymbol{x}), \sigma_j^*(\boldsymbol{x}), \gamma_{jk}^*(\boldsymbol{x}), j = 1, \ldots, J, k = 1, \ldots, K\right\}$, where $\gamma_{jk}^*(\boldsymbol{x})$'s are spline coefficients satisfying that $\sup_{z_j \in \mathcal{T}_j^*}|p(z_j|\boldsymbol{x}) - s^*(z_j;\boldsymbol{x})| \leq CH^{-(\varrho+1)}$, where $s^*(z_j;\boldsymbol{x}) = \sum_{k=1}^{K}\gamma_{jk}^*(\boldsymbol{x})b_k(z_j)$.

Next, we denote the estimators of the optimal parameters generated from nonparametric regression, such as the deep neural network, as $\widetilde{\phi}(\boldsymbol{x}) = \{\widetilde{\mu}_j(\boldsymbol{x}), \widetilde{\sigma}_j(\boldsymbol{x}), \widetilde{\gamma}_{jk}(\boldsymbol{x}), j = 1, \ldots, J, k = 1, \ldots, K\}$. When the observed datapoints $\boldsymbol{x}$ and $\boldsymbol{x}'$ are close enough, the corresponding posteriors $p(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z}|\boldsymbol{x}')$ are close, so do the corresponding optimal parameters $\phi^*(\boldsymbol{x})$ and $\phi^*(\boldsymbol{x}')$. Then, we have

$$
\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}\left\{q_{\widehat{\phi}(\boldsymbol{x}_i)}(\boldsymbol{z}) \mid\mid p(\boldsymbol{z}|\boldsymbol{x}_i)\right\} \\
&\quad= \frac{1}{n}\sum_{i=1}^{n}\log p_\theta(\boldsymbol{x}_i) - \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{ELBO}}\{\widehat{\phi}(\boldsymbol{x}_i)\} \leq \frac{1}{n}\sum_{i=1}^{n}\log p_\theta(\boldsymbol{x}_i) - \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{ELBO}}\{\widetilde{\phi}(\boldsymbol{x}_i)\} \\
&\quad= \frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}\left\{q_{\widetilde{\phi}(\boldsymbol{x}_i)}(\boldsymbol{z}) \mid\mid p(\boldsymbol{z}|\boldsymbol{x}_i)\right\}.
\end{aligned}
$$

According to Lemma 1, we have

$$
\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}\left\{q_{\widetilde{\phi}(\boldsymbol{x}_i)}(\boldsymbol{z}) \mid\mid p(\boldsymbol{z}|\boldsymbol{x}_i)\right\} \\
&\quad= \frac{1}{n}\sum_{i=1}^{n}\int q_{\widetilde{\phi}(\boldsymbol{x}_i)}(\boldsymbol{z})\log\frac{q_{\widetilde{\phi}(\boldsymbol{x}_i)}(\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x}_i)}\mathrm{d}\boldsymbol{z} = \sum_{i=1}^{n}\sum_{j=1}^{J}\int q_{\widetilde{\phi}(\boldsymbol{x}_i)}(z_j)\log\frac{q_{\widetilde{\phi}(\boldsymbol{x}_i)}(z_j)}{p(z_j|\boldsymbol{x}_i)}\mathrm{d}z_j \\
&\quad\leq \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J}\frac{1}{\inf_{z_j\in\widetilde{\mathcal{T}}_j} p(z_j|\boldsymbol{x}_i)}\|p(z_j|\boldsymbol{x}_i) - q_{\widetilde{\phi}(\boldsymbol{x}_i)}(z_j)\|^2 \\
&\quad\leq \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J}\frac{2}{\inf_{z_j\in\widetilde{\mathcal{T}}_j} p(z_j|\boldsymbol{x}_i)}\left\{\|p(z_j|\boldsymbol{x}_i) - q_{\phi^*(\boldsymbol{x}_i)}(z_j)\|^2 + \|q_{\widetilde{\phi}(\boldsymbol{x}_i)}(z_j) - q_{\phi^*(\boldsymbol{x}_i)}(z_j)\|^2\right\} \\
&\quad\leq C_1(JH^{-2(\varrho+1)} + J\epsilon^2) + C_2JH^2\Delta^2,
\end{aligned}
$$

where $\widetilde{\mathcal{T}}_j$ is the support of density function $q_{\widetilde{\phi}(\boldsymbol{x}_i)}(z_j)$ and $\widetilde{\mathcal{T}}_j = [\widetilde{\mu}_j(\boldsymbol{x}), \widetilde{\mu}_j(\boldsymbol{x}) + \widetilde{\sigma}_j(\boldsymbol{x})]$. According to the properties of spline basis functions, we have $n^{-1}\sum_{i=1}^{n}\|q_{\widetilde{\phi}(\boldsymbol{x}_i)}(z_j) - q_{\phi^*(\boldsymbol{x}_i)}(z_j)\|^2 \leq CH\sum_{j=1}^{J}\sum_{k=1}^{K}\{\gamma_{jk}^*(\boldsymbol{x}_i) - \widetilde{\gamma}_{jk}(\boldsymbol{x}_i)\}^2 = O(H^2\Delta^2)$. □

S.2 IMPLEMENTATION DETAILS

S.2.1 Spline Roughness Penalty

We introduce the details of constructing penalty $\mathbf{P}$ matrix. Following the definition of penalty matrix $\mathbf{P}$ in the Section 3.2, for any spline polynomials $s(t) = \sum_{k=1}^{K} \gamma_k b_k(t)$, we can further implement as

$$\mathcal{E}(s) = \int_{\mathcal{T}} \left\{ \sum_{k=1}^{K} \gamma_k b_k''(t) \right\}^2 dt = \sum_{k=1}^{K} \sum_{k'=1}^{K} \gamma_k \gamma_{k'} \int_{\mathcal{T}} b_k''(t) b_{k'}''(t) dt = \boldsymbol{\gamma}^\top \mathbf{P} \boldsymbol{\gamma}, \tag{1}$$

where $\mathbf{P}$ is a $K \times K$ matrix with entries $\int_{\mathcal{T}} b_k''(t) b_{k'}''(t) dt$. Denote that $\boldsymbol{b}''(t)$ as the vector of the second-order derivatives of the spline basis $\boldsymbol{b}(t)$ and $\boldsymbol{b}''(t) = \mathbf{D}^{(2)}\boldsymbol{b}(t) = \mathbf{M}_1\mathbf{M}_2\boldsymbol{b}_{\varrho-2}(t)$ where $\boldsymbol{b}_{\varrho-2}(t)$ is the vector of spline basis with degree $\varrho - 2$ and

$$\mathbf{M}_\ell = (\varrho + 1 - \ell) \begin{pmatrix} \frac{-1}{\upsilon_1 - \upsilon_{-\varrho+\ell}} & 0 & 0 & \cdots 0 & 0 \\ \frac{1}{\upsilon_1 - \upsilon_{-\varrho+\ell}} & \frac{-1}{\upsilon_2 - \upsilon_{1-\varrho+\ell}} & 0 & \cdots 0 & 0 \\ 0 & \frac{1}{\upsilon_2 - \upsilon_{1-\varrho+\ell}} & \frac{-1}{\upsilon_3 - \upsilon_{2-\varrho+\ell}} & \cdots 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{\upsilon_{H+\varrho+1-\ell} - \upsilon_{H+\varrho+1}} \end{pmatrix}, \text{ for } \ell = 1, 2.$$

S.2.2 Details of Reparameterization Trick

LEMMA **2.** *If $\boldsymbol{z} = g(\boldsymbol{\epsilon})$, $\boldsymbol{g}$ is a monotone function, then the density function of $\boldsymbol{z}$ is $\boldsymbol{f}_{\boldsymbol{z}}(\boldsymbol{z}) = \boldsymbol{f}_{\boldsymbol{\epsilon}}\{\boldsymbol{g}^{-1}(\boldsymbol{z})\} \left| \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\epsilon}} \right|$, where $\left| \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\epsilon}} \right|$ is the determinate of the Jacob matrix $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\epsilon}}$.*

In the following, we derive that

$$\mathcal{L}_{\text{IWAE}}(\phi) = \mathbb{E}_{\{\boldsymbol{\epsilon}_t\}_{t=1}^T} \left[\log \frac{1}{T} \sum_{t=1}^{T} \frac{p_\theta \left\{ \boldsymbol{x}, \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{\epsilon}_t \right\}}{\prod_{j=1}^{J} \left\{ \sum_{k=1}^{K} \gamma_{jk}(\boldsymbol{x}) b_k\left(\epsilon_{jt}\right) \right\}} \right] + \sum_{j=1}^{J} \log \sigma_j(\boldsymbol{x}).$$

Note that $\boldsymbol{z}_j = \mu_j(\boldsymbol{x}) + \sigma_j(\boldsymbol{x})\epsilon_j$. Applying the conclusion in Lemma 2, the function $q_\phi(z_j|\boldsymbol{x})$ is

$$\frac{1}{\sigma_j(\boldsymbol{x})} \sum_{k=1}^{K} \gamma_{jk}(\boldsymbol{x}) b_k \left\{ \frac{z_j - \mu_j(\boldsymbol{x})}{\sigma_j(\boldsymbol{x})} \right\}. \tag{2}$$

Plugging in (2) to $\mathcal{L}_{\text{IWAE}}(\phi)$, we have

$$\begin{aligned}
\mathcal{L}_{\text{IWAE}}(\phi) &= \mathbb{E}_{\{q_\phi(\boldsymbol{z}_t|\boldsymbol{x})\}_{t=1}^T} \left[\log \frac{1}{T} \sum_{t=1}^{T} \frac{p_\theta\left(\boldsymbol{x}|\boldsymbol{z}_t\right) p\left(\boldsymbol{z}_t\right)}{q_\phi\left(\boldsymbol{z}_t|\boldsymbol{x}\right)} \right] \\
&= \int \left[\log \frac{1}{T} \sum_{t=1}^{T} \frac{p_\theta\left(\boldsymbol{x}, \boldsymbol{z}_t\right)}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x})} \right] \left[\prod_{t=1}^{T} q_\phi(\boldsymbol{z}_t|\boldsymbol{x}) \right] d\boldsymbol{z}_t \\
&= \int \left[\log \frac{1}{T} \sum_{t=1}^{T} \frac{p_\theta\left(\boldsymbol{x}, \boldsymbol{z}_t\right)}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x})} \right] \left\{ \prod_{t=1}^{T} \prod_{j=1}^{J} \frac{1}{\sigma_j(\boldsymbol{x})}(\boldsymbol{x}) \sum_{k=1}^{K} \gamma_{jk}(\boldsymbol{x}) b_k \left[\frac{z_j - \mu_j(\boldsymbol{x})}{\sigma_j(\boldsymbol{x})} \right] \right\} d\boldsymbol{z}_t \\
&= \int \left[\log \frac{1}{T} \sum_{t=1}^{T} \frac{p_\theta\left(\boldsymbol{x}, \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{\epsilon}_t\right)}{\prod_{j=1}^{J} \frac{1}{\sigma_j(\boldsymbol{x})} \sum_{k=1}^{K} \gamma_{jk}(\boldsymbol{x}) b_k(\epsilon_{jt})} \right] \left\{ \prod_{t=1}^{T} \prod_{j=1}^{J} \left[\sum_{k=1}^{K} \gamma_{jk}(\boldsymbol{x}) b_k(\epsilon_{jt}) \right] \right\} d\boldsymbol{\epsilon}_t \\
&= \int \left[\log \frac{1}{T} \sum_{t=1}^{T} \frac{p_\theta\left(\boldsymbol{x}, \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{\epsilon}_t\right)}{\prod_{j=1}^{J} \sum_{k=1}^{K} \gamma_{jk}(\boldsymbol{x}) b_k(\epsilon_{jt})} \right] \left\{ \prod_{t=1}^{T} \prod_{j=1}^{J} \left[\sum_{k=1}^{K} \gamma_{jk}(\boldsymbol{x}) b_k(\epsilon_{jt}) \right] \right\} d\boldsymbol{\epsilon}_t + \sum_{j=1}^{J} \log \sigma_j(\boldsymbol{x}) \\
&= \mathbb{E}_{\{\boldsymbol{\epsilon}_t\}_{t=1}^T} \left[\log \frac{1}{T} \sum_{t=1}^{T} \frac{p_\theta \left\{ \boldsymbol{x}, \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{\epsilon}_t \right\}}{\prod_{j=1}^{J} \left\{ \sum_{k=1}^{K} \gamma_{jt}(\boldsymbol{x}) b_k\left(\epsilon_{jt}\right) \right\}} \right] + \sum_{j=1}^{J} \log \sigma_j(\boldsymbol{x}).
\end{aligned}$$

S.2.3 Posterior Collapse

In generative models, posterior collapse means that the posterior of the latent variables equals their prior, that is, $p(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{z})$ (Wang et al., 2021). ADVI with Gaussian mixture approximation (GM-ADVI) is one approach to generate flexible posterior approximation (Morningstar et al., 2021). Nonetheless, it may suffer from posterior collapse. An example of the GM-ADVI collapse is given as follows. The latent variable $\boldsymbol{z}$ is generated from a mixture distribution: $p(u) = \text{Categorical}(1/M)$, $p(\boldsymbol{z}|u) = N(\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u)$, where u takes values from 1 to M, $\boldsymbol{\mu}_m$'s are d-dimensional, and $\boldsymbol{\Sigma}_m$ are $d \times d$-dimensional covariance matrix. The observed data $\boldsymbol{x}$ follows $p(\boldsymbol{x}|\boldsymbol{z}; f, \sigma) = N\left(f(\boldsymbol{z}), \sigma^2\mathbf{I}\right)$. Then, the marginal distribution of $\boldsymbol{z}$ is $1/M \sum_{m=1}^{M} N(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$, which can be well approximated via GM-ADVI. We are interested in estimating the latent class u and the conditional distribution of $\boldsymbol{z}|u$ for a given data point $\boldsymbol{x}$. In this case, if we have $\boldsymbol{z}|u = 1/M \sum_{m=1}^{M} N(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ for all the classes, we can still fully capture the mixture distribution of the data. However, all the M mixture components are the same, and thus the latent variable is non-identifiable. In contrast to the above GM-ADVI example, the proposed S-ADVI is based on a set of pre-specified spline density functions and is constrained not to capture the marginal distribution of $\boldsymbol{z}$, preventing the mixture components from shrinking to the prior distribution. Therefore, posterior collapse can be naturally avoided via spline density functions.

The phenomenon that the posterior is approximately (as opposed to exactly) equal to the prior can also lead to posterior collapse. In S-ADVI, one can easily control the phenomena by adding regularity constraints on spline coefficients. Specifically, the regularity constraint can be written as $\{\|\boldsymbol{\gamma}_j(\boldsymbol{x}) - \boldsymbol{\gamma}_0\|\}^{-1}$, where $\boldsymbol{\gamma}_j(\boldsymbol{x})$ is the spline coefficient vector for j-th latent variable and $\boldsymbol{\gamma}_0$ is the spline coefficient vector whose corresponding density function is closest to the prior. The regularity constraint prevents the posteriors from shrinking to the prior distribution.

S.3 ADDITIONAL EXPERIMENT RESULTS

S.3.1 Additional Results in the Posterior Approximation

In Figure S.1 and Table S.1, we present additional results showcasing posterior approximations using S-ADVI, Gaussian-ADVI, and GM-ADVI for Cases 1 through 5, considering varying values of x. Our findings demonstrate that S-ADVI outperforms other methods in most scenarios. Also, in Figure S.2, we compare the convergence of S-ADVI against GM-ADVI for the five cases.

Table S.1: Mean And Standard Deviation (in Bracket) Of Root Integrated Squared Error In Posterior Approximation Based On Gaussian-ADVI, GM-ADVI, Neural Spline, Planar, Radial, And S-ADVI

Method	Case 1			Case 2			Case 3		Case 4			Case 5		
	$x=0$	$x=1$	$x=2$	$x=0$	$x=1$	$x=2$	$x=0$	$x=1$	$x=7$	$x=8$	$x=9$	$x=0.6$	$x=0.7$	$x=0.8$
Gaussian-ADVI	0.295	0.408	0.713	0.268	0.239	0.251	0.630	0.529	0.631	0.876	1.14	0.249	0.246	0.243
	(0.148)	(0.274)	(0.381)	(0.057)	(0.044)	(0.064)	(0.239)	(0.229)	(0.386)	(0.394)	(0.431)	(0.014)	(0.013)	(0.014)
GM-ADVI	0.206	0.353	0.174	0.237	0.211	0.226	0.403	0.171	0.395	0.317	0.487	0.123	0.136	0.137
	(0.103)	(0.152)	(0.095)	(0.031)	(0.026)	(0.040)	(0.115)	(0.121)	(0.107)	(0.096)	(0.134)	(0.017)	(0.019)	(0.020)
Neural Spline	0.402	0.295	0.335	0.356	0.158	0.324	0.528	0.231	0.956	1.059	1.299	0.476	0.486	0.497
	(0.055)	(0.055)	(0.053)	(0.048)	(0.032)	(0.048)	(0.077)	(0.082)	(0.075)	(0.147)	(0.197)	(0.007)	(0.007)	(0.007)
Planar	0.367	0.279	0.345	0.369	0.182	0.316	0.384	0.390	1.043	1.209	1.400	0.470	0.480	0.491
	(0.046)	(0.055)	(0.051)	(0.045)	(0.031)	(0.052)	(0.099)	(0.105)	(0.241)	(0.160)	(0.123)	(0.009)	(0.010)	(0.010)
Radial	0.274	0.258	0.419	0.499	0.212	0.198	0.440	0.283	0.927	1.174	1.480	0.484)	0.495	0.504
	(0.053)	(0.058)	(0.045)	(0.043)	(0.026)	(0.038)	(0.102)	(0.112)	(0.155)	(0.156)	(0.183)	(0.010)	(0.010)	(0.011)
S-ADVI	**0.094**	**0.086**	**0.088**	**0.088**	**0.054**	**0.101**	**0.211**	**0.146**	**0.310**	**0.323**	**0.371**	**0.101**	**0.099**	**0.097**
	(0.029)	(0.046)	(0.054)	(0.027)	(0.020)	(0.028)	(0.070)	(0.041)	(0.080)	(0.101)	(0.141)	(0.024)	(0.025)	(0.025)

S.3.2 The Effect of Hyper-tuning Parameters in Posterior Approximation

We aim to evaluate the effects of hyperparameters, such as the impact of roughness penalty parameters, temperature decay, and the number of random samples in IWAE. See Figure S.3 for an illustration of hyperparameter effects. For both Case 1 and Case 2 in Figure S.3, we notice that the spline roughness penalty can balance the bias and variance. When set to be small, the estimated curve is more wiggling than the scenario with a large roughness penalty. When the roughness penalty is large, the model is overly smooth. The model is not very sensitive to the speed of decay of temperature (Temp.Decay). But model fitness is less than ideal when it is too large or small for different reasons. For example, from Figure 2, both Case 1 and Case 2 show good performances except when

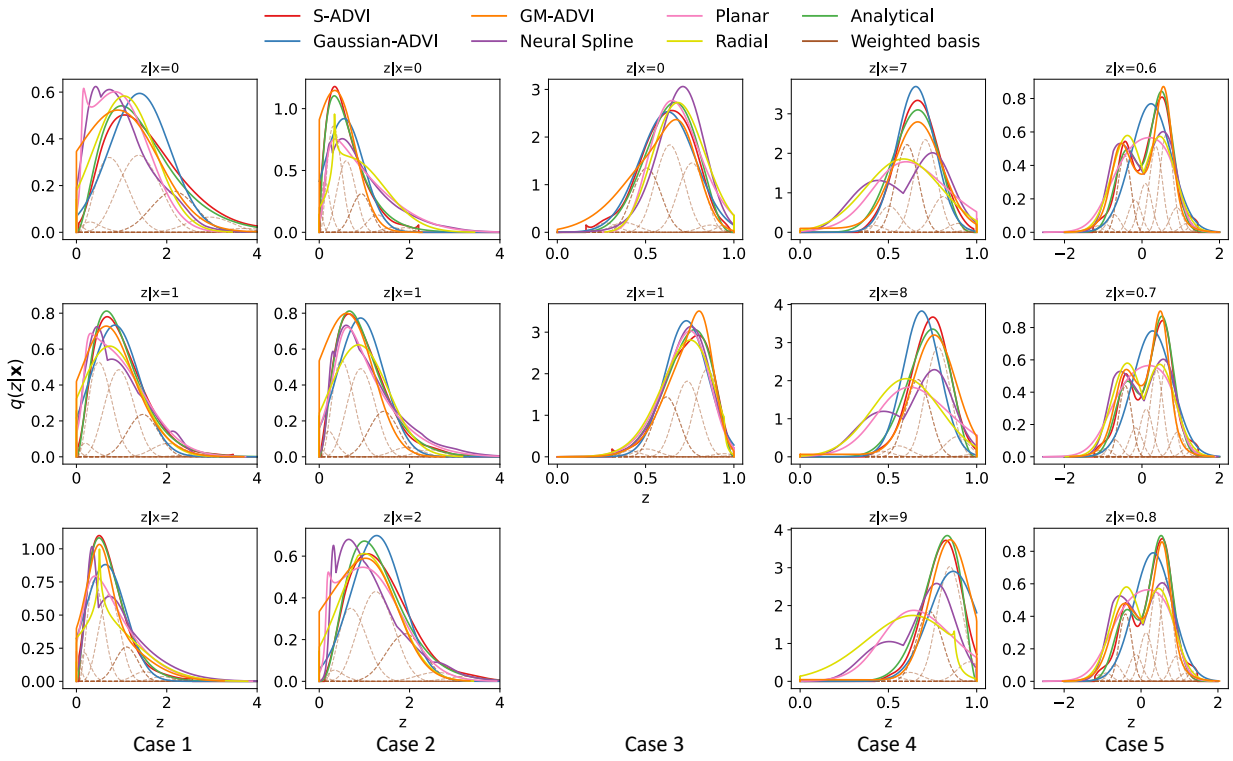
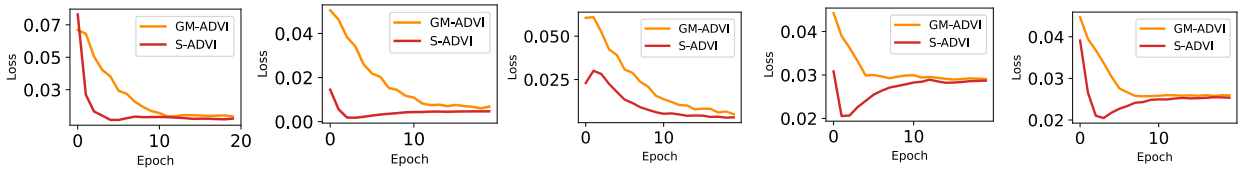


Figure S.1: Visualization Of Approximated Posterior Based On The S-ADVI, Gaussian-ADVI, GM-ADVI, Neural Spline, Planar, And Radial For Cases 1 – 5



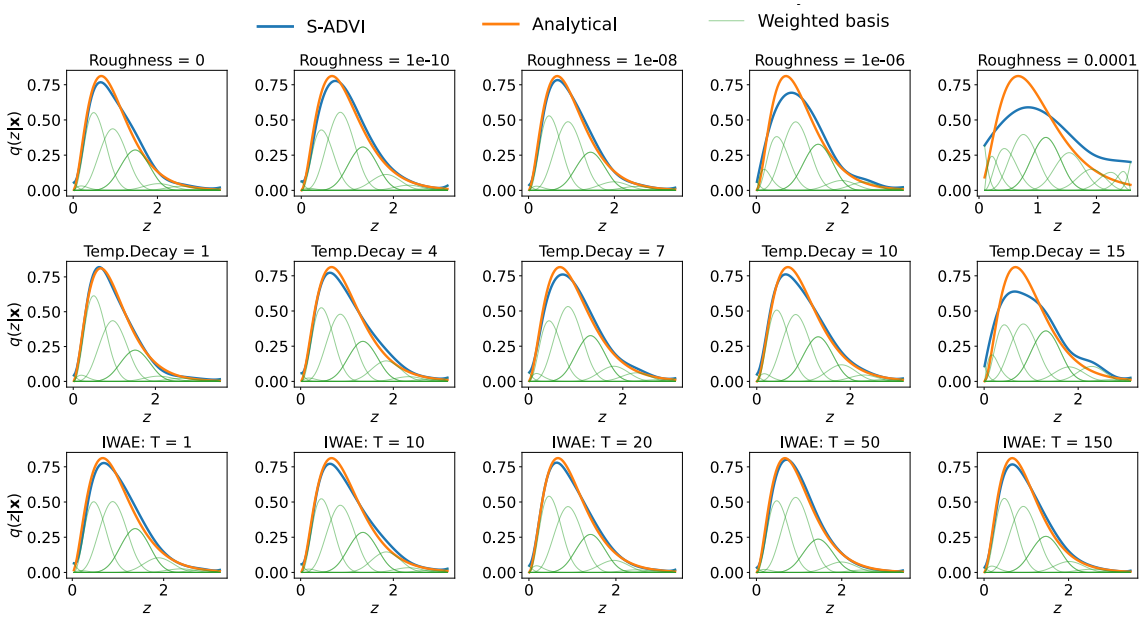
(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4 (e) Case 5

Figure S.2: Comparison Of Convergence Between S-ADVI And GM-ADVI

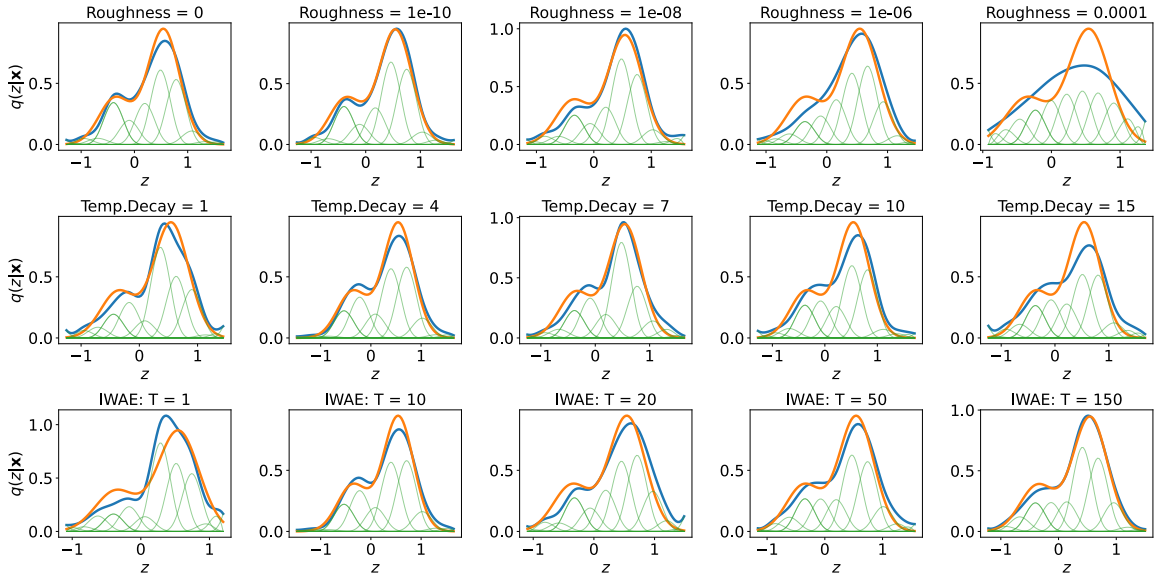
Temp.Decay = 1 or 15. When it is too small, concrete distribution cannot approximate categorical distribution. When it is too large, in the training process, some components may not be fully explored. For the IWAE random samples, the results in Figure S.3 are consistent with previous works. In Figure S.3 (a), for the Gamma posterior distribution, approximation performance varies little with the increasing number of random samples. However, when the posterior distribution is multimodal, the approximation is beneficial from the IWAE random samples.

S.3.3 Additional Results for Single Column Classification

Figure S.4 represents an illustration example of an input sample. The black column is the input for image classification. We train the model with $H = \{1, 3, 5, 7, 9\}$ and $J = \{2, 4, 6, 8\}$. We set $v_0 = 0$ and $v_{H+1} = 1$, choose v_h with equal space for $h = 1, \ldots, H$, and set $T = 10$ for the IWAE. The encoder is initialized as an MLP with two layers of 512 and 256 hidden nodes and ELU activation function (Clevert et al., 2016). The output layer predicts the parameters for distribution over J latent variables. All models are tuned with 10-fold cross-validation and trained for 50 epochs using Adam optimizer with a learning rate 0.001 decayed by 5% for every epoch. We use the $\beta = 0.05$ penalty on the KL divergence term. We choose $\eta = 4$, $\Lambda_0 = 1$, $\Lambda_1 = 0.05$ in annealing such that the temperature decreases from 1 to 0.05 in 15 epochs. Figure S.5 presents the performance comparison for single-column classification for the MNIST dataset with a range of latent variables J and interior knots H. Error



(a) Case 1: Gamma-Exponential



(b) Case 2: Gaussian Mixture

Figure S.3: Hyperparameter Effects On S-ADVI

bars denote the standard error. Figure S.6 demonstrates the convergence of S-ADVI for MINST classification.

To evaluate the influence of hyperparameters, we also test the prediction performance of our method when $J > 8$ and the number of basis spline functions is larger. For latent dimensions $J > 8$, we observe similar performance with average accuracy 0.683 at $J = 10$, 0.686 at $J = 12$, and 0.684 at $J = 14$ for S-ADVI with $H = 3$. In comparison, GM-ADVI achieves 0.679, 0.683, 0.680, and Radial achieves 0.667, 0.673, 0.675, respectively. Given the marginal performance gains with additional latent dimensions, these results were omitted. While penalized spline mitigates overfitting, excessive spline bases lower performance (accuracy drops from 0.680 at $H = 15$ to 0.674 at $H = 30$).

Figure S.4: An Illustration Of A Single Column Of An Input Sample, Where The Black Column Is The Input For Image Classification

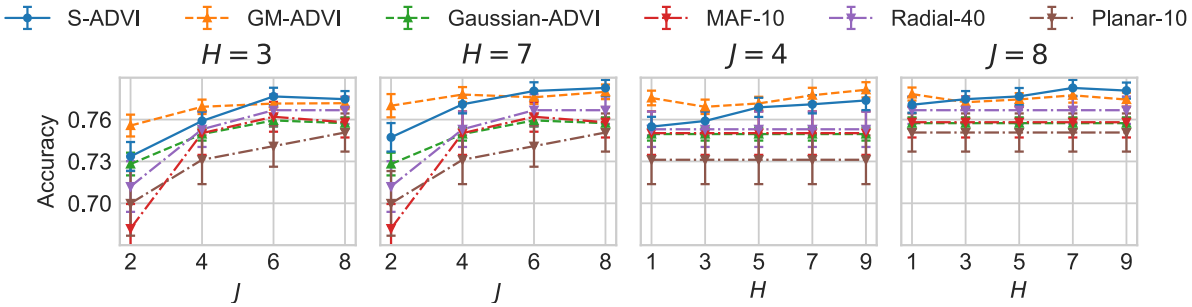


Figure S.5: Performance Comparison For Single Column Classification For The MNIST Dataset With A Range Of Latent Variables J And Interior Knots H (Error Bars Denote The Standard Error)

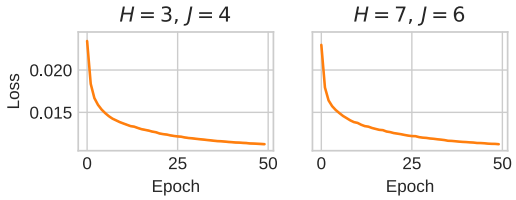
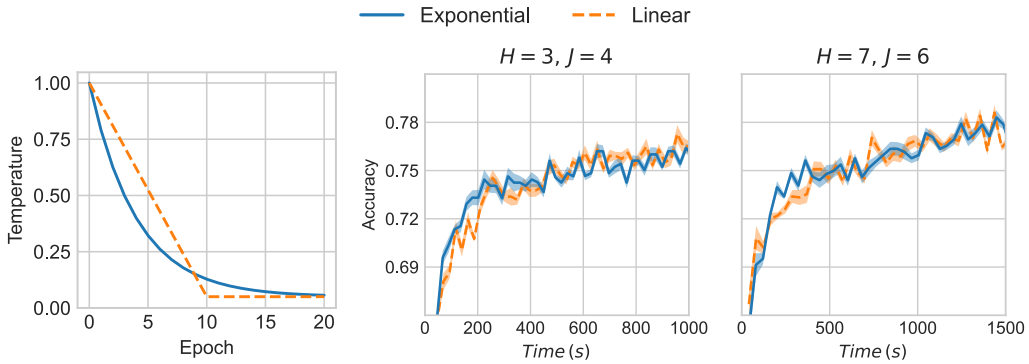


Figure S.6: Convergence Of S-ADVI In Terms Of IWAE Loss For FMNIST

S.3.4 Comparison of Two Annealing Methods



(a) Annealing functions

(b) Performance comparison

Figure S.7: (a) Illustration Of Two Annealing Functions (b) Performance Comparison For Two Annealing Functions Under Different Combinations Of H And J (The shadow denote the standard error)

Annealing is important in the proposed method to ensure most of the combinations of coefficients are explored. In

this section, we evaluate the influences of annealing functions on model performance. We consider two annealing functions in this experiment:

Exponential $\Lambda(c) = \Lambda_1 + (\Lambda_0 - \Lambda_1)e^{-c/\eta}$,

Linear

$$\Lambda(c) = \begin{cases} \Lambda_0 - \frac{\Lambda_0 - \Lambda_1}{\eta'} c & \text{if } c \leq \eta' \\ \Lambda_1 & \text{if } c > \eta' \end{cases}.$$

The parameter η in the exponential annealing function controls the decreasing rate of the temperature, while η' in the linear annealing function determines the decreasing rate and the turning point where the temperature becomes a constant.

We run the same single-column classification task on MNIST as in Section 6.2. The effects of annealing functions are evaluated based on the convergence speed. As in the main paper, we consider two combinations of H and J: ($H = 3$, $J = 4$) and ($H = 7$, $J = 6$), and choose the classification accuracy as the performance metric. For the exponential function, we use $\Lambda_0 = 1$, $\Lambda_1 = 0.05$ and $\eta = 4$. For the linear function, we use $\Lambda_0 = 1$, $\Lambda_1 = 0.05$ and $\eta' = 10$.

The results are shown in Figure S.7. We find that the exponential annealing function used in the main text converges faster than the linear function under both parameter combinations. On the other hand, after the initial epochs, the performance is similar. One potential reason is that the temperature decreases faster with the exponential function at the first few epochs, which helps the model to better approximate the categorical distribution with better parameter estimation.

S.3.5 Imaging Reconstructions

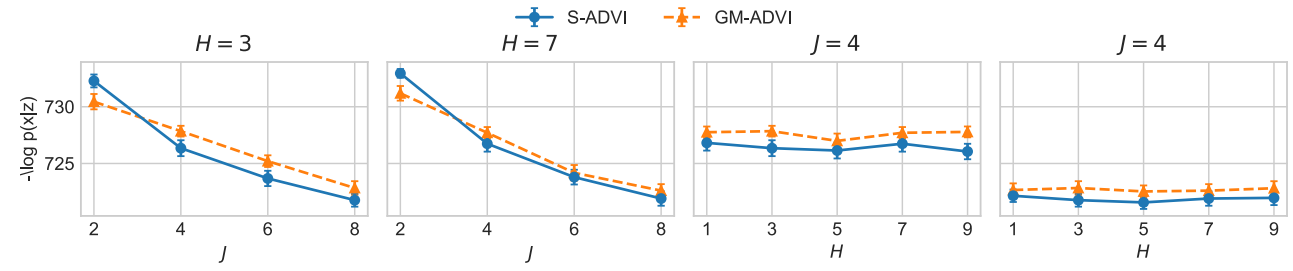
Our previous experiments suggest that the proposed S-ADVI methods can match and outperform other VI-based methods in terms of classification task, even when the data does not contain sufficient information. In this experiment, we compare the S-ADVI and GM-ADVI methods in terms of reconstruction using MNIST (LeCun et al., 1998), FMNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky et al., 2009) datasets.

The network structures of the encoder and decoder are MLP with 2 layers. The numbers of hidden units are 512 and 256 for the encoder, and 256 and 512 for the decoder. We choose $\{0.25, 0.50, 0.75\}$ as the interior knots and use 4 latent variables. We set $T = 10$ for IWAE, use isotropic Gaussian distribution as the prior, and train the model with Adam optimizer with a learning rate of 0.001 decayed by 5% for every epoch. The final output of the decoder includes the means and standard deviations of the pixels. For the CIFAR-10 dataset, we transform the color images to greyscales. We evaluate the reconstruction performance using $-\log p(\boldsymbol{x}|\boldsymbol{z})$, where we assume that $p(\boldsymbol{x}|\boldsymbol{z})$ follows Gaussian distribution.

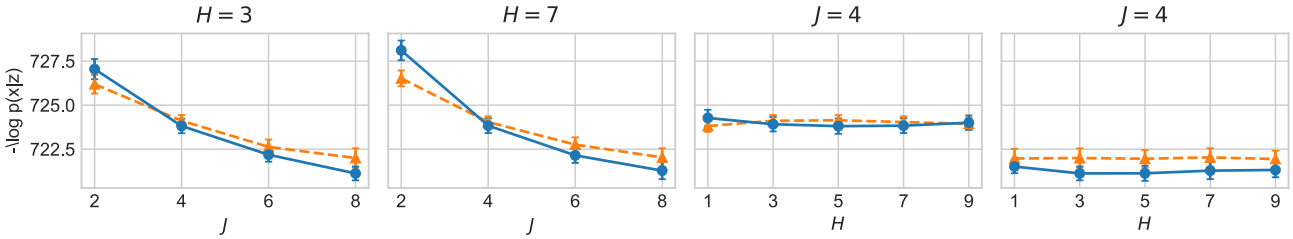
The results are shown in Figure S.8. We find that when the number of latent variables is limited, GM-ADVI outperforms S-ADVI. When J increases, the performance of both models improves, and S-ADVI outperforms GM-ADVI. The number of bases H also influences model performance when H increases from 1 to 3, but further increasing H does not improve model performance much and can lead to overfitting. It is also interesting to note that the advantage of S-ADVI increases for more complicated datasets (e.g., MNIST vs CIFAR-10) with higher J and H. Several facts may contribute to the observation. For example, S-ADVI is capable of capturing bounded and skewed posterior distribution. The pre-specified spline bases can help prevent the model from concentrating on the few most important modes and ignoring those relatively less important modes. The bounded support of the posterior from S-ADVI may mitigate overfitting.

S.3.6 Analysis of the Shape of the Approximated Posterior with VAE

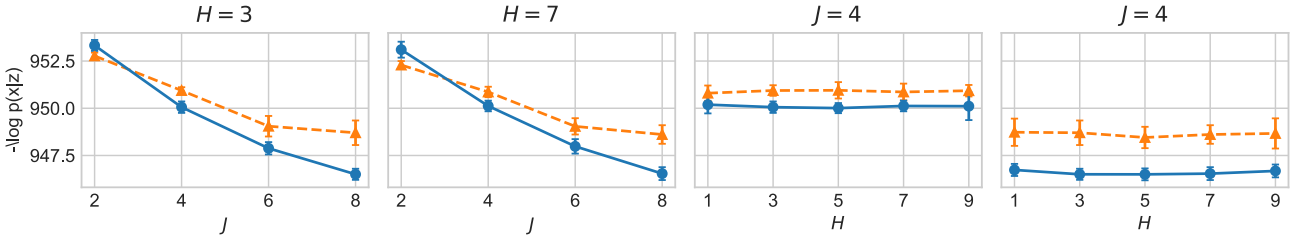
The network structures of the encoder and decoder are MLP with 2 layers. The numbers of hidden units are 512 and 256 for the encoder, and 256 and 512 for the decoder. We use $\beta = 0.05$ penalty on the KL divergence term. We choose $\{0.25, 0.50, 0.75\}$ as the interior knots and use 4 latent variables. We set $T = 10$ for IWAE, use isotropic Gaussian distribution as the prior, and train the model with Adam optimizer with a learning rate of 0.001 decayed by 5% for every epoch.



(a) MNIST



(b) FMNIST



(c) CIFAR-10

Figure S.8: Reconstruction Comparison For Full Image VAE With A Range Of Latent Variables J And Interior Knots H (smaller Is Better). Error Bars Denote The Standard Error.

Examples of the approximated $q_\phi(\epsilon_j|\boldsymbol{x})$ are illustrated in Figure S.9 for digits 0, 2, and 4. We randomly select 100 samples for each digit from the testing set and visualize the approximated posteriors' probability density functions (PDFs) and their average values across all samples. The results show that the proposed method can effectively capture the skewness (e.g., $q_\phi(\epsilon_3|\boldsymbol{x})$) and multimodality (e.g., $q_\phi(\epsilon_1|\boldsymbol{x})$) of the posterior distribution. Furthermore, in general, while labels are not included in the training process, the approximated PDFs of samples of the same digit share similar shapes, whereas samples of different digits can have different shapes. For example, samples of digits 0 and 2 have different PDFs for ϵ_3, and the modes differ for ϵ_4 for samples of digits 0, 2, and 4. These observations imply that the shape of the PDFs of the underlying posteriors can capture the similarities of samples from the same group (samples of the same digit) and the differences of samples from different groups. Figure S.10 shows original samples and S-VAE reconstructions for digit 2, following the same format as Figure 5 in the main text. Similar patterns are observed, where bases with higher coefficients represent larger image regions and capture key image characteristics.

In Figures S.11 and S.12, we visualize the latent variables for digits 1, 3, 5, 6, 7, 8, 9. We find that the shapes of posteriors of certain latent variables can be shared across digits with similar patterns, such as ϵ_1 for digits 3, 5, 8, and 9. We also observe that for some digits (e.g., 5, 6, 8, 9), the shapes of PDF of ϵ are inconsistent within the groups. One possible reason is the existence of pattern variation in the handwritten digits, i.e., samples of the same digit can be further divided into subgroups, where samples from different subgroups can have very different handwritten patterns. As discussed in Section 7, further evaluation and investigation with statistical methods are required to identify and interpret the observed inconsistent PDFs.

We visualize the connections between input samples and the shapes of the approximated $q_\phi(\epsilon_j|\boldsymbol{x})$ for two digits, 6 and 7, in Figure S.13. Based on Figures S.11 and S.12, we focus on the 4th basis (the centermost spline basis) of

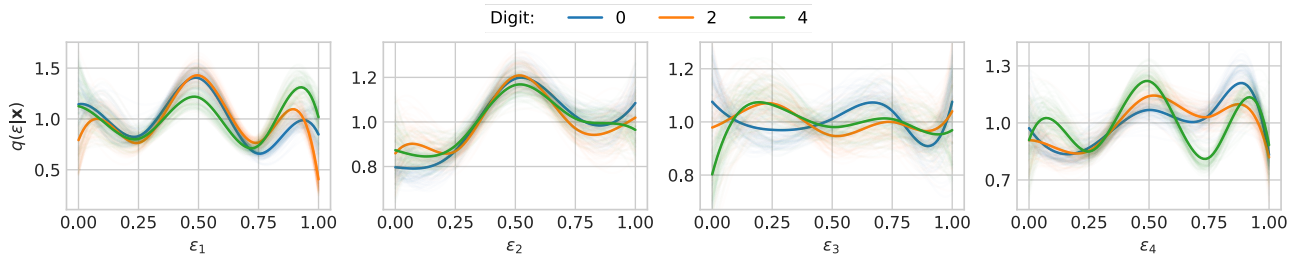


Figure S.9: Illustration Of Shapes Of Latent Variable Distributions For Digits 0, 2, And 4. We Randomly Select 100 Samples For Each Of The Digits. PDFs Of Individual Samples Are In Light Colors, And The Deep Solid Curve Denotes The Averages Of PDF Values

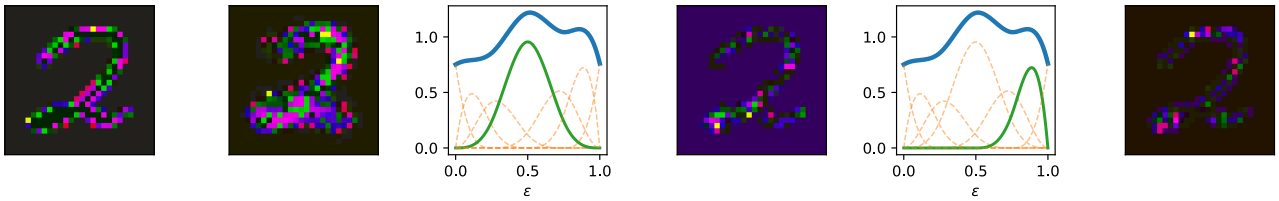
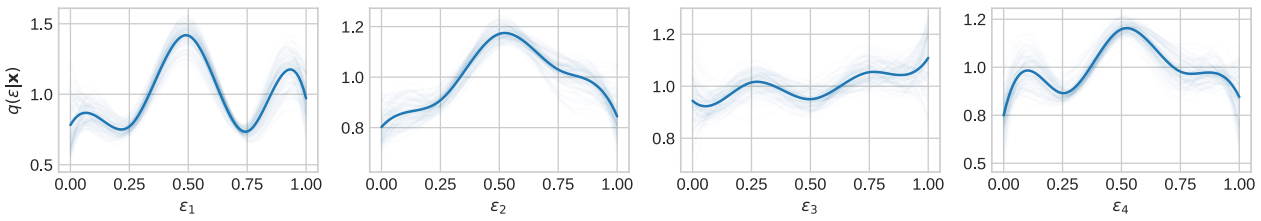
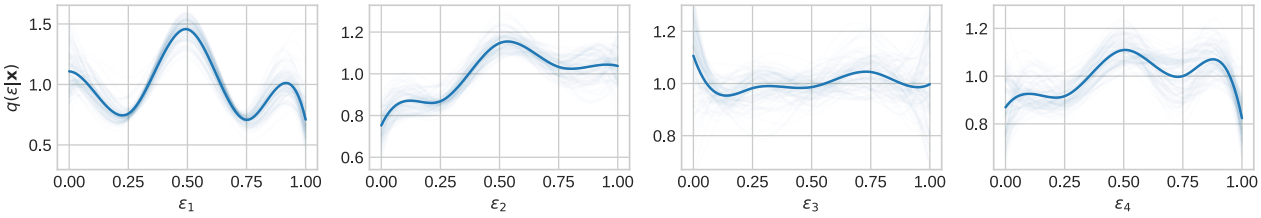


Figure S.10: Analysis Of The Approximated Shape Of Posterior From S-VAE For Digit 2

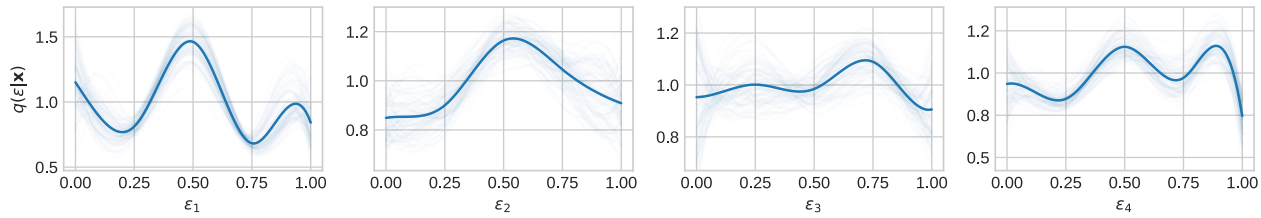
ϵ_1 and the 6th basis of ϵ_4 (the second to the right spline basis). We randomly select three samples for each of the digit. We find that the results are consistent with the examples in the main text for digits 0 and 4. Spline bases with higher weights are usually associated with larger regions in the images. We also observe that the same basis usually points to the same digit region. For example, IG suggests that basis 4 of ϵ_1 is associated with the middle part of digit 6, and basis 6 of ϵ_4 is associated with the lower part of digit 7.



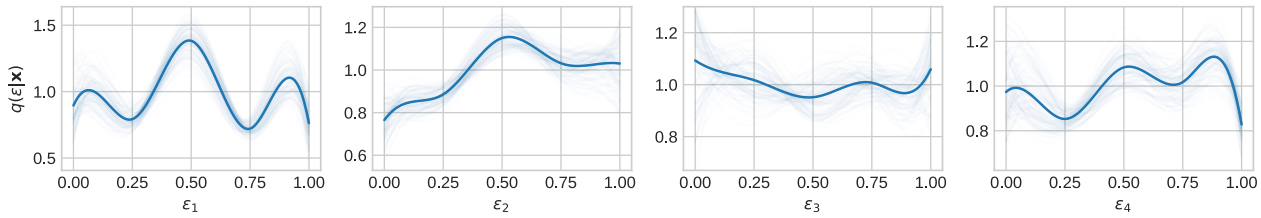
(a) Digit 1



(b) Digit 3

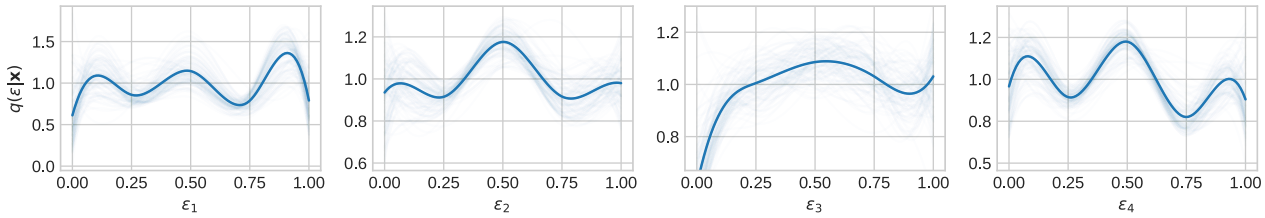


(c) Digit 5

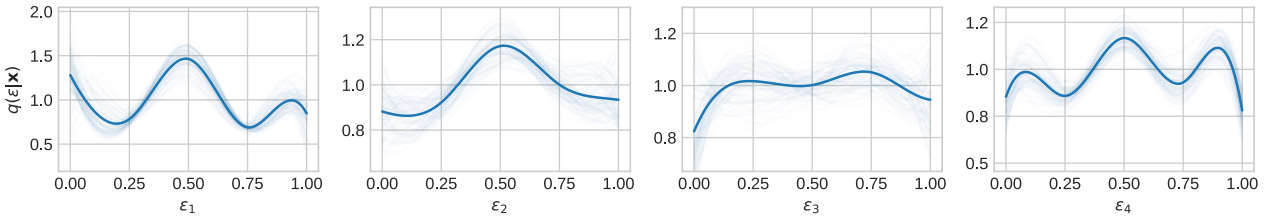


(d) Digit 6

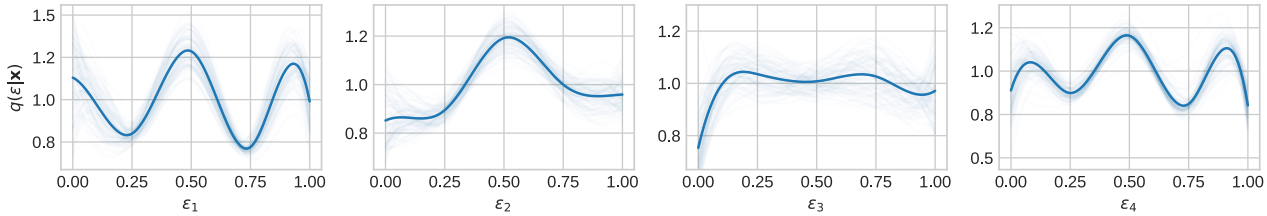
Figure S.11: Visualization Of Latent Distributions For Digits 1, 3, 5, 6



(a) Digit 7

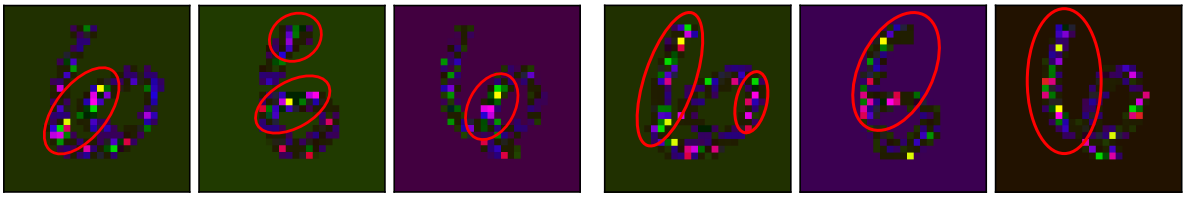


(b) Digit 8



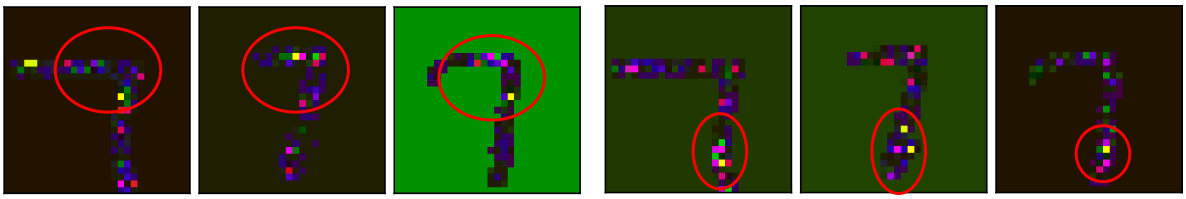
(c) Digit 9

Figure S.12: Visualization Of Latent Distributions For Digits 7, 8, 9



(a) Digit 6, basis 4 of ϵ_1

(b) Digit 6, basis 6 of ϵ_4



(c) Digit 7, basis 4 of ϵ_1

(d) Digit 7, basis 6 of ϵ_4

Figure S.13: Interpretation Of Weighted Spline Bases For Samples Of Digits 6 And 7. Brighter Pixels Correspond To Higher Absolute Attributes. Regions Containing Highly Attributed Pixels Are Highlighted In Red Circles

References

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2016), "Fast and accurate deep network learning by exponential linear units (elus)," in *International Conference on Learning Representations*.

Csiszár, I. and Talata, Z. (2006), "Context tree estimation for not necessarily finite memory processes, via BIC and MDL," *IEEE Transactions on Information theory*, 52, 1007–1016.

Krizhevsky, A., Hinton, G., et al. (2009), "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998), "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86, 2278–2324.

Morningstar, W., Vikram, S., Ham, C., Gallagher, A., and Dillon, J. (2021), "Automatic differentiation variational inference with mixtures," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, vol. 130, pp. 3250–3258.

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

Xiao, H., Rasul, K., and Vollgraf, R. (2017), "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*.