Quantum-Inspired Score-Based Variational Inference

Yuchen Cong
Juntendo University

Zerui Tao
RIKEN-AIP

Chao Li*
RIKEN-AIP

Zhe Sun
Juntendo University

Qibin Zhao
RIKEN-AIP

Abstract

Score-based variational inference (SBVI) provides an alternative to classical Kullback–Leibler (KL)-based variational inference by learning a distribution using only score queries of the target. EigenVI [1] formulates SBVI as an eigenvalue problem, but faces two limitations: the number of variational parameters grows exponentially with the data dimension, and the single-eigenvector formulation cannot represent the full minimum-eigenvalue eigenspace. We propose *QuanVI*, a quantum-inspired SBVI algorithm that addresses both limitations. Rather than selecting a single eigenvector, QuanVI captures a learned subspace through a density-operator formulation. To enable scalable computation, QuanVI employs a matrix-product-operator (MPO) structure parameterized by a quantum tensor network (QTN) with unitary-constrained local tensors and auxiliary ancilla wires. Numerical experiments demonstrate the scalability of QuanVI beyond EigenVI and competitive performance on synthetic distributions and Bayesian posterior-approximation benchmarks, including challenging non-Gaussian targets. The code is available at https://github.com/ChaoLiAtRIKEN/quanvi.

1 Introduction

Variational inference (VI, [2, 3]) approximates a target distribution by finding the closest member of a parameterized family of distributions. Conventional VI methods minimize the Kullback–Leibler (KL) divergence [4] between the variational approximation and the target distribution. In contrast, score-based VI (SBVI) minimizes the Fisher divergence between the two distributions by matching their score functions. Consequently, SBVI is particularly attractive when target score evaluations are available but target density evaluations are computationally expensive or intractable.

Previous studies have investigated SBVI using Gaussian variational families [5, 6]. Recently, EigenVI [1] introduced non-Gaussian variational families based on orthogonal basis expansions. More intriguingly, the work reveals that minimizing Fisher divergence with EigenVI reduces to solving a minimum-eigenvalue problem. Despite its promising performance, EigenVI faces two limitations: first, the number of variational parameters grows *exponentially* with the data dimension, limiting the scalability of EigenVI; second, the single-eigenvector formulation does not account for the possible *multiplicity* of the minimum eigenvalue, potentially limiting the expressive power of the variational family.

To address these limitations, we propose *QuanVI*, a novel SBVI method inspired by quantum computing. Building on EigenVI, QuanVI extends the variational parameters from a single vector to a density operator, allowing the model to capture multiple eigen-directions associated with the minimum eigenvalue. To avoid the exponential growth of parameters, we further represent the

*chao.li@riken.jp

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

density operator as a matrix product operator (MPO) [7] using quantum tensor networks (QTNs). The QTN representation decomposes the high-dimensional density operator into multiplications of low-dimensional local tensors, analogous to quantum circuits. With additional unitary constraints on the local tensors, QuanVI enables tractable probabilistic inference without explicitly computing the partition function during training or inference.

In addition, QuanVI addresses the high query complexity of estimating the loss-induced Hamiltonian M in (3) due to the importance sampling, as required by EigenVI. Under a bounded local dependency assumption, such as a Markov random field with bounded neighborhood size, the Hamiltonian can be represented as a sum of local terms, enabling importance sampling in low-dimensional spaces with reduced query complexity.

Empirically, we evaluate QuanVI on synthetic distributions to examine its scalability and expressiveness, and on Bayesian posterior-approximation benchmarks from POSTERIORDB [8] to assess its practical applicability. We also conduct ablation studies to examine the effects of various hyperparameters. The results demonstrate that QuanVI scales to dimensions beyond the reach of EigenVI while achieving competitive approximation quality on Gaussian, non-Gaussian, and Bayesian posterior-approximation tasks.

2 Preliminaries

In this section, we review EigenVI and the graphical notation of QTNs.

2.1 EigenVI and Its Eigenvalue Formulation

In EigenVI, a variational density $q_\theta : \mathbb{R}^D \to \mathbb{R}_{\geq 0}$ is constructed by an orthogonal function expansion as follows:

$$q_\theta(x) = |\langle \theta, \Phi(x) \rangle|^2, \qquad \Phi(x) = \bigotimes_{d=1}^{D} \phi(x_d), \tag{1}$$

where $\theta \in \mathbb{R}^{K^D}$ denotes a vector corresponding to the variational parameters, $\otimes$ denotes the tensor product, and $\phi : \mathbb{R} \to \mathbb{R}^K$ represents a vector of basis functions, satisfying the orthonormal assumption $\int \phi(t)\phi(t)^\top \, dt = I_K$. If we further assume $\|\theta\|_F = 1$, it can be easily derived that q_θ should be a valid density function, since the square operation in (1) guarantees the non-negativity of the range of $q_\theta(x)$, and normalized θ with orthogonal basis functions makes the integral of $q_\theta(x)$ wrt. x equals one. In the TN community, (1) is also often called continuous-variable Born machine [9].

Suppose that a target density p is unknown but accessible by its score function $\nabla_x \log p(x)$. EigenVI aims to solve the following optimization problem:

$$\min_\theta \frac{1}{2} \int q_\theta(x) \left\| \nabla_x \log p(x) - \nabla_x \log q_\theta(x) \right\|_F^2 dx, \qquad \text{s.t. } \|\theta\|_F = 1. \tag{2}$$

We can see from (2) that EigenVI minimizes the Fisher divergence [10] between p and q_θ.

An important observation established in EigenVI is that solving (2) is equivalent to solving a minimum-eigenvalue problem:

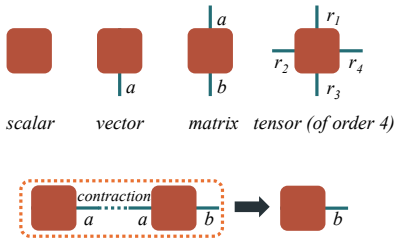
$$\min_\theta \theta^\top M \theta, \qquad \text{s.t. } \|\theta\|_F^2 = 1, \tag{3}$$

where $M \in \mathbb{R}^{K^D \times K^D}$ denotes a loss-induced matrix defined as follows:

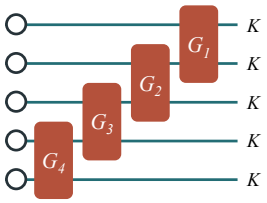
$$M := \int \left(2\dot{\Phi}(x) - \nabla_x \log p(x) \cdot \Phi(x)^\top \right)^\top \left(2\dot{\Phi}(x) - \nabla_x \log p(x) \cdot \Phi(x)^\top \right) dx, \tag{4}$$

where $\dot{\Phi}(x)$ denotes a Jacobian-like form of $\Phi(x)$, explicitly defined in (12). The derivation of M from (2) is provided in Appendix A.1. Hence, an optimal θ should belong to the eigenvector wrt. the minimum eigenvalue of M. Any eigensolver can be applied to tackle the task.

Limitations of EigenVI. Directly solving (3) faces three main challenges. First, EigenVI selects a single eigenvector associated with the minimum eigenvalue of M, overlooking the possibility that this eigenvalue has multiplicity greater than one. This restriction may limit the expressiveness of



(a) Basics of TN graphical notation. Each node represents a tensor, and each edge represents an index (mode) of the tensor. The label on an edge indicates the dimension of the corresponding mode.



(b) A simple QTN example. A QTN is a circuit-like TN architecture with additional unitary constraints on local tensors. In this example, circles represent initial-state tensors (vector) of dimension K, while rectangles denote local tensors of dimensions $K \times K \times K \times K$.

Figure 1: Graphical illustration of tensor-network notation and representations. (a) Basic tensor objects and tensor contraction. (b) A simple QTN example.

the variational family. Second, the dimension of θ grows exponentially with the data dimension D, making the eigenvalue problem in (3) computationally intractable in high dimensions. Third, EigenVI estimates M using importance sampling over the global D-dimensional space, which can require exponentially many target score queries as D increases. These limitations hinder the practical application of EigenVI to high-dimensional problems, despite its promising performance on non-Gaussian distributions [1].

2.2 Quantum Tensor Networks and Graphical Notation

To overcome the limitations of EigenVI, QuanVI builds on tensor networks, which represent high-dimensional tensors through contractions of smaller local tensors. Figure 1a illustrates the graphical notation for tensor contractions. Each node represents a tensor, each edge represents an index, and connected edges indicate contraction. The dimension of a shared index is called the bond dimension. We use this notation throughout the paper to illustrate tensor operations in QuanVI.

We focus on quantum tensor networks (QTNs), a class of tensor networks inspired by quantum circuits, as illustrated in Figure 1b. In our QTN architecture, local tensors consist of *initial-state tensors* (shown as circles) and unitary *gate tensors* (shown as rectangles). Tensor contractions are represented by *wires*, as in quantum circuits. The unitary constraints preserve normalization, eliminating the need for an explicit partition function when combined with orthogonal basis functions. Moreover, the structured contractions in our QTN architecture enable efficient evaluation of marginal probabilities.

3 Method

We now introduce QuanVI, which builds on three key ideas. First, we extend the variational vector θ in EigenVI to a density operator. Second, we use a QTN to represent the density operator efficiently. Third, we exploit local dependencies to express the high-dimensional loss-induced matrix M as a sum of local terms. The density-operator formulation allows multiple eigen-directions associated with the minimum eigenvalue, the QTN parameterization avoids exponentially many explicit variational parameters, and the local decomposition reduces the query complexity of estimating M.

3.1 Extending EigenVI with Density Operators

Recall the EigenVI optimization problem in (3). We generalize this problem to

$$\min_{\rho} \quad \mathrm{tr}(\rho M), \text{ s.t.} \qquad \rho \succeq 0, \quad \mathrm{tr}(\rho) = 1, \quad \mathrm{rank}(\rho) \leq r, \tag{5}$$

where $1 \leq r \leq K^D$. When $r = 1$, we can write $\rho = \theta\theta^\top$ with $\|\theta\|_2 = 1$, recovering the original EigenVI formulation in (3). For general r, the variational parameter becomes a positive semidefinite

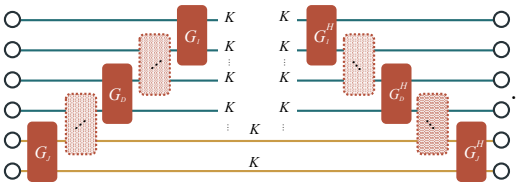
(PSD) matrix $\rho \in \mathbb{R}^{K^D \times K^D}$ with unit trace. Such a matrix is called a *density operator* in quantum information theory. Thus, EigenVI corresponds to the rank-one case of (5).

The density-operator formulation allows multiple eigen-directions to contribute to the variational distribution. In particular, when the minimum eigenvalue of M has multiplicity r, any density operator supported on the corresponding eigenspace is optimal for (5). This includes mixed states that combine all r orthonormal eigenvectors, extending the single-eigenvector representation of EigenVI.

We remark that our recent work [11] explains a connection between (5) and Fisher divergence minimization. Specifically, the minimum loss of (5) is upper bounded by that of EigenVI and lower bounded by $D_F(q_{\rho^\star} \| p)$, where $\rho^\star$ is an optimal solution to (5) and $q_{\rho^\star}(x) := \Phi(x)^\top \rho^\star \Phi(x)$ is the corresponding variational distribution. It implies that minimizing (5) is *not* equivalent to directly minimizing the Fisher divergence, but remains a reasonable surrogate loss, as its minimum is sandwiched between the Fisher divergence of the resulting variational distribution and the minimum Fisher loss of EigenVI.

3.2 Parameterizing the Density Operator ρ with QTNs

In QuanVI, we parameterize the density operator ρ in (5) using a specific QTN architecture. The corresponding graphical representation is given by

$\rho =$  . (6)

As shown in (6), the QTN consists of learnable gate tensors $G_j \in \mathbb{C}^{K^L \times K^L}$, for $j = 1, \ldots, J$, and fixed initial-state tensors. Here, K denotes the wire bond dimension, determined by the *basis size*, while L, referred to as the *local window size*, denotes the number of wires acted upon by each gate tensor. Each entry of the initial-state tensors is set to $e^{i\pi/4}/\sqrt{K}$, yielding a normalized uniform superposition state.

Unlike general QTNs, the model in (6) has three distinctive features. First, the gate tensors are paired with their complex conjugates in a mirrored structure, ensuring that the resulting density operator is PSD. Second, the gate tensors are arranged in a staircase pattern, where each gate shares one wire with its neighbors. This structure is known to be equivalent to an MPO representation [12]. As we show later, the staircase structure also enables efficient computation during both training and inference. Third, the wires in the QTN are divided into two types, distinguished by color: variable wires and ancilla wires. The variable wires correspond to the indices of the density operator ρ, while the ancilla wires enable the representation of mixed states according to the purification theorem [13]. The number of variable wires is equal to the data dimension D, while the number of ancilla wires is a hyperparameter that controls the maximum rank of the density operator.

3.3 Localizing the Loss-Induced Matrix M

In EigenVI, the integral in (4) is estimated by importance sampling using a D-dimensional uniform proposal. However, the estimation variance can grow exponentially with D, requiring exponentially more samples to maintain the same accuracy. To address this challenge, we exploit the local dependency structure of the target distribution, inspired by local interaction models in statistical physics, such as the Ising model. Under this assumption, we show that M can be expressed as a sum of local terms, each involving a bounded number of variables independent of D. This decomposition allows importance sampling to be applied to each local term separately.

Specifically, let $s_d(x) := \partial_{x_d} \log p(x)$ denote the d-th component of the target score. We assume a nearest-neighbor dependency structure, where $s_d(x)$ depends only on the variables in $\mathcal{N}_d := \{d-1, d, d+1\} \cap [D]$. Let $\Phi_{\mathcal{N}_d}(x_{\mathcal{N}_d})$ denote the tensor-product basis restricted to $\mathcal{N}_d$. We define

the local matrix

$$M^{(d)} := \int \Big(2\partial_{x_d}\Phi_{\mathcal{N}_d}(x_{\mathcal{N}_d}) - s_d(x_{\mathcal{N}_d})\Phi_{\mathcal{N}_d}(x_{\mathcal{N}_d})\Big) \cdot \Big(2\partial_{x_d}\Phi_{\mathcal{N}_d}(x_{\mathcal{N}_d}) - s_d(x_{\mathcal{N}_d})\Phi_{\mathcal{N}_d}(x_{\mathcal{N}_d})\Big)^\top dx_{\mathcal{N}_d}. \tag{7}$$

The loss-induced matrix M can then be constructed as

$$M = M^{(1)} \otimes I_K^{\otimes(D-2)} + \sum_{d=2}^{D-1} I_K^{\otimes(d-2)} \otimes M^{(d)} \otimes I_K^{\otimes(D-d-1)} + I_K^{\otimes(D-2)} \otimes M^{(D)}, \tag{8}$$

where I_K denotes the $K \times K$ identity matrix. Each local matrix $M^{(d)}$ has dimension at most $K^3 \times K^3$, independent of D. Thus, each $M^{(d)}$ can be estimated separately through importance sampling over at most three variables, avoiding integration over the full D-dimensional space.

Substituting (8) into the loss in (5), we obtain

$$\mathrm{tr}(\rho M) = \sum_{d=1}^{D} \mathrm{tr}\left(\rho_{\mathcal{N}_d} M^{(d)}\right), \tag{9}$$

where $\rho_{\mathcal{N}_d} := \mathrm{tr}_{[D]\setminus\mathcal{N}_d}(\rho)$ denotes the reduced density operator obtained by taking the partial trace [13] over variables outside $\mathcal{N}_d$. Consequently, each local loss term involves only matrices of dimension at most $K^3 \times K^3$. Figure 2 illustrates how to compute two local loss terms in (9) using tensor-network contractions. The QTN structure enables efficient computation of these reduced density operators without explicitly constructing the full density operator ρ.

We remark that the nearest-neighbor assumption does not necessarily hold for arbitrary target distributions. In general, the construction in (8) provides an approximation to the original loss-induced matrix defined in (4). However, as established in [11], this construction is exact when the target distribution is a path-graph Markov random field.

4 Algorithm

In this section, we present the algorithms for efficient training and inference with QuanVI.

4.1 Training

For each d, let $\widetilde{M}^{(d)}$ denote an estimate of $M^{(d)}$ obtained through importance sampling. Following (9), we define the training loss of QuanVI as $\mathcal{L}(\{G_j\}_{j=1}^J) := \sum_{d=1}^D \mathrm{tr}\left(\rho_{\mathcal{N}_d}\widetilde{M}^{(d)}\right)$, where $\rho_{\mathcal{N}_d}$ is parameterized by the gate tensors $\{G_j\}_{j=1}^J$.

Training proceeds as follows. First, the gate tensors are initialized using Haar-Gaussian-distributed unitary matrices. Next, the loss is evaluated through tensor-network contractions, proceeding wire by wire from top to bottom. Finally, the gate tensors are updated by gradient descent on the Stiefel manifold until convergence.

This training procedure offers two advantages. First, for fixed K and L, the contractions require only polynomial computational time and memory in D. Second, Stiefel manifold optimization preserves the unitary constraints of the gate tensors, ensuring that ρ remains a valid density operator without explicit normalization. Implementation details are provided in our Python code, with a complexity analysis in Appendix B.

4.2 Inference

We consider four inference tasks for QuanVI: density queries, marginalization, conditionalization, and sampling. All these tasks can be performed efficiently through tensor-network contractions.

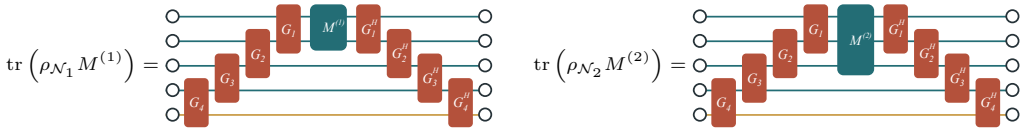
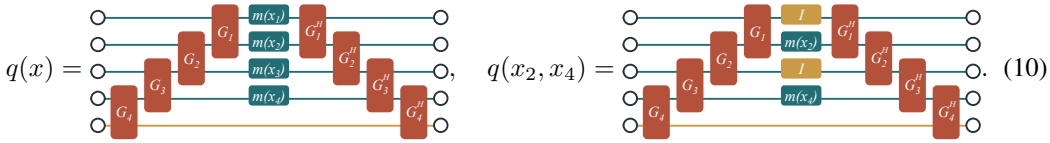


Figure 2: Illustration of tensor-network contractions for $\mathrm{tr}(\rho_{\mathcal{N}_1} M^{(1)})$ and $\mathrm{tr}(\rho_{\mathcal{N}_2} M^{(2)})$ in (9), with $D = 4$, $L = 2$, and one ancilla wire.

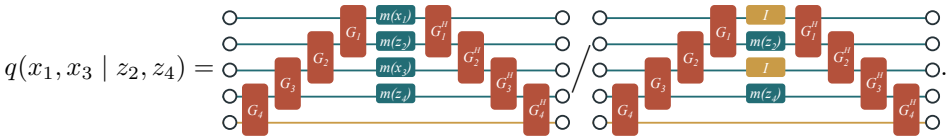
Density queries and marginalization. After training, density queries and marginalization can be performed through tensor-network contractions, based on $q(x) = \Phi(x)^\top \rho \Phi(x)$. As an illustrative example, consider $x = (x_1, x_2, x_3, x_4)$ with $D = 4$, $L = 2$, and one ancilla wire. Let $m(x_d) := \phi(x_d)\phi(x_d)^\top$ for each $d \in [D]$. The joint density $q(x)$ and the marginal density $q(x_2, x_4)$, obtained by integrating out x_1 and x_3, can be computed as follows:



(10)

Here, marginalization is performed by replacing $m(x_d)$ with the identity matrix I for each marginalized variable x_d, since the orthonormality of the basis functions ensures that $\int m(x_d)\, dx_d = \int \phi(x_d)\phi(x_d)^\top dx_d = I$.

Conditional density. Conditional densities can be calculated as ratios of tensor-network contractions. Under the same setting with $D = 4$, $L = 2$, and one ancilla wire, consider the conditional density of (x_1, x_3) given $(x_2, x_4) = (z_2, z_4)$: $q(x_1, x_3 \mid z_2, z_4) = \frac{q(x_1, z_2, x_3, z_4)}{q(z_2, z_4)}$. The numerator is obtained by fixing the local matrices at (x_1, z_2, x_3, z_4), while the denominator is obtained by replacing the matrices at sites 1 and 3 with identity matrices. The corresponding tensor-network contractions are illustrated as follows:



(11)

Sampling. Sampling follows the probabilistic chain rule: $q(x_1, \ldots, x_D) = q(x_1) \prod_{d=2}^{D} q(x_d \mid x_{<d})$, where $x_{<d} := (x_1, \ldots, x_{d-1})$. Following EigenVI [1], QuanVI generates samples sequentially using this factorization. At each step d, the previously sampled variables $x_{<d}$ are fixed, while the future variables $x_{>d}$ are marginalized out by replacing their local matrices with identity matrices. The resulting tensor-network contractions yield the one-dimensional conditional density $q(x_d \mid x_{<d}) = \frac{q(x_{\leq d})}{q(x_{<d})}$. We then numerically integrate this density to obtain its conditional cumulative distribution function (CDF), draw $u \sim \mathrm{Uniform}(0, 1)$, and obtain x_d by inverting the CDF. Repeating this procedure for $d = 1, \ldots, D$ produces a sample from q.

Density queries, marginalization, conditional density evaluation, and sampling all rely on the same tensor-network contraction framework. A detailed complexity analysis is provided in Appendix B.

5 Experiments

We evaluate QuanVI through three sets of experiments. First, we assess its performance and scalability on synthetic distributions across different dimensions and target characteristics. Second, we evaluate posterior approximation on PosteriorDB [8]. Finally, we conduct ablation studies on the window size L, basis size K, and number of ancilla wires E.

Table 1: Forward KL divergence (mean $\pm$ standard deviation over five runs; lower is better). The best mean in each row is highlighted in **bold**. For QuanVI, E denotes the number of ancilla wires. OOM indicates out-of-memory, and "–" indicates unstable runs.

D	Target	Baselines					QuanVI	
		GSM [5]	BAM [6]	ADVI [14]	EigenVI [1]	MoG [15]	$E=0$	$E=2$
5	Gaussian	0.9520 $\pm$1.3282	$\mathbf{\leq 1\times10^{-2}}$	$\mathbf{\leq 1\times10^{-2}}$	$\mathbf{\leq 1\times10^{-2}}$	0.0152 $\pm$0.0009	$\mathbf{\leq 1\times10^{-2}}$	$\mathbf{\leq 1\times10^{-2}}$
	X-shape	2.2494 $\pm$0.4961	3.0904 $\pm$0.2373	1.6816 $\pm$0.0716	0.4007 $\pm$0.0743	1.3039 $\pm$0.0269	**0.2613 $\pm$0.0035**	0.2661 $\pm$0.0054
	GMM3	1.6469 $\pm$0.5895	2.4417 $\pm$0.2273	1.1967 $\pm$0.0313	0.4454 $\pm$0.0613	0.4436 $\pm$0.0617	0.1608 $\pm$0.0112	**0.1598 $\pm$0.0042**
	Funnel	3.3535 $\pm$0.5241	4.8814 $\pm$0.2082	2.8012 $\pm$0.0653	1.3435 $\pm$0.1065	**0.8183 $\pm$0.0742**	4.3944 $\pm$1.8299	2.6412 $\pm$0.4579
	Ring	30.7895 $\pm$8.8223	39.8048 $\pm$2.5664	2.1094 $\pm$0.0221	1.7526 $\pm$0.1454	2.0528 $\pm$0.2233	**0.8861 $\pm$0.0048**	0.8913 $\pm$0.0144
10	Gaussian	$\mathbf{\leq 1\times10^{-2}}$	$\mathbf{\leq 1\times10^{-2}}$	$\mathbf{\leq 1\times10^{-2}}$	OOM	0.0814 $\pm$0.0024	$\mathbf{\leq 1\times10^{-2}}$	$\mathbf{\leq 1\times10^{-2}}$
	X-shape	16.0632 $\pm$5.3750	24.9337 $\pm$1.6719	5.8356 $\pm$0.0990	OOM	6.5115 $\pm$0.1666	3.2554 $\pm$0.0670	**3.1928 $\pm$0.0655**
	GMM3	7.0765 $\pm$1.1840	24.0342 $\pm$1.3684	5.4414 $\pm$0.0758	OOM	5.1997 $\pm$0.3525	**3.1225 $\pm$0.0504**	3.1551 $\pm$0.0551
	Funnel	–	29.5528 $\pm$4.4531	7.0246 $\pm$0.0623	OOM	**5.0368 $\pm$0.3313**	9.0460 $\pm$0.8675	8.8288 $\pm$1.0972
	Ring	9.1494 $\pm$1.9204	64.7096 $\pm$1.8813	6.3256 $\pm$0.1146	OOM	7.7603 $\pm$0.4014	3.8400 $\pm$0.0213	**3.8302 $\pm$0.0271**
20	Gaussian	$\mathbf{\leq 1\times10^{-2}}$	$\mathbf{\leq 1\times10^{-2}}$	0.0124 $\pm$0.0035	OOM	0.2884 $\pm$0.0088	$\mathbf{\leq 1\times10^{-2}}$	$\mathbf{\leq 1\times10^{-2}}$
	X-shape	14.3388 $\pm$2.1860	38.0141 $\pm$2.2533	8.4358 $\pm$0.1412	OOM	9.9434 $\pm$0.2302	**8.1714 $\pm$0.8125**	8.9266 $\pm$0.5087
	GMM3	11.4537 $\pm$0.9701	38.2367 $\pm$0.5113	8.1050 $\pm$0.1880	OOM	8.4863 $\pm$0.4095	**7.4542 $\pm$1.4717**	7.7778 $\pm$1.1659
	Funnel	–	41.6810 $\pm$3.9465	9.8232 $\pm$0.1614	OOM	**7.6320 $\pm$0.2791**	13.1389 $\pm$0.9869	13.4216 $\pm$0.9977
	Ring	13.5824 $\pm$0.2441	77.6129 $\pm$5.2570	**8.9786 $\pm$0.1421**	OOM	10.6938 $\pm$0.3243	9.1621 $\pm$1.5400	9.0459 $\pm$1.0407
100	Gaussian	$\mathbf{\leq 1\times10^{-2}}$	$\mathbf{\leq 1\times10^{-2}}$	0.1308 $\pm$0.0020	OOM	2.2290 $\pm$0.0119	$\mathbf{\leq 1\times10^{-2}}$	$\mathbf{\leq 1\times10^{-2}}$
	X-shape	75.3810 $\pm$2.1274	186.0990 $\pm$4.2477	47.2649 $\pm$0.2927	OOM	72.7630 $\pm$6.0097	42.2971 $\pm$1.2528	**40.5473 $\pm$4.0462**
	GMM3	73.5309 $\pm$3.1276	187.8098 $\pm$4.1833	47.1609 $\pm$0.2688	OOM	68.3951 $\pm$7.4535	**43.0097 $\pm$2.3348**	43.0798 $\pm$2.9883
	Funnel	85.7464 $\pm$3.1931	188.2434 $\pm$3.0382	48.8944 $\pm$0.2271	OOM	57.0316 $\pm$0.4351	**46.4010 $\pm$3.1020**	47.0005 $\pm$3.8384
	Ring	73.7223 $\pm$2.1166	224.9920 $\pm$4.1197	48.1236 $\pm$0.2075	OOM	76.9241 $\pm$2.1116	43.7935 $\pm$1.0249	**43.1974 $\pm$1.5993**

5.1 Synthetic target distributions

Target distributions and evaluation metric. We consider five synthetic target distributions: Gaussian, X-shape, GMM3, Funnel, and Ring, illustrated in Figure 3 of the Appendix. The non-Gaussian targets are extended from two to D dimensions through nonlinear Markov-chain augmentation, while the Gaussian target is constructed directly in D dimensions with local correlations. Details are provided in Appendix C.2.2. Following EigenVI, we evaluate approximation quality using the forward KL divergence $\mathrm{KL}(p\|q)$, estimated with 10^4 Monte Carlo samples drawn from p.

Baselines. We compare QuanVI with ADVI [14], BAM [6], GSM [5], MoG [15], and EigenVI [1]. For QuanVI, we consider $E=0$ and $E=2$, corresponding to models without and with ancilla wires, respectively. EigenVI is evaluated only for $D\leq 5$, as its global $K^D\times K^D$ eigendecomposition becomes computationally infeasible in higher dimensions. We also evaluate normalizing flows, but report their results separately in Appendix C.2.4 due to unstable performance across targets. Hyperparameter settings are provided in Appendix C.2.3.

Discussion. Table 1 shows that QuanVI achieves competitive accuracy with EigenVI in low dimensions while scaling to higher dimensions where EigenVI becomes computationally infeasible. At $D=5$, QuanVI matches or outperforms EigenVI on most targets. In contrast, EigenVI cannot scale to higher dimensions due to its global $K^D\times K^D$ eigendecomposition.

On Gaussian targets, QuanVI consistently achieves near-zero KL divergence across all tested dimensions, demonstrating its effectiveness on distributions that satisfy the local dependency assumption. For non-Gaussian targets, QuanVI achieves the best or near-best results on X-shape, GMM3, and Ring in most settings. The main exception is Funnel, where the default configuration is less competitive. As shown in Section 5.3, increasing the local window size substantially improves approximation accuracy on Funnel, at the cost of higher computational complexity. This illustrates the trade-off between expressiveness and computational efficiency.

Figure 4 compares samples generated by different methods, projected onto the first two dimensions of the $D=20$ X-shape target. While GSM and the mixture-based baselines fail to recover the characteristic X-shaped geometry, QuanVI more closely reproduces the target distribution.

Table 2: Bayesian posterior approximation results measured by forward Fisher divergence (mean $\pm$ standard deviation over four runs; lower is better). Best mean values are highlighted in **bold**. For QuanVI, E denotes the number of ancilla wires. `bball` denotes `hmm_bball_0`, and "–" indicates unstable runs.

Model	D	GSM	BAM	ADVI	MoG	NF [16]	EigenVI	QuanVI ($E=0$)	QuanVI ($E=2$)
`gpregr`	3	1.2107 $\pm$0.0345	1.1962 $\pm$0.0293	1.1509 $\pm$0.0177	7.6146 $\pm$3.9904	3.3739 $\pm$0.4973	**0.0828 $\pm$0.0681**	0.2180 $\pm$0.0463	0.2330 $\pm$0.0369
`hmm`	4	148.1225 $\pm$2.5882	139.7349 $\pm$3.9980	168.9795 $\pm$32.6976	–	–	**4.0295 $\pm$1.5069**	5.0783 $\pm$0.6214	4.8047 $\pm$0.4491
`bball`	6	749.1771 $\pm$2.5298	19.9691 $\pm$0.6849	746.7151 $\pm$2.0978	–	–	18.5057 $\pm$3.0322	13.7126 $\pm$0.5198	**13.4460 $\pm$0.3597**

Table 3: Effect of local window size L on forward KL divergence (mean $\pm$ standard deviation over five runs; lower is better). The best mean in each row is highlighted in **bold**. "–" indicates unavailable results.

D	Target	$L=1$	$L=2$	$L=3$	$L=4$	$L=5$
5	X-shape	6.773 $\pm$0.040	0.265 $\pm$0.007	0.248 $\pm$0.005	0.255 $\pm$0.008	**0.246 $\pm$0.012**
	GMM3	9.266 $\pm$4.613	0.160 $\pm$0.004	0.156 $\pm$0.005	**0.155 $\pm$0.007**	**0.155 $\pm$0.005**
	Funnel	9.299 $\pm$2.148	2.838 $\pm$1.262	1.606 $\pm$0.212	**1.388 $\pm$0.093**	1.481 $\pm$0.265
	Ring	11.050 $\pm$3.082	**0.879 $\pm$0.018**	0.891 $\pm$0.025	0.919 $\pm$0.027	0.893 $\pm$0.024
10	X-shape	16.203 $\pm$1.615	3.212 $\pm$0.069	2.956 $\pm$0.053	**2.859 $\pm$0.053**	2.869 $\pm$0.060
	GMM3	19.865 $\pm$5.100	3.146 $\pm$0.065	2.905 $\pm$0.049	2.734 $\pm$0.068	**2.728 $\pm$0.097**
	Funnel	18.983 $\pm$3.115	7.929 $\pm$1.195	6.367 $\pm$0.435	4.383 $\pm$0.527	**2.812 $\pm$0.050**
	Ring	20.669 $\pm$3.089	3.822 $\pm$0.047	3.580 $\pm$0.074	3.450 $\pm$0.060	**3.343 $\pm$0.064**
20	X-shape	36.863 $\pm$2.211	7.914 $\pm$1.599	7.709 $\pm$0.350	**7.636 $\pm$0.290**	–
	GMM3	35.936 $\pm$2.577	8.331 $\pm$0.415	7.740 $\pm$0.141	**7.543 $\pm$0.137**	–
	Funnel	38.551 $\pm$5.300	13.766 $\pm$0.583	11.773 $\pm$0.185	**7.007 $\pm$0.234**	–
	Ring	42.651 $\pm$4.803	9.159 $\pm$0.646	8.369 $\pm$0.364	**7.942 $\pm$0.140**	–

5.2 Bayesian posterior approximation

Next, we evaluate QuanVI on Bayesian posterior approximation tasks from POSTERIORDB [8]. Since the normalized target densities are unavailable, we follow EigenVI and evaluate approximation quality using the forward Fisher divergence, estimated from reference samples provided by the dataset.

Table 2 presents the numerical results. Across three posterior benchmarks, QuanVI achieves accuracy comparable to EigenVI while substantially outperforming the other baselines. In particular, QuanVI achieves the lowest Fisher divergence on `hmm_bball_0`. Figure 5 compares the one-dimensional marginals of QuanVI and EigenVI against reference samples on this task, demonstrating that QuanVI more accurately captures the non-Gaussian structure of the posterior.

5.3 Ablation study

We further conduct ablation studies to examine how key design choices affect approximation quality and the trade-off between expressiveness and computational cost.

Local window size L. Recall that L denotes the local window size, or the number of wires acted upon by each gate tensor. A larger L allows each gate to capture more complex dependencies, but also increases the number of parameters. Since each gate is represented by a $K^L \times K^L$ matrix, its parameter complexity scales as $O(K^{2L})$. Importantly, this exponential dependence is on the local window size L, rather than the full dimension D. Thus, keeping L small allows QuanVI to scale to high-dimensional distributions with manageable computational costs.

We study this trade-off across all non-Gaussian targets, varying $L \in \{1, \ldots, 5\}$ for $D \in \{5, 10\}$ and $L \in \{1, \ldots, 4\}$ for $D = 20$. Table 3 reports the results.

Across all targets, $L = 1$ yields substantially worse approximation quality, highlighting the importance of modeling dependencies between variables. Increasing L to 2 already brings substantial

Table 4: Effect of basis size K on forward KL divergence (mean $\pm$ standard deviation over five runs; lower is better). The best mean among the displayed settings in each row is highlighted in **bold**. Complete results are provided in Table 7.

D	Target	$K=1$	$K=3$	$K=5$	$K=7$	$K=9$
5	X-shape	2.024 ±0.010	0.863 ±0.023	0.265 ±0.007	0.072 ±0.004	**0.024** ±0.003
	GMM3	2.198 ±0.029	0.846 ±0.029	0.160 ±0.004	0.049 ±0.002	**0.020** ±0.001
	Funnel	2.141 ±0.033	**1.593** ±0.117	2.838 ±1.262	3.488 ±0.317	4.247 ±0.154
	Ring	4.700 ±0.017	1.595 ±0.028	0.879 ±0.018	0.289 ±0.004	**0.276** ±0.005
10	X-shape	4.831 ±0.029	4.399 ±0.106	3.212 ±0.069	1.865 ±0.029	**0.787** ±0.037
	GMM3	4.984 ±0.032	4.486 ±0.053	3.146 ±0.065	1.772 ±0.063	**0.823** ±0.078
	Funnel	**4.934** ±0.037	5.220 ±0.134	7.929 ±1.195	9.420 ±0.761	11.118 ±0.511
	Ring	7.488 ±0.035	5.144 ±0.029	3.822 ±0.047	2.056 ±0.050	**1.062** ±0.045
20	X-shape	10.268 ±0.088	9.515 ±0.209	7.914 ±1.599	**5.289** ±1.775	5.345 ±0.896
	GMM3	10.407 ±0.039	10.067 ±0.752	8.331 ±0.415	6.399 ±1.763	**4.801** ±1.391
	Funnel	**10.331** ±0.073	10.671 ±0.397	13.766 ±0.583	14.322 ±0.513	15.645 ±1.036
	Ring	12.870 ±0.052	10.416 ±0.190	9.159 ±0.646	**5.483** ±1.505	5.543 ±1.176

Table 5: Effect of the number of ancilla wires E on forward KL divergence (mean $\pm$ standard deviation over five runs; lower is better). The best mean in each row is highlighted in **bold**. Complete results are provided in Table 8. "–" indicates untested settings.

D	Target	$E=0$	$E=2$	$E=4$	$E=6$	$E=8$	$E=10$
5	X-shape	0.436 ±0.162	**0.388** ±0.040	0.396 ±0.038	–	–	–
	GMM3	**0.173** ±0.023	0.202 ±0.033	0.203 ±0.014	–	–	–
	Funnel	**3.131** ±0.553	3.763 ±0.782	3.193 ±0.342	–	–	–
	Ring	1.054 ±0.143	1.065 ±0.204	**0.927** ±0.035	–	–	–
10	X-shape	**3.231** ±0.224	3.253 ±0.188	3.339 ±0.250	3.277 ±0.241	3.385 ±0.177	3.366 ±0.248
	GMM3	3.405 ±0.343	3.216 ±0.099	3.250 ±0.125	3.340 ±0.324	**3.091** ±0.166	3.234 ±0.136
	Funnel	9.052 ±1.045	8.772 ±0.581	8.603 ±0.808	9.063 ±0.322	9.084 ±0.486	**7.870** ±0.844
	Ring	3.948 ±0.187	3.956 ±0.308	**3.829** ±0.191	3.966 ±0.099	3.953 ±0.249	4.052 ±0.126
20	X-shape	8.294 ±1.607	7.947 ±1.502	8.093 ±0.189	8.519 ±0.607	**7.306** ±1.071	8.491 ±1.104
	GMM3	8.021 ±0.862	7.916 ±1.518	8.552 ±0.452	8.646 ±0.715	**7.648** ±1.596	8.154 ±1.810
	Funnel	13.143 ±1.655	12.637 ±1.569	13.661 ±0.550	13.336 ±1.518	**12.345** ±1.309	13.843 ±1.085
	Ring	8.080 ±1.967	8.285 ±1.318	7.829 ±1.846	7.532 ±1.904	8.062 ±1.856	**7.407** ±1.978

improvements, while larger windows provide further gains, especially in higher dimensions. The improvement is particularly pronounced for Funnel, whose conditional-scale dependence is difficult to capture with small windows. These results highlight the trade-off between expressiveness and computational efficiency.

Number of basis functions K. Recall that K denotes the number of basis functions, which determines the dimension of the local feature space. A larger K provides a richer basis for approximating the target distribution, but also increases the number of parameters in each gate tensor, which scales as $O(K^{2L})$. We evaluate $K \in \{1, \ldots, 10\}$ across all non-Gaussian targets at $D \in \{5, 10, 20\}$. Representative results for $K \in \{1, 3, 5, 7, 9\}$ are reported in Table 4, with complete results provided in Table 7.

Increasing K generally improves approximation accuracy on X-shape, GMM3, and Ring, although the gains vary across dimensions. However, larger bases do not always lead to better results. In particular, performance deteriorates sharply in several higher-dimensional settings at $K = 10$, suggesting increased optimization difficulty. Funnel exhibits a different pattern, where increasing K does not consistently improve approximation accuracy. Together with the local-window ablation, these results suggest that richer local features alone are insufficient: effective approximation also requires sufficient capacity to model dependencies between variables.

Ancilla size E. Recall that E denotes the number of ancilla wires, which enable QuanVI to represent mixed states through purification. Increasing E expands the representational capacity of the model, but also introduces additional parameters and computational costs. We evaluate $E \in \{0, 2, 4\}$ for $D = 5$ and $E \in \{0, 2, 4, 6, 8, 10\}$ for $D \in \{10, 20\}$ across all non-Gaussian targets, with $E = 0$ corresponding to the pure-state baseline. Representative results are reported in Table 5, with complete results provided in Table 8.

Introducing ancilla wires improves approximation accuracy in several settings, demonstrating the potential benefit of mixed-state representations. However, increasing E does not consistently improve performance, and the gains vary across targets and dimensions. Similarly, larger E does not consistently reduce the variability across runs. These results suggest that mixed-state representations can improve approximation quality, but increasing the number of ancilla wires alone does not guarantee better accuracy or more stable optimization.

6 Related Work

Variational inference (VI). VI approximates Bayesian posteriors through KL-divergence minimization or ELBO maximization [2, 17, 18]. Probabilistic circuits (PCs) provide tractable probabilistic models [19, 20], with connections to tensor factorizations, sum-of-squares representations, and positive semidefinite matrix computations [21, 22, 23]. Score-based VI (SBVI) instead minimizes Fisher divergence using target score evaluations [10, 24, 5, 6, 25]. EigenVI [1] extends SBVI beyond Gaussian families through orthogonal basis expansions, reducing Fisher-divergence minimization to an eigenvalue problem. However, its explicit tensor-product coefficients grow exponentially with dimension, and its formulation uses a single eigenvector. QuanVI extends EigenVI to density operators and employs a compact QTN parameterization to improve scalability.

Tensor networks (TNs). Tensor trains (TT/MPS) and matrix product operators (MPOs) represent high-dimensional arrays through local tensor contractions [26, 27], with applications in density estimation [28] and distributional optimization [29]. Born machines construct distributions from squared amplitudes [30, 9], while locally purified TNs support mixed-state representations [31]. Recent theory establishes approximation hardness and polynomial-bond guarantees for MPO Born machines under suitable assumptions [11]. Circuit-inspired TNs have also been explored in quantum learning and state preparation [32, 12], model adaptation [33, 34], and joint structure search [35]. QuanVI combines QTN purification with a Fisher-derived spectral loss and local loss approximation, enabling tractable probabilistic inference and scalable SBVI.

7 Conclusion

We introduced QuanVI, a quantum-inspired score-based variational inference method based on density operators and quantum tensor networks. The density-operator formulation extends EigenVI from a single eigenvector to mixed-state representations, allowing multiple eigen-directions to contribute to the variational solution. A local approximation of the loss enables efficient estimation using low-dimensional computations. The QTN parameterization further avoids constructing the global operator, enabling QuanVI to scale to $D = 100$ in our experiments.

Experiments on synthetic distributions and Bayesian posterior benchmarks demonstrate accurate approximation of Gaussian targets and competitive performance on structured non-Gaussian distributions. Ablation studies further illustrate the trade-offs between expressiveness, computational efficiency, and optimization stability. These results demonstrate the potential of QuanVI for scalable score-based variational inference and motivate future research on adaptive QTN architectures and more general dependency structures.

8 Limitations

QuanVI uses a local approximation of the loss, which may introduce errors for distributions with long-range dependencies. Its expressiveness depends on the QTN architecture, while increasing model capacity raises computational costs without guaranteeing better accuracy. Training can also be slower than simpler variational baselines. Finally, broader Bayesian benchmarks with matched computational budgets are needed to assess its practical advantages.

Acknowledgments

YC, CL, ZS, and QZ were supported by JSPS KAKENHI Grant Number JP24K03005. ZT was supported by JSPS KAKENHI Grant Number JP26K21321 and the RIKEN Incentive Research Project. QZ was additionally supported by JSPS KAKENHI Grant Number JP23K28109 and the RIKEN TRIP initiative (RIKEN Quantum). YC, CL, and ZS also received support from the JHPC-quantum Test User Program.

References

[1] Diana Cai, Chirag Modi, Charles C Margossian, Robert M Gower, David M Blei, and Lawrence K Saul. Eigenvi: score-based variational inference with orthogonal function expansions. *Advances in Neural Information Processing Systems*, 37:132691–132721, 2024.

[2] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[3] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[4] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[5] Chirag Modi, Robert Gower, Charles Margossian, Yuling Yao, David Blei, and Lawrence Saul. Variational inference with gaussian score matching. *Advances in Neural Information Processing Systems*, 36:29935–29950, 2023.

[6] Diana Cai, Chirag Modi, Loucas Pillaud-Vivien, Charles Margossian, Robert M Gower, David Blei, and Lawrence K Saul. Batch and match: black-box variational inference with a score-based divergence. In *Forty-first International Conference on Machine Learning*, 2024.

[7] Frank Verstraete, Juan J Garcia-Ripoll, and Juan Ignacio Cirac. Matrix product density operators: Simulation of finite-temperature and dissipative systems. *Physical review letters*, 93(20):207204, 2004.

[8] Måns Magnusson, Jakob Torgander, Paul-Christian Bürkner, Lu Zhang, Bob Carpenter, and Aki Vehtari. posteriordb: Testing, benchmarking and developing bayesian inference algorithms. *arXiv preprint arXiv:2407.04967*, 2024.

[9] Alex Meiburg, Jing Chen, Jacob Miller, Raphaëlle Tihon, Guillaume Rabusseau, and Alejandro Perdomo-Ortiz. Generative learning of continuous data by tensor networks. *SciPost Physics*, 18(3):096, 2025.

[10] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[11] Chao Li, Zerui Tao, Yuchen Cong, Jian Xu, and Qibin Zhao. On the approximation complexity of matrix product operator (MPO) born machines. *arXiv preprint arXiv:2605.11471*, 2026. URL https://arxiv.org/abs/2605.11471.

[12] Shi-Ju Ran. Encoding of matrix product states into quantum circuits of one- and two-qubit gates. *Physical Review A*, 101(3):032310, 2020. doi: 10.1103/PhysRevA.101.032310. URL https://doi.org/10.1103/PhysRevA.101.032310.

[13] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[14] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *Journal of machine learning research*, 18(14):1–45, 2017.

[15] Marguerite Petit-Talamon, Marc Lambert, and Anna Korba. Variational inference with mixtures of isotropic gaussians. *arXiv preprint arXiv:2506.13613*, 2025.

[16] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[17] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of machine learning research*, 2013.

[18] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

[19] Andy Shih and Stefano Ermon. Probabilistic circuits for variational inference in discrete graphical models. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/31784d9fc1fa0d25d04eae50ac9bf787-Abstract.html.

[20] Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. In *Advances in Neural Information Processing Systems*, volume 34, pages 13189–13201, 2021. URL https://papers.neurips.cc/paper_files/paper/2021/hash/6e01383fd96a17ae51cc3e15447e7533-Abstract.html.

[21] Lorenzo Loconte, Antonio Mari, Gennaro Gala, Robert Peharz, Cassio de Campos, Erik Quaeghebeur, Gennaro Vessio, and Antonio Vergari. What is the relationship between tensor factorizations and circuits (and how can we exploit it)? *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=Y7dRmpGiHj.

[22] Lorenzo Loconte, Stefan Mengel, and Antonio Vergari. Sum of squares circuits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19077–19085, 2025. doi: 10.1609/aaai.v39i18.34100. URL https://ojs.aaai.org/index.php/AAAI/article/view/34100.

[23] Pedro Zuidberg Dos Martires. A quantum information theoretic approach to tractable probabilistic models. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pages 3013–3030. PMLR, 2025. URL https://proceedings.mlr.press/v286/zuidberg-dos-martires25a.html.

[24] Longlin Yu and Cheng Zhang. Semi-implicit variational inference via score matching. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sd90a2ytrt.

[25] Diana Cai, Robert M. Gower, David Blei, and Lawrence K. Saul. Fisher meets feynman: score-based variational inference with a product of experts. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=yG8vmj3EAU.

[26] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[27] Rom'an Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of physics*, 349:117–158, 2014.

[28] Georgii S Novikov, Maxim E Panov, and Ivan V Oseledets. Tensor-train density estimation. In *Uncertainty in Artificial Intelligence*, pages 1321–1331. PMLR, 2021.

[29] Yuning Qiu, Andong Wang, Chao Li, Haonan Huang, Guoxu Zhou, and Qibin Zhao. STEPS: Sequential probability tensor estimation for text-to-image hard prompt search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28640–28650, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Qiu_STEPS_Sequential_Probability_Tensor_Estimation_for_Text-to-Image_Hard_Prompt_Search_CVPR_2025_paper.html.

[30] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Physical Review X*, 8(3):031012, 2018.

[31] Ivan Glasser, Ryan Sweke, Nicola Pancotti, Jens Eisert, and Ignacio Cirac. Expressive power of tensor-network factorizations for probabilistic modeling. *Advances in neural information processing systems*, 32, 2019.

[32] William Huggins, Piyush Patil, Bradley Mitchell, K. Birgitta Whaley, and E. Miles Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and Technology*, 4(2):024001, 2019. doi: 10.1088/2058-9565/aaea94. URL https://doi.org/10.1088/2058-9565/aaea94.

[33] Zhuo Chen, Rumen Dangovski, Charlotte Loh, Owen Dugan, Di Luo, and Marin Soljačić. QuanTA: Efficient High-Rank Fine-Tuning of LLMs with Quantum-Informed Tensor Adaptation. *Advances in Neural Information Processing Systems*, 37:92210–92245, 2024.

[34] Binghua Li, Ziqing Chang, Tong Liang, Chao Li, Toshihisa Tanaka, Shigeki Aoki, Qibin Zhao, and Zhe Sun. Parameter-efficient fine-tuning of 3D DDPM for MRI image generation using tensor networks. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2025*, volume 15963 of *Lecture Notes in Computer Science*, pages 382–392. Springer Nature Switzerland, 2025. doi: 10.1007/978-3-032-04965-0_36. URL https://doi.org/10.1007/978-3-032-04965-0_36.

[35] Joint TN-SS. Joint structure search for tensor network operators inspired by symmetry breaking. OpenReview manuscript, 2026. URL https://openreview.net/forum?id=DZ76Xr7zct.

A Preliminary

A.1 Derivation of M

We derive the quadratic form in Eq. (3). Let $a_\theta(x) := \langle \theta, \Phi(x) \rangle$, so that $q_\theta = a_\theta^2$ and $\nabla_x \log q_\theta = 2\nabla_x a_\theta / a_\theta$. Define the derivative matrix $\dot{\Phi}(x) \in \mathbb{R}^{D \times K^D}$ by

$$\dot{\Phi}(x) = \begin{pmatrix} \dot{\phi}(x_1) \otimes \left[\bigotimes_{d=2}^{D} \phi(x_d)^\top\right] \\ \vdots \\ \left[\bigotimes_{d=1}^{\ell-1} \phi(x_d)^\top\right] \otimes \dot{\phi}(x_\ell) \otimes \left[\bigotimes_{d=\ell+1}^{D} \phi(x_d)^\top\right] \\ \vdots \\ \left[\bigotimes_{d=1}^{D-1} \phi(x_d)^\top\right] \otimes \dot{\phi}(x_D) \end{pmatrix}, \tag{12}$$

where

$$\dot{\phi}(x_d) := \left[\frac{d\phi_1(x_d)}{dx_d}, \frac{d\phi_2(x_d)}{dx_d}, \ldots, \frac{d\phi_K(x_d)}{dx_d}\right] \in \mathbb{R}^{1 \times K}.$$

Thus, $\nabla_x a_\theta(x) = \dot{\Phi}(x)\theta$. Let $s(x) := \nabla_x \log p(x)$. Then

$$\begin{aligned} q_\theta(x) \left\| \nabla_x \log p(x) - \nabla_x \log q_\theta(x) \right\|_F^2 &= a_\theta(x)^2 \left\| s(x) - 2\frac{\dot{\Phi}(x)\theta}{a_\theta(x)} \right\|_F^2 \\ &= \left\| a_\theta(x) s(x) - 2\dot{\Phi}(x)\theta \right\|_F^2 \\ &= \left\| \left(s(x)\Phi(x)^\top - 2\dot{\Phi}(x) \right) \theta \right\|_F^2. \end{aligned}$$

Integrating over x gives $\int q_\theta \|s - \nabla_x \log q_\theta\|^2 dx = \theta^\top M \theta$, with M defined in Eq. (4); that is, the objective in Eq. (2) equals $\frac{1}{2}\theta^\top M \theta$. Since M is an integral of Gram matrices, it is positive semidefinite.

B Detailed Complexity Analysis

Let D be the number of variables, E the number of ancilla wires, K the wire dimension, L the local window size, and N the number of generated samples. The implementation uses

$$J = D + E - L + 1$$

gates, each containing K^{2L} entries. Let B denote the total proposal-sample budget used to estimate all local operators in one loss evaluation. In the implementation, this budget is set to $\max(B, 100D)$ and divided across the D local operators.

Memory. The trainable gates require $O(JK^{2L})$ memory. The stored local operators require $O(DK^6)$ memory, since each acts on at most three variables. The full density operator would require $O(K^{2D})$ memory, which the tensor-network representation avoids.

Training. At each optimization step, the code estimates D local operators using a total of at most $\max(B, 100D)$ proposal samples, resampling the operators at each step. The local Gram-matrix construction costs

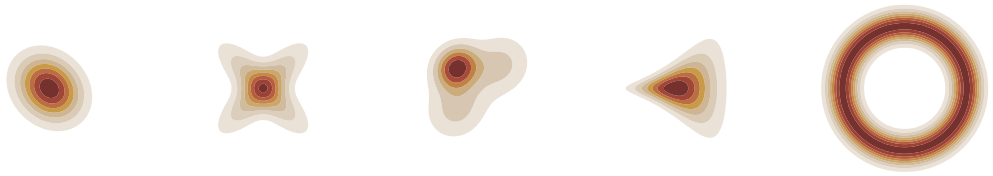
$$O\big(\max(B, 100D)(DK + K^6)\big),$$

excluding the cost of evaluating the target score. This corresponds to $\max(B, 100D)$ full D-dimensional score evaluations per step.

The loss is a sum of D local contractions. Under the contraction order used by the implementation, their cost is bounded by $O(DJK^{O(L)})$ per step. The SGDG update applies a Cayley step to each $K^L \times K^L$ gate, costing $O(JK^{3L})$. Thus, for T optimization steps, the total cost is

$$O\Big(T\left[\max(B, 100D)(DK + K^6) + DJK^{O(L)} + JK^{3L}\right]\Big),$$

in addition to the target-score evaluation cost. For fixed K and L, the tensor-network and local-operator costs grow polynomially with D and E.



(a) Gaussian (b) X-shape (c) GMM3 (d) Funnel (e) Ring

Figure 3: Two-dimensional marginal distributions over the first two variables of the synthetic targets used in our experiments. Darker colors indicate higher probability density within each distribution.

Inference. A density query at one point requires a tensor-network contraction costing $O(JK^{3L})$ under the contraction scheme used here. Autoregressive sampling evaluates a conditional density on an n_{grid}-point grid for each sample and coordinate. Drawing N samples therefore costs

$$O\big(NDn_{\text{grid}}JK^{3L}\big)\,,$$

plus the cost of interpolation on the grid. The grid range is $[-5, 5]$; the scripts use 100 grid points for synthetic runs and 500 for PosteriorDB runs.

C Experiments

C.1 Computational resources

The synthetic benchmark experiments were run on CPU nodes with AMD EPYC 9654 processors and 1.5 TiB memory. The posterior benchmark experiments were run on CPU nodes with Intel Xeon Silver 4316 processors. The ablation studies were run on 112-core CPU nodes with 128 GB memory. No GPUs were used.

C.2 Synthetic experiments

C.2.1 Two-dimensional base targets

We specify the four two-dimensional non-Gaussian base targets shown in Fig. 3. Let $u = (x_1, x_2)^\top \in \mathbb{R}^2$, and write $p_0(u)$ for the corresponding base density. Throughout, $\mathcal{N}(\,\cdot\,; \mu, \Sigma)$ denotes a Gaussian density with mean μ and covariance Σ; for a univariate Gaussian, the second parameter denotes the variance.

GMM3. The three-component Gaussian mixture is

$$p_0(u) = 0.4\,\mathcal{N}\big(u; \mu_1, \tfrac{1}{2}\Sigma_0\big) + 0.3\,\mathcal{N}(u; \mu_2, \Sigma_0) + 0.3\,\mathcal{N}(u; \mu_3, \Sigma_0),$$

where

$$\mu_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \qquad \mu_2 = \begin{bmatrix} 1.1 \\ 1.1 \end{bmatrix}, \qquad \mu_3 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \qquad \Sigma_0 = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}.$$

X-shape. The X-shaped target is an equally weighted mixture of two zero-mean Gaussians:

$$p_0(u) = \frac{1}{2}\mathcal{N}(u; 0, \Sigma_+) + \frac{1}{2}\mathcal{N}(u; 0, \Sigma_-),$$

with

$$\Sigma_+ = \begin{bmatrix} 1.65 & 1.35 \\ 1.35 & 1.65 \end{bmatrix}, \qquad \Sigma_- = \begin{bmatrix} 1.65 & -1.35 \\ -1.35 & 1.65 \end{bmatrix}.$$

These covariance matrices are obtained by rotating $\mathrm{diag}(3, 0.3)$ by $+\pi/4$ and $-\pi/4$, respectively.

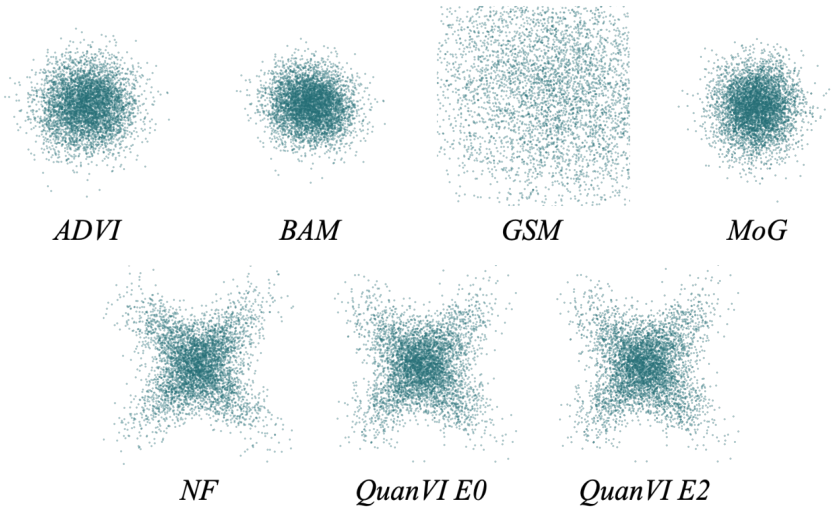


Figure 4: Sample visualization on the first two dimensions of the X-shape target with $D = 20$, comparing all baselines with our method.

Ring. The ring-shaped target has density

$$p_0(u) = \frac{1}{Z_{\text{ring}}} \exp\left(-\frac{(\|u\|_2 - r_0)^2}{2\sigma_{\text{ring}}^2}\right), \qquad r_0 = 3, \quad \sigma_{\text{ring}} = 0.5,$$

where the normalizing constant is

$$Z_{\text{ring}} = 2\pi \int_0^\infty r \exp\left(-\frac{(r - r_0)^2}{2\sigma_{\text{ring}}^2}\right) dr.$$

The density is rotationally symmetric and concentrates around the circle of radius r_0.

Funnel. The funnel target is defined by

$$x_1 \sim \mathcal{N}(0, \sigma_{\text{f}}^2), \qquad x_2 \mid x_1 \sim \mathcal{N}(0, e^{x_1}), \qquad \sigma_{\text{f}} = 1.2.$$

Equivalently, its joint density is

$$p_0(u) = \mathcal{N}(x_1; 0, \sigma_{\text{f}}^2)\,\mathcal{N}(x_2; 0, e^{x_1}).$$

The conditional standard deviation of x_2 is $e^{x_1/2}$, producing a narrow neck for negative x_1 and an increasingly dispersed region for positive x_1.

C.2.2 Construction of High-dimensional Synthetic Targets

We describe the construction of the high-dimensional synthetic targets used in the experiments. For the non-Gaussian targets, namely **GMM3**, **X-shape**, **Ring**, and **Funnel**, we start from a two-dimensional base target and augment it with a nonlinear Markov chain. The **Gaussian** target is constructed directly in D dimensions as a locally correlated Gaussian distribution, as described at the end of this section.

Let $p_0(x_1, x_2)$ denote one of the base two-dimensional non-Gaussian target densities in Fig. 3. For ambient dimension $D \geq 2$, define $N = D - 2$ auxiliary variables $z_1, \ldots, z_N$, with $z_0 = x_2$. The augmented target is

$$p(x_1, x_2, z_1, \ldots, z_N) = p_0(x_1, x_2) \prod_{i=1}^{N} \mathcal{N}\big(z_i;\, h_i(z_{i-1}),\, \sigma_{\text{aug}}^2\big)\,.$$

Hence the dependency structure is a single chain $x_2 \to z_1 \to z_2 \to \cdots \to z_N$, i.e., in coordinate indices $2 \to 3 \to \cdots \to D$.

Each $h_i : \mathbb{R} \to \mathbb{R}$ is selected deterministically from a fixed smooth function bank:

$$h(x) \in \Big\{ \pm 1.5\sin(x),\ \pm 1.5\cos(x),\ \pm 1.5\sin(2x),\ \pm 1.5\cos(2x),\\ \pm 1.5\big(\sigma(4x) - 1/2\big),\ \pm 1.5\tanh(2x),\ \pm 1.5\tanh(4x),\\ \pm 1.5\sin(x)\tanh(x),\ \pm 1.5\cos(x)\tanh(x),\ \pm 1.5\,\frac{x}{1+x^2} \Big\},$$

where $\sigma(\cdot)$ is the logistic sigmoid. In implementation, this bank is expanded deterministically by repetition, and we use the first $N = D - 2$ transforms as $h_1, \ldots, h_N$. Therefore, when D increases, the transform pattern continues in the same fixed order.

Sampling is sequential:

$$(x_1, x_2) \sim p_0, \qquad z_i = h_i(z_{i-1}) + \sigma_{\text{aug}}\epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0,1), \quad i = 1, \ldots, N.$$

Thus the original two-dimensional geometry is preserved in the marginal (x_1, x_2), while higher coordinates introduce nonlinear local dependencies along the chain.

The log-density is available in closed form:

$$\log p(x_1, x_2, z_1, \ldots, z_N) = \log p_0(x_1, x_2) - \frac{N}{2}\log(2\pi\sigma_{\text{aug}}^2) - \frac{1}{2\sigma_{\text{aug}}^2}\sum_{i=1}^{N}(z_i - h_i(z_{i-1}))^2 .$$

This expression is used for exact density evaluation. The score is also analytic from the Markov factorization. Let $s_0(x_1, x_2) = \nabla_{(x_1,x_2)} \log p_0(x_1, x_2)$. Then

$$\nabla_{x_1} \log p = [s_0(x_1, x_2)]_1, \nabla_{x_2} \log p = [s_0(x_1, x_2)]_2 + \frac{h_1'(x_2)}{\sigma_{\text{aug}}^2}\left(z_1 - h_1(x_2)\right),$$

and for $i = 1, \ldots, N$,

$$\nabla_{z_i} \log p = \frac{h_i(z_{i-1}) - z_i}{\sigma_{\text{aug}}^2} + \mathbf{1}_{\{i<N\}}\frac{h_{i+1}'(z_i)}{\sigma_{\text{aug}}^2}\left(z_{i+1} - h_{i+1}(z_i)\right).$$

The first term is the parent contribution $\log p(z_i \mid z_{i-1})$, and the second term, when $i < N$, is the child contribution from $\log p(z_{i+1} \mid z_i)$.

Overall, this construction yields high-dimensional targets that retain the non-Gaussian two-dimensional structure in (x_1, x_2), while adding controllable nonlinear local dependencies through a single path-structured chain. Exact sampling, exact log-density, and exact score are all available, and both sampling and score evaluation scale linearly with D.

Gaussian target. The Gaussian target is constructed directly in D dimensions rather than by augmenting a two-dimensional base distribution. Specifically, we use

$$p(x) = \mathcal{N}(x; 0, \Sigma), \qquad \Sigma = \Lambda^{-1},$$

where $\Lambda \in \mathbb{R}^{D\times D}$ is a tridiagonal precision matrix with diagonal entries 1 and neighboring off-diagonal entries 0.2:

$$\Lambda_{ii} = 1, \qquad \Lambda_{i,i+1} = \Lambda_{i+1,i} = 0.2.$$

All other entries are zero. This construction gives a locally correlated Gaussian distribution whose dependency structure follows a path graph, matching the Markov dependency assumption used in the main text. The log-density and score are available in closed form:

$$\log p(x) = -\frac{1}{2}x^\top \Lambda x + C, \qquad \nabla_x \log p(x) = -\Lambda x,$$

where C is the normalizing constant.

C.2.3 Hyperparameter settings for synthetic experiments

We summarize the hyperparameter settings used for the synthetic experiments. The target distributions are `xshape`, `ring`, `funnel`, `gmm3`, and `gaussian`. Unless otherwise stated, the ambient dimensions are

$$D \in \{5, 10, 20, 100\}.$$

Each configuration is repeated with five independent random seeds, and the evaluation follows Section 5.1. Following EigenVI, both EigenVI and QuanVI use normalized Hermite functions as the basis ϕ.

GSM. We run Gaussian score matching (GSM) with the Gaussian variational family. For each target and dimension, we use 2000 iterations. No learning rate or mini-batch size is required for this baseline.

BaM. We run Batch-and-Match VI (BaM) with the Gaussian variational family. The optimization budget is 2000 iterations. The batch size is set to

$$B_{\mathrm{BaM}} = \max(1, D-1),$$

which is the largest value satisfying the implementation constraint $B_{\mathrm{BaM}} < D$.

ADVI. We run ADVI with the same iteration budget as BaM, namely 2000 iterations. The batch size is also set to

$$B_{\mathrm{ADVI}} = \max(1, D-1).$$

Other optimizer parameters are kept at their default implementation values.

NF. We run the normalizing-flow baseline using the same synthetic targets and dimensions. The optimization budget is 2000 iterations. The number of Monte Carlo samples used by the NF baseline is set as

$$N_{\mathrm{NF}} = \mathrm{clip}(256D,\ 512,\ 8192).$$

Thus, for $D \in \{5, 10, 20, 100\}$, this gives

$$N_{\mathrm{NF}} \in \{1280, 2560, 5120, 8192\}.$$

Other architecture and optimizer parameters are kept at the default values of the NF implementation.

EigenVI. We use $K = 5$ basis functions, the same as QuanVI. The importance proposal is the product uniform distribution $\mathrm{Unif}([-5, 5])^{\otimes D}$, and the number of importance samples is $N_{\mathrm{EigenVI}} = \mathrm{clip}(2000D,\ 10^4,\ 2 \times 10^5)$. For $D > 5$, the global $K^D \times K^D$ matrix M cannot be stored in the available memory; for example, $K^D \approx 9.8 \times 10^6$ at $D = 10$. The corresponding entries are reported as OOM in Table 1.

MoG. We run mixture-of-Gaussians VI with isotropic covariance components. The default learning rate is 10^{-2}, the default number of iterations is 2000, the default number of mixture components is 5, and the default gradient batch size is 10. For more challenging targets, we use the following target-specific settings:

$$K_{\mathrm{mix}} = \begin{cases} 10, & \texttt{gmm3} \text{ and } \texttt{ring}, \\ 20, & \texttt{funnel}, \\ 5, & \text{otherwise}, \end{cases}$$

and

$$B_{\mathrm{grad}} = \begin{cases} 15, & \texttt{gmm3} \text{ and } \texttt{ring}, \\ 20, & \texttt{funnel}, \\ 10, & \text{otherwise}. \end{cases}$$

For the `funnel` target, we also reduce the learning rate to 10^{-3} and increase the number of iterations to 3500. The KL estimation batch size is $B_{\mathrm{KL}} = 1000$. We set `vgmm_scale_cov`$= 5$ and `vgmm_sample_boule`$= 5$.

QuanVI. For QuanVI, we use the same synthetic targets and dimensions as above. We run both the pure-state and mixed-state variants by setting

$$E \in \{0, 2\},$$

where $E = 0$ corresponds to a pure-state ansatz and $E = 2$ introduces ancilla degrees of freedom for mixed-state representations. At each optimization step, the total number of samples used to estimate all D local operators scales linearly with dimension:

$$B_{\mathrm{QuanVI}} = 500D.$$

This total budget is divided evenly across the D local operators, giving about 500 samples per operator at each step. For example, $B_{\mathrm{QuanVI}} = 10^4$ when $D = 20$, and $B_{\mathrm{QuanVI}} = 5 \times 10^4$ when $D = 100$. We use $K = 5$ basis functions and local window size $L = 2$, and train for 2000 optimization steps without target-specific tuning. All other optimizer parameters are kept at the default values of the implementation unless explicitly stated otherwise.

Table 6: Normalizing flow results on synthetic targets over five runs. Values report the median forward KL divergence, with the number of divergent runs in parentheses.

D	Gaussian	X-shape	GMM3	Funnel	Ring
5	0.0109 (0/5)	0.0319 (0/5)	0.0297 (0/5)	0.0778 (2/5)	0.2682 (0/5)
10	0.0247 (0/5)	4.36×10^{17} (5/5)	0.6085 (1/5)	1.05×10^{15} (4/5)	6.25×10^{28} (3/5)
20	0.0224 (0/5)	184.8571 (1/5)	5.06×10^{13} (3/5)	1.7613 (0/5)	3.7114 (1/5)
100	0.0774 (1/5)	37.5930 (1/5)	404.6059 (2/5)	3.38×10^{23} (4/5)	44.0952 (0/5)

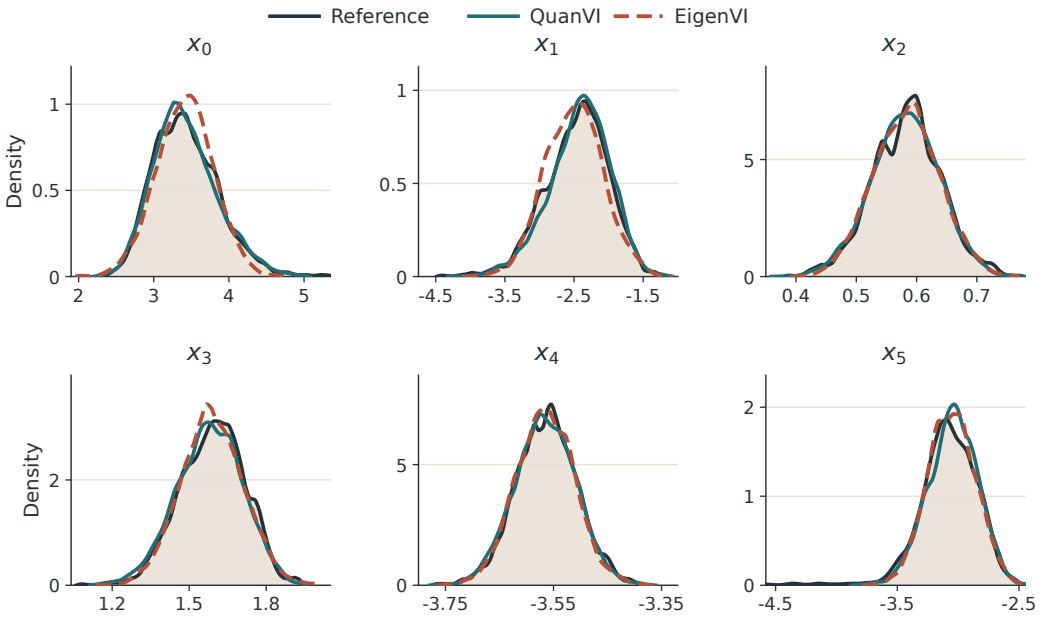


Figure 5: One-dimensional marginals of the six coordinates of `hmm_bball_0`, comparing QuanVI and EigenVI with reference posterior samples.

C.2.4 Result of NF

We report the normalizing-flow baseline results in Table 6 for completeness. Although NF achieves low KL divergence on some low-dimensional targets, its results are highly unstable across different targets and dimensions. In several high-dimensional or strongly non-Gaussian settings, the KL divergence becomes extremely large, indicating numerical or optimization instability. This sensitivity makes NF difficult to use as a reliable main-table baseline in our experimental setting. Therefore, we omit NF from the main comparison table and include the detailed results here in the appendix.

C.3 Posterior experiments

C.3.1 Standardization

For the POSTERIORDB experiments, we apply a two-stage reparameterization before variational training. First, the constrained Stan parameters $\boldsymbol{\theta}$ are mapped to unconstrained coordinates $\mathbf{x} \in \mathbb{R}^D$ using Stan's bijective transformation. The target log-density and score in this space are evaluated through BRIDGESTAN with the Jacobian adjustment enabled.

We then standardize $\mathbf{x}$ using a full-covariance Gaussian preconditioner $\mathcal{N}(\widehat{\mathbf{m}}, \widehat{\boldsymbol{\Sigma}})$. Let $\widehat{\boldsymbol{\Sigma}} = \mathbf{L}\mathbf{L}^\top$. We define

$$\mathbf{y} = \mathbf{L}^{-1}(\mathbf{x} - \widehat{\mathbf{m}}), \qquad \mathbf{x} = \widehat{\mathbf{m}} + \mathbf{L}\mathbf{y}.$$

The corresponding target score in standardized coordinates is

$$\nabla_{\mathbf{y}} \log \tilde{p}(\mathbf{y}) = \mathbf{L}^\top \nabla_{\mathbf{x}} \log p(\mathbf{x}), \qquad \mathbf{x} = \widehat{\mathbf{m}} + \mathbf{L}\mathbf{y}.$$

The preconditioner $(\widehat{\mathbf{m}}, \widehat{\mathbf{\Sigma}})$ is obtained before QuanVI training using GSM for 10^4 iterations with batch size 16; BaM is used for `hmm_bball_0`, where GSM is numerically unstable. Reference posterior draws are not used to construct the preconditioner. The same preconditioner is shared by all methods that use standardization.

QuanVI is trained directly in the standardized coordinates $\mathbf{y}$: basis functions are evaluated in this space and collocation points are sampled from $[-5, 5]^D$. For evaluation, the learned score is transformed back to the unconstrained coordinates as $\nabla_{\mathbf{x}} \log q(\mathbf{x}) = \mathbf{L}^{-\top} \nabla_{\mathbf{y}} \log \tilde{q}(\mathbf{y})$. All reported Fisher divergences are computed in this common unconstrained $\mathbf{x}$-space. Synthetic targets are already constructed on a comparable scale and therefore use the identity transformation.

C.3.2 Hyperparameter settings for posterior benchmark experiments

We summarize the hyperparameter settings used for the posterior benchmark experiments. We evaluate three PosteriorDB targets, denoted by the short names `hmm`, `hmm_bball_0`, and `gpregr`. These short names are mapped to the corresponding PosteriorDB models as follows:

Short name	PosteriorDB model
`hmm`	`hmm_example-hmm_example`
`hmm_bball_0`	`bball_drive_event_0-hmm_drive_0`
`gpregr`	`gp_pois_regr-gp_regr`

Each configuration is repeated with four independent random seeds, and we report the mean and standard deviation across runs. Unless otherwise stated, the same hyperparameters are used for all three targets. Different from the synthetic experiments, several posterior baselines use Gaussian initialization or standardization obtained from GSM or BaM fits.

GSM. We run Gaussian score matching with the Gaussian variational family. GSM is used both as a baseline and as an initialization method for other posterior baselines. We use 10000 iterations and batch size 16.

BaM. We run Batch-and-Match VI with the Gaussian variational family. We use 1000 iterations and batch size 16. BaM is initialized from the GSM-fitted mean and covariance with `init_cov_scale`$=$ 1.0 and `init_cov_jitter`$= 10^{-6}$.

ADVI. We run ADVI for 1000 iterations with batch size 16 and learning rate 10^{-4}. ADVI is initialized from the GSM-fitted mean and covariance with `init_cov_scale`$=$ 1.0 and `init_cov_jitter`$= 10^{-6}$.

NF. We run the normalizing-flow baseline without using Gaussian initialization files. The flow uses hidden dimension 32, 8 layers, 1000 optimization iterations, 16 Monte Carlo samples, learning rate 10^{-3}, and $\beta = 1.0$.

EigenVI. Since the size of the global eigenvalue problem grows as K^D, we use a dimension-dependent basis size: $K = 4$ for `gpregr` and `hmm` ($D \leq 5$), and $K = 2$ for `hmm_bball_0` ($D = 6$). We use 60000 importance samples, and the target is standardized as described in Appendix C.3.1.

MoG. We run mixture-of-Gaussians VI with isotropic covariance components and natural gradient covariance updates. We use

$$\texttt{mode} = \texttt{iso}, \qquad \texttt{variance_update} = \texttt{ngd}, \qquad \eta = 10^{-4}.$$

The number of iterations is 10000, the number of mixture components is 50, the KL batch size is $B_{\mathrm{KL}} = 1000$, and the gradient batch size is $B_{\mathrm{grad}} = 20$. We set `vgmm_sample_boule`$=$ 1 and `vgmm_scale_cov`$=$ 1. MoG is initialized from GSM-fitted Gaussian statistics whenever available, with `init_cov_scale`$=$ 1.0, `init_mean_scale`$=$ 0.25, `init_component_cov_scale`$=$ 0.5, and `init_cov_jitter`$= 10^{-6}$. If these files are unavailable, MoG falls back to random initialization.

Table 7: Complete ablation results for basis size $K \in \{1, \ldots, 10\}$ on forward KL divergence (mean $\pm$ std over five runs; lower is better). Bold indicates the lowest mean in each row.

D	Target	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
5	X-shape	2.024 $\pm$0.010	1.282 $\pm$0.022	0.863 $\pm$0.023	0.433 $\pm$0.007	0.265 $\pm$0.007	0.157 $\pm$0.005	0.072 $\pm$0.004	0.033 $\pm$0.003	0.024 $\pm$0.003	**0.012** $\pm$0.001
	GMM3	2.198 $\pm$0.029	1.338 $\pm$0.015	0.846 $\pm$0.029	0.363 $\pm$0.011	0.160 $\pm$0.004	0.145 $\pm$0.006	0.049 $\pm$0.002	0.029 $\pm$0.004	0.020 $\pm$0.001	**0.008** $\pm$0.000
	Funnel	2.141 $\pm$0.033	2.059 $\pm$0.282	**1.593** $\pm$0.117	3.937 $\pm$1.857	2.838 $\pm$1.262	2.876 $\pm$0.181	3.488 $\pm$0.317	3.927 $\pm$0.074	4.247 $\pm$0.154	4.629 $\pm$0.348
	Ring	4.700 $\pm$0.017	2.441 $\pm$0.028	1.595 $\pm$0.028	1.585 $\pm$0.032	0.879 $\pm$0.018	0.447 $\pm$0.013	0.289 $\pm$0.004	1.025 $\pm$0.025	0.276 $\pm$0.005	**0.108** $\pm$0.004
10	X-shape	4.831 $\pm$0.029	4.453 $\pm$0.024	4.399 $\pm$0.106	3.962 $\pm$0.153	3.212 $\pm$0.069	2.923 $\pm$0.092	1.865 $\pm$0.029	0.948 $\pm$0.036	**0.787** $\pm$0.037	14.654 $\pm$1.949
	GMM3	4.984 $\pm$0.032	4.486 $\pm$0.057	4.486 $\pm$0.053	3.994 $\pm$0.077	3.146 $\pm$0.065	2.957 $\pm$0.039	1.772 $\pm$0.063	1.005 $\pm$0.042	**0.823** $\pm$0.078	13.896 $\pm$0.996
	Funnel	**4.934** $\pm$0.037	5.106 $\pm$0.197	5.220 $\pm$0.134	7.851 $\pm$1.292	7.929 $\pm$1.195	8.453 $\pm$0.844	9.420 $\pm$0.761	8.129 $\pm$1.184	11.118 $\pm$0.511	12.922 $\pm$0.716
	Ring	7.488 $\pm$0.035	5.577 $\pm$0.095	5.144 $\pm$0.029	5.125 $\pm$0.102	3.822 $\pm$0.047	3.241 $\pm$0.131	2.056 $\pm$0.050	1.937 $\pm$0.047	**1.062** $\pm$0.045	15.105 $\pm$1.830
20	X-shape	10.268 $\pm$0.088	6.410 $\pm$0.106	9.515 $\pm$0.209	8.481 $\pm$1.162	7.914 $\pm$1.599	7.996 $\pm$0.677	**5.289** $\pm$1.775	5.751 $\pm$1.048	5.345 $\pm$0.896	25.636 $\pm$4.765
	GMM3	10.407 $\pm$0.039	6.360 $\pm$0.054	10.067 $\pm$0.752	9.254 $\pm$0.486	8.331 $\pm$0.415	8.177 $\pm$1.017	6.399 $\pm$1.763	5.545 $\pm$1.195	4.801 $\pm$1.391	**4.346** $\pm$1.101
	Funnel	10.331 $\pm$0.073	**7.110** $\pm$0.328	10.671 $\pm$0.397	13.264 $\pm$1.533	13.766 $\pm$0.583	14.424 $\pm$0.430	14.322 $\pm$0.513	14.617 $\pm$0.698	15.645 $\pm$1.036	20.068 $\pm$0.902
	Ring	12.870 $\pm$0.052	7.555 $\pm$0.074	10.416 $\pm$0.190	10.280 $\pm$0.426	9.159 $\pm$0.646	8.074 $\pm$0.895	**5.483** $\pm$1.505	5.603 $\pm$1.437	5.543 $\pm$1.176	21.898 $\pm$0.972

Table 8: Complete ablation results for ancilla size E on forward KL divergence (mean $\pm$ std over five runs; lower is better). Bold indicates the lowest mean in each row. A dash denotes an untested setting.

D	Target	$E=0$	$E=1$	$E=2$	$E=3$	$E=4$	$E=5$	$E=6$	$E=7$	$E=8$	$E=9$	$E=10$
5	X-shape	0.436 $\pm$0.162	0.350 $\pm$0.050	0.388 $\pm$0.040	0.392 $\pm$0.052	0.396 $\pm$0.038	**0.349** $\pm$0.016	–	–	–	–	–
	GMM3	**0.173** $\pm$0.023	0.195 $\pm$0.021	0.202 $\pm$0.033	0.206 $\pm$0.031	0.203 $\pm$0.014	0.214 $\pm$0.076	–	–	–	–	–
	Funnel	3.131 $\pm$0.553	2.985 $\pm$0.562	3.763 $\pm$0.782	3.434 $\pm$0.811	3.193 $\pm$0.342	**2.670** $\pm$0.714	–	–	–	–	–
	Ring	1.054 $\pm$0.143	0.967 $\pm$0.066	1.065 $\pm$0.204	1.083 $\pm$0.180	**0.927** $\pm$0.035	1.046 $\pm$0.123	–	–	–	–	–
10	X-shape	**3.231** $\pm$0.224	3.382 $\pm$0.142	3.253 $\pm$0.188	3.334 $\pm$0.135	3.339 $\pm$0.250	3.415 $\pm$0.195	3.277 $\pm$0.241	3.321 $\pm$0.100	3.385 $\pm$0.177	3.470 $\pm$0.198	3.366 $\pm$0.248
	GMM3	3.405 $\pm$0.343	3.283 $\pm$0.269	3.216 $\pm$0.099	3.309 $\pm$0.104	3.250 $\pm$0.125	3.206 $\pm$0.255	3.340 $\pm$0.324	3.239 $\pm$0.184	**3.091** $\pm$0.166	3.244 $\pm$0.206	3.234 $\pm$0.136
	Funnel	9.052 $\pm$1.045	8.781 $\pm$0.682	8.772 $\pm$0.581	9.151 $\pm$0.397	8.603 $\pm$0.808	8.688 $\pm$1.054	9.063 $\pm$0.322	8.614 $\pm$0.840	9.084 $\pm$0.486	8.680 $\pm$0.808	**7.870** $\pm$0.844
	Ring	3.948 $\pm$0.187	4.086 $\pm$0.398	3.956 $\pm$0.308	3.912 $\pm$0.288	**3.829** $\pm$0.191	3.958 $\pm$0.117	3.966 $\pm$0.099	4.073 $\pm$0.254	3.953 $\pm$0.249	4.063 $\pm$0.122	4.052 $\pm$0.126
20	X-shape	8.294 $\pm$1.607	7.664 $\pm$1.668	7.947 $\pm$1.502	8.764 $\pm$1.985	8.093 $\pm$0.189	**6.971** $\pm$1.496	8.519 $\pm$0.607	8.062 $\pm$1.324	7.306 $\pm$1.071	8.374 $\pm$1.411	8.491 $\pm$1.104
	GMM3	8.021 $\pm$0.862	7.944 $\pm$1.242	7.916 $\pm$1.518	8.068 $\pm$0.772	8.552 $\pm$0.452	8.500 $\pm$1.484	8.646 $\pm$0.715	8.308 $\pm$0.442	**7.648** $\pm$1.596	8.213 $\pm$0.556	8.154 $\pm$1.810
	Funnel	13.143 $\pm$1.655	13.480 $\pm$0.675	12.637 $\pm$1.569	13.677 $\pm$0.840	13.661 $\pm$0.550	12.756 $\pm$1.658	13.336 $\pm$1.518	13.543 $\pm$0.657	**12.345** $\pm$1.309	13.405 $\pm$1.369	13.843 $\pm$1.085
	Ring	8.080 $\pm$1.967	7.945 $\pm$1.077	8.285 $\pm$1.318	8.726 $\pm$1.454	7.829 $\pm$1.846	8.691 $\pm$1.005	7.532 $\pm$1.904	9.348 $\pm$0.442	8.062 $\pm$1.856	9.374 $\pm$0.495	**7.407** $\pm$1.978

QuanVI. QuanVI uses the same standardization as EigenVI, with $E \in \{0, 2\}$, $K = 5$, and $B = 5000$ training samples. The model is trained for 1000 steps with learning rate 10^{-1}, momentum 0.9, and complex double precision. Samples are drawn by inverse-CDF sampling on a 500-point grid.

C.4 Ablation study

All ablations use the non-Gaussian targets of Appendix C.2.2 at $D \in \{5, 10, 20\}$, five random seeds, and forward KL estimated from 10^4 target samples. Each ablation varies one hyperparameter while fixing the others at $L = 2$, $K = 5$, and $E = 0$. The varied ranges are given in Section 5.3. To cover a broad range of settings, every configuration is trained for 1000 steps instead of 2000, so some configurations may not have fully converged. While representative results are presented in the main text, the complete results for the basis size K and ancilla size E are provided in Tables 7 and 8, respectively.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the paper's scope: a density-operator formulation for degenerate low-energy eigenspaces, an MPO-motivated QTN parameterization for tractable computation, and empirical evaluation on synthetic targets, Bayesian posterior benchmarks, and ablations. The claims are framed as methodological and empirical, with scalability tied to local tensor-network structure rather than a general worst-case guarantee.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the main limitation that QuanVI relies on local structure for efficient tensor-network contractions. It also notes that larger local windows can improve expressiveness but increase memory and runtime, as shown in the ablation studies. The conclusion further identifies extensions to richer dependency graphs and adaptive tensor-network architectures as future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theorem-style theoretical results. It provides derivations of the density-operator formulation and complexity analysis under the stated locality assumption, which are used to motivate and analyze the method.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the information needed to reproduce the main experiments, including target construction, PosteriorDB benchmark mapping, evaluation metrics, baselines, hyperparameters, number of runs, ablation settings, and computational resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code needed to reproduce the main experimental results is included in the supplementary material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper reports the relevant training and evaluation details, including target construction, evaluation metrics, baselines, hyperparameters, optimization budgets, number of runs, and computational resources, with additional settings provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper reports means and standard deviations over repeated runs for the main quantitative comparisons and ablation studies where multiple runs are performed. These statistics provide uncertainty information for the empirical comparisons, although some large-scale settings use a limited number of repetitions due to computational cost.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides the main compute-resource information needed to contextualize the experiments, including CPU types, memory, and which resources were used for the synthetic, posterior, and ablation experiments. Detailed runtime comparisons are reported for settings where computational cost is directly analyzed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics and confirm that the research complies with the ethical standards and submission guidelines of the conference.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is methodological and does not present a broader impacts discussion beyond the standard scope of algorithmic development and benchmark evaluation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release data or models with high misuse risk, such as pretrained generative models, language models, image generators, or scraped datasets. The released code supports methodological experiments on synthetic targets and public benchmark models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper credits the existing assets and prior work used in the experiments, including POSTERIORDB, EigenVI, and the baseline methods. Public benchmark assets are used according to their intended research purpose, and the corresponding sources are cited in the main text and appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new datasets, models, or benchmark assets. The supplementary code is provided only to reproduce the reported experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as an important, original, or non-standard component of the core methodology, experiments, or scientific claims. Any use of LLMs, if applicable, was limited to writing, editing, or formatting assistance.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.