Rotated Mean-Field Variational Inference and Iterative Gaussianization

Yifan Chen[1] and Sifan Liu[2]

[1]Department of Mathematics, University of California, Los Angeles
[2]Department of Statistical Science, Duke University

October 2025

Abstract

We propose to perform mean-field variational inference (MFVI) in a rotated coordinate system that reduces correlations between variables. The rotation is determined by principal component analysis (PCA) of a cross-covariance matrix involving the target's score function. Compared with standard MFVI along the original axes, MFVI in this rotated system often yields substantially more accurate approximations with negligible additional cost.

MFVI in a rotated coordinate system defines a rotation and a coordinatewise map that together move the target closer to Gaussian. Iterating this procedure yields a sequence of transformations that progressively transforms the target toward Gaussian. The resulting algorithm provides a computationally efficient way to construct flow-like transport maps: it requires only MFVI subproblems, avoids large-scale optimization, and yields transformations that are easy to invert and evaluate. In Bayesian inference tasks, we demonstrate that the proposed method achieves higher accuracy than standard MFVI, while maintaining much lower computational cost than conventional normalizing flows.

1 Introduction

Sampling from an unnormalized density is a fundamental problem in statistics, with applications in Bayesian inference, inverse problems, statistical mechanics, and many others. Markov chain Monte Carlo (MCMC; [44, 25]) provides a general-purpose framework for this task, but it can become computationally expensive when applied to large datasets and often suffers from slow mixing in high dimensions. The difficulties are compounded when the target distribution exhibits strong correlations, variable curvature, or multiple modes—features that frequently arise in hierarchical models and demand sophisticated algorithmic adaptation. Additionally, while ensemble methods enable some degree of parallelization, the sequential dependence inherent in Markov chains limits their scalability on modern hardware.

Variational inference (VI; [31, 57, 7]) offers a scalable alternative by recasting sampling as an approximation problem: the goal is to find a distribution within a pre-specified variational family that is closest to the target under a divergence measure. A widely used variational family is the mean-field family, which approximates the target with a product distribution. Mean-field variational inference

(MFVI) is conceptually simple and can be implemented efficiently using coordinate ascent algorithms (CAVI; [6]). MFVI has been extensively studied, including its consistency and convergence rates in statistical estimation [5, 61], applications to model selection [62], algorithmic convergence of CAVI [2, 38, 4] and alternative algorithms [56, 16, 30], and theoretical properties for log-concave targets [36]. Despite its computational appeal and theoretical understanding, MFVI is known to underestimate marginal variances and to yield over-confident uncertainty quantification, as it cannot capture dependencies among coordinates [58, 45].

Beyond MFVI, richer variational families can be constructed to capture correlations, such as by using multivariate Gaussian or mixture distributions [46, 39, 10]. A more flexible approach, inspired by generative modeling, parametrizes the variational family as the pushforward of a simple reference distribution (e.g., a standard Gaussian) through an expressive class of transport maps. Normalizing flows [50, 47] offer a general framework for this construction by composing multiple simple, invertible transformations. With sufficient capacity, flows achieve leading performance in density estimation and generative modeling [14, 17, 60]. For sampling tasks, recent empirical studies show that flows can approximate complex targets with high accuracy, but only under demanding conditions: the model must be highly expressive, optimization requires very large Monte Carlo batches (e.g., 2^{19} samples), and the learning rate must be carefully tuned [8, 1]. While normalizing flows offer a promising approach to sampling, the training cost remains prohibitively high for many practical applications.

Motivated by the tractability of MFVI and the compositional design of normalizing flows, we propose a simple but effective strategy: construct expressive transport maps by iteratively performing MFVI in different coordinate systems. The key insight is that while MFVI in any fixed coordinate system produces only a product distribution, alternating between MFVI steps and orthogonal rotations can capture complex dependencies. This raises a natural question: *What is the best coordinate system in which to perform MFVI?*

To address this question, we analyze how the choice of coordinate system affects the reduction in Kullback-Leibler (KL) divergence achieved by MFVI. We show that this reduction is governed by the *projected Fisher information* [35] of the target distribution p relative to the standard Gaussian. We further derive a convenient lower bound for this quantity, expressed in terms of the matrix

$$H = \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,I_d)}\left[\mathbf{x}\big(\nabla \log p(\mathbf{x}) + \mathbf{x}\big)^\intercal\right],$$

which is the cross-covariance between $\mathbf{x} \sim \mathcal{N}(0, I_d)$ and the relative score $\nabla \log p(\mathbf{x}) + \mathbf{x}$. Importantly, this lower bound is maximized when the rotation is chosen as the eigenvectors of H, yielding a natural PCA-type procedure that we call *relative score PCA*.

Performing MFVI in such a rotated coordinate system yields a transport map composed of a rotation followed by a coordinatewise transformation, together moving the target closer to the standard Gaussian. Repeating this procedure leads to an iterative scheme in which each iteration consists of two steps:

- **Relative score PCA**: Compute the principal components of H to determine the rotation for the current target distribution;
- **MFVI**: Perform mean-field variational inference in this rotated coordinate system to obtain a coordinatewise map.

The composition progressively pushes the target toward the standard Gaussian, with each iteration involving only a PCA step and an MFVI update—both computationally efficient and

straightforward to implement. We refer to this iterative algorithm as *iterative Gaussianization*, as it conceptually parallels the Gaussianization method for density estimation [11]. In this analogy, the rotation serves as a "decorrelation" step, aiming to make the coordinates of the target distribution as independent as possible, while the MFVI step acts as "marginal Gaussianization," transforming each coordinate separately toward Gaussian. In Gaussianization for density estimation [11], the rotation is chosen via linear independent component analysis (ICA), a special case of projection pursuit [20, 29, 21]. In the same spirit, the proposed relative score PCA can be viewed as a projection pursuit method in the sampling context, where one does not have access to samples but can query the score function of the target. Just as projection pursuit iteratively uncovers "interesting" structure in data through linear projections, the proposed method iteratively reveals the "non-Gaussian" structure of the target distribution.

Performing MFVI on a rotated target can also be viewed as a form of *model reparametrization*, a long-standing idea in Bayesian computation for improving the efficiency of inference algorithms. Classical examples include the centered and non-centered parametrizations in hierarchical models [48]. Recent work has extended this principle to variational inference, where suitable reparametrizations can substantially improve the quality of variational approximations [55]. In contrast to these model-specific strategies, the proposed rotation method provides a data-driven, model-agnostic reparametrization that relies solely on the score function of the target distribution. Moreover, the proposed relative score PCA complements existing PCA-based approaches such as active subspaces [12] and certified dimension reduction [59], which seek informative low-dimensional subspaces. In contrast, our goal is to identify rotations that best facilitate coordinatewise updates; see Section 4.2 for a detailed comparison.

In summary, the main contributions of this work are as follows:

1. We develop a principled approach to enhance standard MFVI by performing it along the principal components identified by relative score PCA. The PCA step incurs negligible overhead yet can significantly improve the accuracy of the mean-field variational approximation.

2. Building on this idea, we introduce an iterative framework that alternates between relative score PCA and MFVI, progressively transforming the target distribution toward the standard Gaussian. The resulting transport maps achieve expressiveness comparable to normalizing flows while avoiding their heavy training costs. They are easily invertible with tractable Jacobian determinants, enabling efficient approximate sampling and density evaluation.

3. Our theoretical analysis establishes a characterization of Gaussianity: if a distribution admits the standard Gaussian as its optimal mean-field approximation under any rotation, then the distribution itself must be standard Gaussian (see Theorem 3.2). This result identifies the standard Gaussian as the unique stationary point of the proposed iterative algorithm and provides a characterization of mean-field optimality under rotations that may be of independent interest.

The remainder of the paper is organized as follows. Section 2 reviews MFVI and introduces the idea of performing it in rotated coordinate systems. Section 3 analyzes how the choice of rotation influences the KL divergence achieved by MFVI. Section 4 builds on this analysis to develop the relative score PCA procedure for selecting rotations, and discusses its connections with Stein discrepancy and alternative score-based PCA methods. Section 5 extends the one-step approach into the full iterative Gaussianization algorithm. Section 6 presents numerical results on a variety of

sampling tasks. Finally, Section 7 concludes the paper, and Appendices A and B provide proofs and additional experimental details.

2 MFVI with rotations

We begin by reviewing mean-field variational inference (MFVI) and then describe how it can be performed in rotated coordinate systems.

Let p denote the target distribution supported on $\mathbb{R}^d$, with density $p(\mathbf{x}) \propto \exp(-U(\mathbf{x}))$ assumed to be continuously differentiable. Let $\mathcal{P}_{\text{prod}}$ be the set of all product measures on $\mathbb{R}^d$ whose densities are positive and continuously differentiable. Define $\mathcal{F}$ as the set of coordinatewise diffeomorphisms on $\mathbb{R}^d$, i.e. $\mathcal{F} = \{F : \mathbb{R}^d \to \mathbb{R}^d \mid F(\mathbf{x}) = (F_1(x_1), \ldots, F_d(x_d)), F_i : \mathbb{R} \to \mathbb{R} \text{ diffeomorphism}\}$. For a distribution p and diffeomorphism F on $\mathbb{R}^d$, the pushforward $F\#p$ denotes the distribution of $F(X)$ when $X \sim p$. Let $\gamma = \mathcal{N}(0, I_d)$ denote the standard Gaussian distribution on $\mathbb{R}^d$.

2.1 Mean-field variational inference

MFVI approximates the target p with a product distribution that minimizes the KL divergence:

$$q^* = \operatorname*{argmin}_{q \in \mathcal{P}_{\text{prod}}} \ \mathrm{KL}(q \,\|\, p), \tag{1}$$

where the Kullback-Leibler (KL) divergence is defined as

$$\mathrm{KL}(q \,\|\, p) = \mathbb{E}_q \left[\log \frac{q(\mathbf{x})}{p(\mathbf{x})} \right].$$

The solution q^* of problem (1) satisfies the first-order optimality condition

$$\nabla_{x_i} \log q_i^*(x_i) = \mathbb{E}_{q^*_{-i}} \left[\nabla_{x_i} \log p(\mathbf{x}) \right], \quad i = 1, \ldots, d, \tag{2}$$

where the derivatives are understood in the weak sense, and $\mathbb{E}_{q^*_{-i}}$ denotes expectation under q^* conditioned on x_i. See [2, Equation 4.7] for a derivation under the assumptions that q^* has finite m-th moment ($m \geq 2$), and $|\log p(\mathbf{x})| \leq c(1 + \|\mathbf{x}\|_2^m)$, $\|\nabla \log p(\mathbf{x})\|_2 \leq c(1 + \|\mathbf{x}\|_2^m)$ for some $c > 0$ and almost every $\mathbf{x} \in \mathbb{R}^d$. We assume these conditions throughout.

In general, the solution of the mean-field equation (2) need not be unique. If the potential function $U(\mathbf{x})$ is convex, all solutions of (2) are global minimizers of the MFVI problem (1). If U is strongly convex, the global minimizer is unique [36, Theorem 1.1, Proposition 3.9].

The MFVI problem (1) can equivalently be formulated as finding the best coordinatewise map that pushes the standard Gaussian γ forward to p:

$$F^* = \operatorname*{argmin}_{F \in \mathcal{F}} \ \mathrm{KL}(F\#\gamma \,\|\, p).$$

Once F^* is obtained, we can Gaussianize the target by defining $p^* := F^{*-1}\#p$, which is closer to the standard Gaussian than the original target p, since

$$\mathrm{KL}(\gamma \,\|\, p^*) = \mathrm{KL}(F^*\#\gamma \,\|\, p) \leq \mathrm{KL}(\gamma \,\|\, p).$$

Algorithm 1 Coordinatewise Gaussianization via MFVI

Input: Target distribution p

Solve for $F^* = \operatorname{argmin}_{F \in \mathcal{F}} \mathrm{KL}(F\#\gamma \,\|\, p)$

return $p^* := F^{*-1}\#p$

The equality follows from the invariance of KL divergence under invertible transformations, and the inequality follows from the optimality of F^*. Thus, MFVI yields a coordinatewise map that Gaussianizes the target, as summarized in Algorithm 1.

After the Gaussianization step in Algorithm 1, the transformed target p^* has the standard Gaussian γ as its optimal MF approximation. By the optimality condition (2), this implies that

$$\mathbb{E}_{\gamma_{-i}}\left[\nabla_{x_i} \log p^*(\mathbf{x})\right] = -x_i, \quad i = 1, \ldots, d. \tag{3}$$

2.2 MFVI in a rotated coordinate system

Performing MFVI in the standard coordinate system produces only coordinatewise maps. To introduce dependence among variables, we alternate coordinatewise maps with orthogonal rotations. Given an orthogonal matrix $R \in \mathbb{R}^{d\times d}$ with $RR^\intercal = I_d$, we first rotate the target distribution to $p_R = R\#p$, whose density is $p_R(\mathbf{x}) = p(R^\intercal \mathbf{x})$, and then perform MFVI for the rotated target. Specifically, we solve

$$F^* = \operatorname*{argmin}_{F \in \mathcal{F}} \mathrm{KL}(F\#\gamma \,\|\, p_R).$$

As before, once F^* is found, the target is transformed to $p_R^* := F^{*-1}\#p_R$, which is closer to the standard Gaussian than the original target p, since

$$\mathrm{KL}(\gamma \,\|\, p_R^*) \le \mathrm{KL}(\gamma \,\|\, p_R) = \mathrm{KL}(\gamma \,\|\, p),$$

where the equality holds because KL divergence is invariant under orthogonal transformations and the standard Gaussian distribution γ is rotationally invariant.

The goal is to choose a rotation R that minimizes $\mathrm{KL}(\gamma \,\|\, p_R^*)$. In the ideal situation where p_R is already a product distribution, the MF approximation becomes exact, and $\mathrm{KL}(\gamma \,\|\, p_R^*) = 0$. This motivates choosing R so that p_R has as little dependence across coordinates as possible.

To build intuition, we consider the copula representation of p. By Sklar's theorem [51], any continuous density p can be written as the product of its marginals p_j and a copula c that encodes dependence:

$$p(\mathbf{x}) = \prod_{j=1}^{d} p_j(x_j) \cdot c(P_1(x_1), \ldots, P_d(x_d)),$$

where P_j is the cumulative distribution function (CDF) of p_j, and c is a joint density on $[0,1]^d$ with uniform marginals. Based on the copula expression of p, the KL divergence between γ and p can be decomposed as [24]

$$\mathrm{KL}(\gamma \,\|\, p) = \sum_{j=1}^{d} \mathrm{KL}(\gamma_j \,\|\, p_j) + \mathrm{KL}(\mathrm{Unif}(0,1)^d \,\|\, c(P \circ \Phi^{-1}(\mathbf{u}))), \tag{4}$$

where Φ^{-1} is the inverse CDF of standard normal, and $P \circ \Phi^{-1}$ is applied coordinatewise to the uniform vector $\mathbf{u} \sim \mathrm{Unif}(0,1)^d$. The first term in (4) measures the non-Gaussianity of each marginal p_j, while the second term measures the dependence among components.

Since $\mathrm{KL}(\gamma \,\|\, p) = \mathrm{KL}(\gamma \,\|\, p_R)$, any rotation R that reduces dependence among the coordinates of p_R necessarily increases the non-Gaussianity of its marginals. Thus, to make p_R closer to a product distribution, one should find a rotation that exposes the non-Gaussian features of p along the coordinate axes.

3 MFVI improvement

Performing MFVI on the rotated target p_R achieves the KL divergence $\inf_{F \in \mathcal{F}} \mathrm{KL}(F\#\gamma \,\|\, p_R)$. Our goal is to choose R in order to minimize this quantity. To this end, we define the MFVI improvement as

$$\Delta_{\mathrm{MFVI}}(p) = \mathrm{KL}(\gamma \,\|\, p) - \inf_{F \in \mathcal{F}} \mathrm{KL}(F\#\gamma \,\|\, p),$$

which measures the reduction in KL divergence achieved by MFVI, relative to the baseline KL divergence between γ and p.

We use γ as the baseline for three reasons. First, because γ is rotationally invariant,

$$\Delta_{\mathrm{MFVI}}(p_R) = \mathrm{KL}(\gamma \,\|\, p) - \inf_{F \in \mathcal{F}} \mathrm{KL}(F\#\gamma \,\|\, p_R).$$

Thus, maximizing $\Delta_{\mathrm{MFVI}}(p_R)$ over R is equivalent to minimizing $\inf_{F \in \mathcal{F}} \mathrm{KL}(F\#\gamma \,\|\, p_R)$, which is our original goal. Second, by Maxwell's theorem, the isotropic Gaussian is the only product measure that is rotationally invariant, so no other product distribution would satisfy the first property. Third, when MFVI updates (Algorithm 1) are applied iteratively, the transformed target at each stage has γ as its optimal mean-field approximation. Hence, at the start of each iteration, γ is the natural baseline against which to measure improvement. We emphasize, however, that the analysis in this section does not rely explicitly on the assumption that γ is the optimal mean-field approximation of p.

3.1 Projected Fisher information

The intuition from Section 2.2 suggests that the rotation R should be chosen to expose the non-Gaussian structure of the target p along the coordinate axes. To formalize this, we need a quantitative measure of coordinatewise non-Gaussianity that is both computationally tractable and easy to optimize over R. A natural first candidate is the Fisher divergence $\mathcal{D}_F(\gamma, p) = \mathbb{E}_\gamma \left[\|\nabla \log \gamma(\mathbf{x}) - \nabla \log p(\mathbf{x})\|_2^2 \right]$. However, the Fisher divergence is invariant under orthogonal transformations; that is $\mathcal{D}_F(\gamma, p_R) = \mathcal{D}_F(\gamma, p)$. As a result, it provides no guidance for choosing R.

Instead, we turn to the *projected Fisher information* introduced by [35], which captures precisely the coordinatewise discrepancy that MFVI seeks to correct. For a product distribution q and a target distribution p, it is defined as

$$\tilde{I}(q, p) = \sum_{i=1}^{d} \mathbb{E}_q \left[\Big(\mathbb{E}_{q_{-i}} \big(\nabla_i \log q(\mathbf{x}) - \nabla_i \log p(\mathbf{x}) \big) \Big)^2 \right], \tag{5}$$

where $\mathbb{E}_{q_{-i}}$ denotes the expectation under q conditioned on the i-th coordinate. The conditional expectation $\mathbb{E}_{q_{-i}}[\nabla_i \log q - \nabla_i \log p]$ discards the score information that is orthogonal to the i-th coordinate. Therefore, $\tilde{I}(q,p)$ can be interpreted as the squared norm of the orthogonal projection of the score mismatch onto the subspace of coordinatewise functions. It measures the portion of the discrepancy between q and p that can be corrected by adjusting only the marginals.

The next lemma shows that vanishing projected Fisher information is equivalent to the mean-field optimality condition.

Lemma 3.1. *A product distribution q satisfies the first-order optimality condition (2) if and only if $\tilde{I}(q,p) = 0$.*

Proof. Note that $\tilde{I}(q,p) = 0$ if and only if $\mathbb{E}_{q_{-i}}[\nabla_i \log q - \nabla_i \log p] = 0$ almost surely for $1 \le i \le d$. This condition is equivalent to $\nabla_i \log q_i(x_i) = \mathbb{E}_{q_{-i}}[\nabla_i \log p]$, which coincides with the first-order optimality condition for the MFVI problem in Equation (2). □

The following result shows that if the standard Gaussian γ satisfies the MFVI optimality condition for p_R under almost every rotation R, then p itself must be Gaussian.

Theorem 3.2. *If $\tilde{I}(\gamma, p_R) = 0$ for almost every orthogonal matrix R, then $p = \gamma$.*

The projected Fisher information $\tilde{I}(q,p)$ is not a divergence: $\tilde{I}(q,p) = 0$ does not imply $q = p$. However, for the Gaussian reference γ, if $\tilde{I}(\gamma, p_R)$ is zero for almost every R, then $p = \gamma$. Consequently, the quantity $\tilde{I}(\gamma, p_R)$, when maximized or averaged over R, does behave as a divergence between p and γ. The proof proceeds by showing that all higher-order Hermite coefficients of $\log(p/\gamma)$ vanish if $\tilde{I}(\gamma, p_R) = 0$ for almost every R; see details in Appendix A.1.

If p is log-concave and $\tilde{I}(\gamma, p) = 0$, then the standard Gaussian γ is already the optimal MF approximation of p, thus the MFVI improvement $\Delta_{\mathrm{MFVI}}(p)$ must be 0. However, unless $p = \gamma$, we can always rotate p by some orthogonal matrix R so that $\tilde{I}(\gamma, p_R)$ is strictly positive, in which case γ is not the MF approximation of p_R, thus $\Delta_{\mathrm{MFVI}}(p_R)$ is strictly positive.

Following the above discussion, each MFVI step can be seen as driving the projected FI $\tilde{I}(\gamma, p_R)$ to zero. The larger $\tilde{I}(\gamma, p_R)$ is, the greater the reduction in projected FI, which suggests choosing R to maximize $\tilde{I}(\gamma, p_R)$. Next, we show that projected Fisher information is directly linked to the MFVI improvement.

3.2 Projected FI controls MFVI improvement

We now make precise the connection between $\tilde{I}(\gamma, p)$ and the MFVI improvement $\Delta_{\mathrm{MFVI}}(p)$, following the techniques of [35]. In particular, it is shown in [35] that

$$\Delta_{\mathrm{MFVI}}(p) = \int_0^\infty \tilde{I}(q_t, p)\mathrm{d}t, \tag{6}$$

where q_t is the law of the independent projection of the Langevin dynamics $\{\mathbf{X}_t\}_{t\ge 0}$ at time t, which is the solution of

$$\mathrm{d}X_{t,i} = -\mathbb{E}_{q_{t,-i}}[\nabla_i U(\mathbf{X}_t)]\,\mathrm{d}t + \sqrt{2}\mathrm{d}B_{t,i}, \quad 1 \le i \le d.$$

Here, $U(\mathbf{x}) = -\log p(\mathbf{x})$, the initial condition is $\mathbf{X}_0 \sim \gamma$, and $\mathbf{B}_t = (B_{t,1}, \ldots, B_{t,d}) \in \mathbb{R}^d$ is a standard Brownian motion in $\mathbb{R}^d$. In this dynamics, each coordinate $X_{t,i}$ evolves with a drift term

that depends only on its own value. As a result, the coordinates of $\mathbf{X}_t$ remain independent for all $t \geq 0$. The law of $\mathbf{X}_t$, denoted as q_t, is always a product distribution.

Equation (6) shows that the MFVI improvement is the time integral of the projected Fisher information along the independent projection dynamics. Furthermore, when the log density is strongly concave and smooth, the MFVI improvement can be bounded above and below by the initial projected FI $\tilde{I}(\gamma, p)$, up to a correction term, as stated in the following theorem.

Theorem 3.3 (Adapted from [35, Theorem 2.5])**.** *Suppose* $p(\mathbf{x}) \propto \exp(-U(\mathbf{x}))$*, with* $\lambda I_d \preceq \nabla^2 U(\mathbf{x}) \preceq L I_d$ *for some constants* $0 < \lambda \leq L < \infty$*. Then*

$$\frac{1}{2L}\Big(\tilde{I}(\gamma, p) - J\Big) \;\leq\; \Delta_{\mathrm{MFVI}}(p) \;\leq\; \frac{1}{2\lambda}\Big(\tilde{I}(\gamma, p) - J\Big),$$

where $J = 2\sum_{i=1}^{d} \int_0^\infty \mathbb{E}_{q_{t,i}}\left[(\nabla_i h_{t,i}(x_i))^2\right] \mathrm{d}t$ *and*

$$h_{t,i}(x_i) = \nabla \log q_{t,i}(x_i) + \mathbb{E}_{q_{t,-i}}\left[\nabla_i U(\mathbf{x})\right].$$

The proof is adapted from [35] and is provided in Appendix A.2.

This theorem shows that the MFVI improvement is controlled by $\tilde{I}(\gamma, p) - J$. Since $J \leq \tilde{I}(\gamma, p)$, the correction term J is of the same or a smaller order than $\tilde{I}(\gamma, p)$. This implies that the projected Fisher information $\tilde{I}(\gamma, p)$ serves as a key quantity governing the MFVI improvement, and motivates choosing the rotation R to maximize $\tilde{I}(\gamma, p_R)$.

3.3 Lower bound on projected FI

The previous discussion motivates maximizing $\tilde{I}(\gamma, p_R)$ over R. However, computing and optimizing $\tilde{I}(\gamma, p_R)$ directly is difficult, since it involves nested expectations and optimization over the orthogonal group. To address this, we derive a convenient lower bound on $\tilde{I}(\gamma, p_R)$ that makes the problem more tractable.

Let $h(\mathbf{x}) = \nabla \log p(\mathbf{x}) + \mathbf{x}$ denote the relative score between p and γ, and define the cross-covariance matrix between $\mathbf{x} \sim \gamma$ and $h(\mathbf{x})$ as

$$H = H(p) = \mathbb{E}_\gamma\left[\mathbf{x} h(\mathbf{x})^\intercal\right]. \tag{7}$$

Theorem 3.4 (Lower bound on projected FI)**.** *The projected Fisher information* $\tilde{I}(\gamma, p_R)$ *is lower bounded by*

$$\tilde{I}(\gamma, p_R) \geq \sum_{i=1}^{d} (RHR^\intercal)_{ii}^2.$$

Equality holds if $p = \mathcal{N}(0, \Sigma)$*.*

Proof of Theorem 3.4. By the definition of $\tilde{I}(\gamma, p)$, we have

$$\tilde{I}(\gamma, p) = \sum_{i=1}^{d} \mathbb{E}_{\gamma_i}\left[(\mathbb{E}_{\gamma_{-i}}\left[\nabla_i \log p(\mathbf{x}) + x_i\right])^2\right] = \sum_{i=1}^{d} \mathbb{E}_{\gamma_i}\left[(\mathbb{E}_{\gamma_{-i}}\left[h_i(\mathbf{x})\right])^2\right].$$

Applying Cauchy-Schwarz inequality,

$$\mathbb{E}_{\gamma_i}\left[\left(\mathbb{E}_{\gamma_{-i}}\left[h_i(\mathbf{x})\right]\right)^2\right]\cdot\mathbb{E}_{\gamma_i}\left[x_i^2\right]\geq\left(\mathbb{E}_{\gamma_i}\left[\mathbb{E}_{\gamma_{-i}}\left[h_i(\mathbf{x})\right]\cdot x_i\right]\right)^2=\left(\mathbb{E}_{\gamma}\left[x_i h_i(\mathbf{x})\right]\right)^2.$$

Therefore, $\tilde{I}(\gamma,p)\geq\sum_{i=1}^d H(p)_{ii}^2$. In particular, we have $\tilde{I}(\gamma,p_R)\geq\sum_{i=1}^d H(p_R)_{ii}^2$, where

$$\begin{aligned}H(p_R)&=\mathbb{E}_\gamma\left[\mathbf{x}(R\nabla\log p(R^\intercal\mathbf{x})+\mathbf{x})^\intercal\right]\overset{\mathbf{y}=R^\intercal\mathbf{x}}{=}\mathbb{E}_\gamma\left[(R\mathbf{y})(R\nabla\log p(\mathbf{y})+R\mathbf{y})^\intercal\right]\\&=R\,\mathbb{E}_\gamma\left[\mathbf{y}(\nabla\log p(\mathbf{y})+\mathbf{y})^\intercal\right]R^\intercal=RH(p)R^\intercal.\end{aligned}$$

This proves that $\tilde{I}(\gamma,p_R)\geq\sum_{i=1}^d(RHR^\intercal)_{ii}^2$, where $H=H(p)$.

Finally, when $p=\mathcal{N}(0,\Sigma)$, one can check that $H=I_d-\Sigma^{-1}$, and $\tilde{I}(\gamma,p)=\sum_{i=1}^d(\Sigma_{ii}^{-1}-1)^2=\sum_{i=1}^d H_{ii}^2$, which shows that the bound is tight. □

As a sanity check, if $\tilde{I}(\gamma,p)=0$ and $R=I_d$, then $\tilde{I}(\gamma,p_R)=\tilde{I}(\gamma,p)=0$. In this case, the lower bound is also 0, since $\sum_{i=1}^d(RHR^\intercal)_{ii}^2=\sum_{i=1}^d H_{ii}^2=0$. The last equality follows from the MFVI optimality condition in Equation (3). In addition, the matrix H is symmetric because $H=\mathbb{E}_\gamma\left[\nabla^2\log p(\mathbf{x})+I_d\right]$ by Stein's identity.

Theorem 3.4 thus provides a convenient lower bound for $\tilde{I}(\gamma,p_R)$ that is straightforward to compute and optimize, as we show next.

4 Relative score PCA

Building on the analysis in the previous section, we propose to choose the rotation R by solving

$$\max_{R^\intercal R=I_d}\ \sum_{i=1}^d(RHR^\intercal)_{ii}^2. \tag{8}$$

The solution is straightforward: the optimal R is the orthogonal matrix that diagonalizes H.

Proposition 4.1. *Suppose H has eigendecomposition $H=VDV^\intercal$, where $D=\mathrm{diag}(\nu_1,\ldots,\nu_d)$ is the diagonal matrix of eigenvalues. Then $\sum_{i=1}^d(RHR^\intercal)_{ii}^2$ is maximized when $R=V^\intercal$, in which case*

$$\tilde{I}(\gamma,p_R)\geq\sum_{i=1}^d\nu_i^2.$$

In practice, it is often unnecessary to compute the full eigendecomposition of H. One can instead compute only the leading eigenvectors corresponding to the eigenvalues with largest magnitude. A practical strategy is to select the top r eigenvectors that explain, for example, 95% of the total variance. In particular, if $R^\intercal$ consists of r eigenvectors of H corresponding to eigenvalues $\nu_1,\ldots,\nu_r$, then $\tilde{I}(\gamma,p_R)\geq\sum_{i=1}^r\nu_i^2$.

If R is drawn uniformly at random, the expected $\tilde{I}(\gamma,p_R)$ also admits a lower bound that depends on the spectrum of H.

Corollary 4.2 (Random rotation)**.** *If R is a uniformly random orthogonal matrix, then*

$$\mathbb{E}_R\left[\tilde{I}(\gamma, p_R)\right] \geq \frac{1}{d+2}\big[2\sum_{i=1}^{d}\nu_i^2 + (\sum_{i=1}^{d}\nu_i)^2\big],$$

where the expectation is taken with respect to R drawn uniformly from the orthogonal group.

The proof is provided in Appendix A.3. This suggests that if R is chosen randomly, the expected MFVI improvement deteriorates by a factor of order $1/d$ compared to the PCA-based choice.

In the remainder of this section, we provide some connections with Stein discrepancy as well as other gradient-based PCA methods, and describe how to implement the proposed algorithm in practice.

4.1 Connection with Stein discrepancy

To better understand the role of the matrix $H = \mathbb{E}_\gamma\left[\mathbf{x}(\nabla \log p(\mathbf{x}) + \mathbf{x})^\intercal\right]$ and why its eigenvectors provide a principled choice of rotation, we establish a connection with Stein discrepancy and Stein variational methods.

For a vector-valued test function $g : \mathbb{R}^d \to \mathbb{R}^d$ and a distribution $p(\mathbf{x}) \propto \exp(-U(\mathbf{x}))$, the Stein operator is defined as

$$\mathcal{T}_p g(\mathbf{x}) = \nabla \log p(\mathbf{x})^\intercal g(\mathbf{x}) + \nabla \cdot g(\mathbf{x}).$$

For sufficiently regular g, Stein's identity holds: $\mathbb{E}_p\left[\mathcal{T}_p g(\mathbf{x})\right] = 0$ [23]. The Stein discrepancy between γ and p with respect to a given test function g is

$$\left|\mathbb{E}_\gamma\left[\mathcal{T}_p g(\mathbf{x})\right]\right|,$$

which measures how much Stein's identity is violated under γ, with larger values indicating a greater discrepancy between the two distributions.

For a linear test function $g(\mathbf{x}) = A\mathbf{x}$ with $A \in \mathbb{R}^{d\times d}$, we have

$$\left|\mathbb{E}_\gamma\left[\mathcal{T}_p g(\mathbf{x})\right]\right| = \left|\mathbb{E}_\gamma\left[\nabla \log p(\mathbf{x})^\intercal A\mathbf{x} + \operatorname{tr}(A)\right]\right| = \left|\operatorname{tr}(HA)\right|.$$

Thus, the matrix H fully encodes the Stein discrepancy between γ and p within the class of linear test functions. In particular, for $A = \theta\theta^\intercal$ for a unit vector θ, the Stein discrepancy simplifies to

$$\left|\mathbb{E}_\gamma\left[\mathcal{T}_p g(\mathbf{x})\right]\right| = |\theta^\intercal H \theta|. \tag{9}$$

This suggests that the eigenvectors of $H = \mathbb{E}_\gamma\left[I_d - \nabla^2 U(\mathbf{x})\right]$ identify the directions along which p deviates most from γ, as measured by Stein discrepancy. In particular, eigenvectors with large negative eigenvalues correspond to directions where U exhibits greater average curvature than the Gaussian, indicating that p is more concentrated than γ, whereas large positive eigenvalues correspond to directions where p is flatter. In short, the eigenvectors of H provide the principal directions of discrepancy between p and the Gaussian, while the sign and magnitude of the associated eigenvalues indicate both the nature (concentration vs. flatness) and the severity of the deviation. Therefore, aligning the coordinate system with the eigenvectors of H directly targets the most non-Gaussian directions.

The Stein operator also characterizes the instantaneous rate of change in KL divergence under perturbations of γ. If γ_ε is the law of $\mathbf{x} + \varepsilon g(\mathbf{x})$ for $\mathbf{x} \sim \gamma$, then (see [40, Theorem 3.1])

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathrm{KL}(\gamma_\varepsilon \,\|\, p)\big|_{\varepsilon=0} = -\mathbb{E}_\gamma\left[\mathcal{T}_p g(\mathbf{x})\right].$$

Hence, the perturbation g that maximizes $\mathbb{E}_\gamma\left[\mathcal{T}_p g(\mathbf{x})\right]$ yields the steepest instantaneous decrease of KL divergence. This variational principle forms the basis of the Stein variational gradient descent (SVGD) algorithm [40].

For mean-field approximation, the update is restricted to coordinatewise perturbations. If we further restrict g to linear maps, i.e. $g(\mathbf{x}) = A\mathbf{x}$ with $A = \mathrm{diag}(\mathbf{a})$, then

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathrm{KL}(\gamma_\varepsilon \,\|\, p)\big|_{\varepsilon=0} = -\operatorname{tr}(HA).$$

Optimizing over diagonal A with $\|A\|_{\mathrm{Frob}} \le 1$ yields

$$\max_{A \text{ diag: } \|A\|_{\mathrm{Frob}} \le 1} \operatorname{tr}(HA) = (\sum_{i=1}^{d} H_{ii}^2)^{1/2},$$

achieved when $\mathbf{a} \propto \mathrm{diag}(H)$.

If we apply the coordinatewise linear perturbation to the rotated target p_R, whose corresponding H matrix is $H(p_R) = RHR^\intercal$, the maximal rate of KL decrease becomes

$$\Big(\sum_{i=1}^{d} (RHR^\intercal)_{ii}^2\Big)^{1/2}.$$

This coincides with the objective derived in Theorem 3.4 from the perspective of projected Fisher information. This provides yet another justification for choosing R to be the eigenvectors of of H: it yields the rotation under which coordinatewise linear updates achieve the steepest descent in KL divergence.

4.2 Other gradient-based PCA methods

Using PCA-based techniques to identify important directions for a function or probability distribution is a well-established strategy. For instance, in constructing low-dimensional response surfaces for high-dimensional functions f, the active subspace method [12] identifies directions of greatest variability by computing the principal components of $\mathbb{E}\left[\nabla_{\mathbf{x}} f(\mathbf{x}) \nabla_{\mathbf{x}} f(\mathbf{x})^\intercal\right]$.

In Bayesian analysis, PCA-based dimension reduction has also been widely explored [13, 59]. The certified dimension reduction (CDR) method [59] seeks to approximate a distribution ν by modifying another distribution μ along a low-dimensional subspace. Specifically, CDR considers approximations of the form

$$\nu_{R_r,\ell}(\mathbf{x}) \propto \ell(R_r^\intercal \mathbf{x}) \cdot \mu(\mathbf{x}),$$

where $R_r \in \mathbb{R}^{d\times r}$ is an orthogonal projection onto an r-dimensional subspace and ℓ is a profile function. In the Bayesian setting, ν is the posterior, μ is the prior, and such approximations are effective when the likelihood updates the prior primarily along a low-dimensional subspace.

When μ satisfies a subspace logarithmic Sobolev inequality, this inequality provides an upper bound on the reconstruction error $\inf_\ell \mathrm{KL}(\nu \,\|\, \nu_{R_r,\ell})$. The bound is minimized when the columns of R_r are chosen as the leading eigenvectors of the relative Fisher information matrix

$$\mathrm{FI}(\nu, \mu) = \mathbb{E}_\nu \left[\nabla \log \frac{\nu}{\mu} (\nabla \log \frac{\nu}{\mu})^\intercal \right],$$

which is the central idea behind CDR [59].

In our setting, if we take $\nu = p$ as the target and $\mu = \gamma$ as the standard Gaussian, the matrix $\mathrm{FI}(p, \gamma)$ is not tractable, since evaluating it requires samples from p—the very challenge we aim to overcome. This is also why CDR relies on iterative procedures. A natural alternative is to swap the roles of p and γ, considering instead $\mathrm{FI}(\gamma, p)$, which can be estimated by sampling from γ. However, although the leading eigenvectors of $\mathrm{FI}(\gamma, p)$ identify the subspace that are, in some sense, optimal for modifying p to better approximate γ, these directions are not necessarily well-suited for constructing mean-field approximations.

To illustrate, consider the Gaussian target $p = \mathcal{N}(\mu, \Sigma)$. In this case, the optimal rotation is the one that diagonalizes Σ, making p_R a product measure. However, for this target $\mathrm{FI}(\gamma, p) = (\Sigma^{-1} - I_d)^2 + \Sigma^{-1}\mu\mu^\intercal\Sigma^{-1}$, whose eigenvectors generally do not align with those of Σ. Even when $\mu = 0$, where $\mathrm{FI}(\gamma, p) = (\Sigma^{-1} - I_d)^2$, the eigenvectors of $\mathrm{FI}(\gamma, p)$ may still fail to recover those of Σ. For example, if $\Sigma^{-1} = \begin{pmatrix} 1 & \varepsilon \\ \varepsilon & 1 \end{pmatrix}$, then $\mathrm{FI}(\gamma, p) = \varepsilon^2 I_d$, whose eigenvectors are not identifiable. This occurs because the two eigenvalues $\pm\varepsilon$ of $\Sigma^{-1} - I_d$ collapse after squaring, erasing the directional information.

By contrast, for the Gaussian target, the eigenvectors of $H = I_d - \Sigma^{-1}$ always diagonalize Σ, yielding the correct rotation for mean-field approximation. These examples illustrate the importance of the rotation choice in our algorithm: rotations that are optimal for other objectives may not necessarily be suitable for mean-field approximation.

4.3 Implementation

In practice, the matrix H can be estimated via Monte Carlo

$$\widehat{H} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \Big(\nabla \log p(\mathbf{x}_i) + \mathbf{x}_i \Big)^\intercal, \quad \mathbf{x}_i \overset{iid}{\sim} \gamma, \quad 1 \le i \le N,$$

and then symmetrized as $(\widehat{H} + \widehat{H}^\intercal)/2$. An alternative unbiased estimator is $\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \nabla \log p(\mathbf{x}_i)^\intercal + I_d$, but the form given above may yield lower variance when p is close to γ, since in that regime $\nabla \log p(\mathbf{x}) + \mathbf{x}$ is close to zero.

Rather than computing the full eigendecomposition of $\widehat{H}$, it is often sufficient to compute only the leading r eigenvectors associated with the largest eigenvalues in magnitude. These eigenvectors then form the first r columns of $R^\intercal$. We can encode such an orthogonal matrix R efficiently as a product of r Householder reflections $I_d - 2w_j w_j^\intercal$ for unit vectors w_j, $1 \le j \le r$. This representation requires only $O(rd)$ storage and $O(rd)$ computation to apply R or $R^\intercal$ to a vector in $\mathbb{R}^d$.

Because H is the cross-covariance matrix between $\mathbf{x} \sim \gamma$ and the relative score $\nabla \log p(\mathbf{x}) + \mathbf{x}$, we term this approach relative score PCA. A summary of the procedure is given in Algorithm 2.

Algorithm 2 Relative score PCA

Input: Target distribution p; number of PCs r; Monte Carlo sample size N

Generate $\mathbf{x}_i \overset{iid}{\sim} \mathcal{N}(0, I_d)$ for $1 \le i \le N$

Compute $\widehat{H} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i(\nabla \log p(\mathbf{x}_i) + \mathbf{x}_i)^\intercal$ and symmetrize as $\widehat{H} = (\widehat{H} + \widehat{H}^\intercal)/2$

return Top r eigenvectors of $\widehat{H}$ corresponding to the largest eigenvalues in magnitude.

5 Iterative Gaussianization

We have discussed how to perform MFVI in a rotated coordinate system, which yields a rotation and a coordinatewise map that transforms the target closer to the standard Gaussian. This procedure can be naturally repeated to progressively pushes the target distribution toward Gaussian.

Specifically, let $p = p^{(0)}$ denote the target distribution. After $k-1$ iterations ($k \ge 1$), suppose the transformed target distribution is $p^{(k-1)}$. In the k-th iteration, we first choose an orthogonal matrix $R_k \in \mathbb{R}^{d\times d}$ (e.g., via relative score PCA) and define the rotated target $p^{(k-1)}_{R_k} = R_k \# p^{(k-1)}$. Next, we solve the MFVI problem to find the optimal coordinatewise transformation

$$F_k^* = \operatorname*{argmin}_{F\in\mathcal{F}} \ \mathrm{KL}(F\#\gamma \,\|\, p^{(k-1)}_{R_k}).$$

Finally, we apply the inverse transformation to Gaussianize the rotated target, yielding the new target

$$p^{(k)} = F_k^{*-1} \# p^{(k-1)}_{R_k}.$$

The iterative Gaussianization procedure is summarized in Algorithm 3.

Algorithm 3 Iterative Gaussianization

Input: Target distribution p; number of iterations K

Initialize $p^{(0)} = p$

for $k = 1$ to K **do**

Select orthogonal matrix R_k

Define rotated target $p^{(k-1)}_{R_k} = R_k \# p^{(k-1)}$

Solve $F_k^* = \operatorname*{argmin}_{F\in\mathcal{F}} \mathrm{KL}(F\#\gamma \,\|\, p^{(k-1)}_{R_k})$

Update $p^{(k)} = F_k^{*-1} \# p^{(k-1)}_{R_k}$

end for

return Gaussianization transformation $F_K^{*-1} \circ R_K \circ \cdots \circ F_1^{*-1} \circ R_1$

After k iterations, the cumulative transformation of the original target leads to

$$p^{(k)} = (F_k^{*-1} \circ R_k \circ \cdots \circ F_1^{*-1} \circ R_1)\# p.$$

Equivalently, the inverse of the sequence of transformations pushes the standard Gaussian toward the target distribution, resulting in the approximation

$$q^{(k)} = (R_1^\intercal \circ F_1^* \circ \ldots \circ R_k^\intercal \circ F_k^*)\#\gamma. \tag{10}$$

The Jacobian determinant of this transformation is the product of the determinants of each coordinatewise map F_i^*, since rotations have unit determinant. Therefore, the density of $q^{(k)}$ can be evaluated efficiently, which can be used in a subsequent step to correct for the bias in $q^{(k)}$ through importance sampling or MCMC.

By construction, each iteration brings the target distribution closer to the standard Gaussian in KL divergence, regardless of how R_k is chosen. We have the following proposition.

Proposition 5.1. *The KL divergence is non-increasing in k:*

$$\begin{aligned}&\mathrm{KL}(\gamma \,\|\, p^{(k)}) \le \mathrm{KL}(\gamma \,\|\, p^{(k-1)}),\\ &\mathrm{KL}(q^{(k)} \,\|\, p) \le \mathrm{KL}(q^{(k-1)} \,\|\, p), \quad \forall\, k \ge 1.\end{aligned}$$

Moreover, Lemma 3.1 and Theorem 3.2 together imply that if $p^{(k-1)} \neq \gamma$, then there exists a rotation R_k such that $\tilde{I}(\gamma, R_k \# p^{(k-1)}) > 0$, thus $\mathrm{KL}(\gamma \,\|\, p^{(k)}) < \mathrm{KL}(\gamma \,\|\, p^{(k-1)})$. In other words, if the target is not yet standard Gaussian, then strict improvement is possible by choosing an appropriate rotation.

5.1 Stationary guarantee

Now suppose we draw the orthogonal matrices at random from a distribution with full support (e.g., uniform), and define the average MFVI improvement as

$$\overline{\Delta}_{\mathrm{MFVI}}(p) = \mathbb{E}_R \left[\Delta_{\mathrm{MFVI}}(R \# p)\right].$$

This functional is divergence-like: it is always nonnegative, and it vanishes if and only if $p = \gamma$. Indeed, if $\overline{\Delta}_{\mathrm{MFVI}}(p) = 0$, then $\Delta_{\mathrm{MFVI}}(R \# p) = 0$ for almost every R, which implies $\tilde{I}(\gamma, R \# p) = 0$ for almost every R. By Theorem 3.2, this forces $p = \gamma$.

The algorithm implies $\Delta_{\mathrm{MFVI}}(R_k \# p^{(k-1)}) = \mathrm{KL}(\gamma \,\|\, p^{(k-1)}) - \mathrm{KL}(\gamma \,\|\, p^{(k)})$. Summing over $k = 1, \ldots, K$,

$$\sum_{k=1}^{K} \Delta_{\mathrm{MFVI}}(R_k \# p^{(k-1)}) = \mathrm{KL}(\gamma \,\|\, p^{(0)}) - \mathrm{KL}(\gamma \,\|\, p^{(K)}) \le \mathrm{KL}(\gamma \,\|\, p).$$

Therefore, $\sum_{k=1}^{K} \overline{\Delta}_{\mathrm{MFVI}}(p^{(k-1)}) \le \mathrm{KL}(\gamma \,\|\, p)$, which implies that $\overline{\Delta}_{\mathrm{MFVI}}(p^{(K)}) \to 0$ almost surely as $K \to \infty$, and $\min_{1 \le k \le K} \overline{\Delta}_{\mathrm{MFVI}}(p^{(k-1)}) = O(\frac{1}{K})$. As $\overline{\Delta}_{\mathrm{MFVI}}(p) = 0$ if and only if $p = \gamma$, the above can be understood as a first-order stationary guarantee of the iterative algorithm [3].

5.2 Convergence for Gaussian target

When the target distribution is multivariate Gaussian, $p = \mathcal{N}(0, \Sigma)$, the behavior of our algorithm can be analyzed more explicitly. In particular, we can characterize how quickly the KL divergence decreases at each iteration. Here, we consider rotation matrices that are drawn uniformly at random. In the context of Gaussianization for density estimation, random rotations have previously been proposed by [37] as a computationally efficient alternative to ICA and as a way to escape local minima. More recently, Draxler et al. [15] studied the convergence rate of Gaussianization with random rotations in the density estimation setting, where the objective function is the forward KL divergence $\mathrm{KL}(p \,\|\, \gamma)$, which is different from the reverse KL in our sampling setting.

In Algorithm 3, after each iteration the target is transformed so that its MF approximation is the standard Gaussian γ. Therefore, we assume that the MF approximation of $p = \mathcal{N}(0, \Sigma)$ is already γ. Given a randomly sampled rotation R, the KL divergence after MFVI is decreased from $\mathrm{KL}(\gamma \,\|\, \mathcal{N}(0, \Sigma))$ to $\inf_{q \in \mathcal{P}_{\mathrm{prod}}} \mathrm{KL}(q \,\|\, \mathcal{N}(0, R\Sigma R^\intercal))$.

The following theorem shows that, in expectation over random rotations, the KL divergence contracts by a multiplicative factor that depends on the condition number of Σ and the dimension d.

Theorem 5.1. *Let $\kappa = \lambda_{\max}(\Sigma)/\lambda_{\min}(\Sigma)$ denote the condition number of Σ. Then*

$$\mathbb{E}_R \left[\inf_{q \in \mathcal{P}_{\mathrm{prod}}} \mathrm{KL}(q \,\|\, \mathcal{N}(0, R\Sigma R^\intercal)) \right] \leq \left(1 - \frac{2}{(d+2)\kappa^2}\right) \cdot \mathrm{KL}(\gamma \,\|\, \mathcal{N}(0, \Sigma)),$$

where the expectation is taken with respect to the uniform distribution over rotation matrices R.

The proof is given in Appendix A.4.

Note that in the contraction factor $(1 - \frac{2}{(d+2)\kappa^2})$, κ is the condition number of the target distribution at the current iteration. As the transformed target approaches γ, the condition number converges to 1, and the contraction factor approaches $(1 - \frac{2}{d+2})$. Moreover, the theorem shows that the number of iterations required to reduce the KL divergence below a fixed threshold grows linearly with the dimension d, which coincides with the rate proved in [15] for the forward KL setting.

Figure 1 provides a numerical demonstration. In each plot, the y-axis is the number of iterations required in order for the KL divergence (averaged over 30 independent replicates) to fall below 0.01. The left two panels vary the dimension d, while the right two panels vary the condition number κ. The first and third panels correspond to random rotations, while the second and fourth panels correspond to the relative score PCA in Algorithm 2, where the top principal components explaining more than 50% of the variance are selected.

These results confirm that with random rotations, the number of iterations required scales linearly with d, but sublinearly with κ. In contrast, the PCA-based rotation strategy requires far fewer iterations, growing much more slowly with dimension and remaining nearly constant in condition number. This highlights the advantage of using relative score PCA over random rotations for selecting rotations.

5.3 Related work

We discuss some related work on Gaussianization for density estimation and normalizing flows.

Gaussianization for density estimation In density estimation, Gaussianization methods iteratively transform data to become more Gaussian. Each iteration alternates between rotations, which reduce dependence among coordinates, and marginal Gaussianization, which transforms each marginal to a standard Gaussian. In its original form [11], rotations are determined by independent component analysis (ICA), which does not have closed-form solutions and can be slow to compute in high dimensions. An alternative is to choose the rotation matrices by PCA, but the iterative algorithm may get stuck in local optima [37]. Laparra et al. [37] propose to use random rotations, which makes each iteration faster to compute, but may require more iterations to converge. From the perspective of continuous-time particle flows, a related density estimator is developed by [54] and later extended by [53], which constructs the Gaussianization map by alternating between random rotations and simple transformations that are either coordinatewise maps or radial expansions.

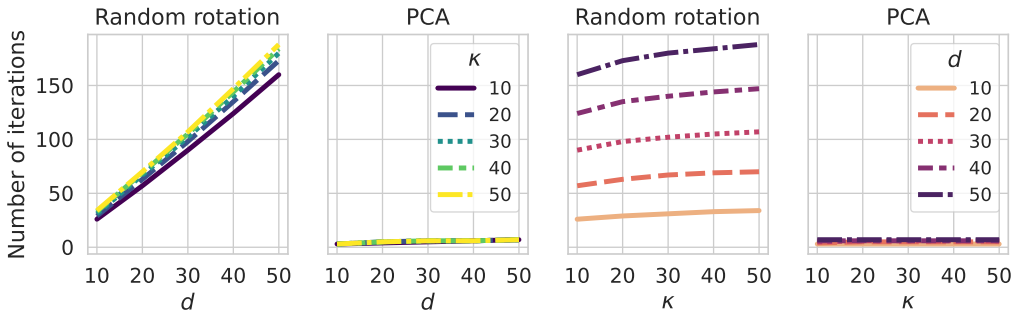

Figure 1: Number of iterations required for the KL divergence to fall below 0.01. The left two panels vary the dimension d, and the right two panels vary the condition number κ. The first and third panels correspond to random rotations, while the second and fourth panels use the PCA-based rotation. Results are averaged over 30 independent replicates.

Gaussianization flow [43] treats the rotation matrices as trainable parameters, which are parametrized as products of Householder transformations. The flow is trained as a usual normalizing flow, where the rotations and componentwise transformations are optimized jointly. The authors show that the resulting Gaussianization flow is a universal approximator given a sufficient number of layers.

Normalizing flows Normalizing flows define the variational family as the pushforward of a simple reference distribution, typically $\gamma = \mathcal{N}(0, I_d)$, through a diffeomorphism τ_θ parametrized by $\theta \in \Theta$. The pushforward measure is denoted $q_\theta := \tau_\theta \# \gamma$. By the change of variable formula, the density of q_θ is given by

$$q_\theta(\mathbf{x}) = \gamma(\tau_\theta^{-1}(\mathbf{x})) \cdot |\nabla \tau_\theta^{-1}(\mathbf{x})|,$$

where τ_θ^{-1} is the inverse of the transport map τ_θ and $|\nabla \tau_\theta^{-1}(\mathbf{x})|$ is the absolute value of the determinant of the Jacobian of τ_θ^{-1} at $\mathbf{x}$. The KL divergence between q_θ and p can be expressed as

$$\mathrm{KL}(q_\theta \,\|\, p) = \mathbb{E}_\gamma \left[\log \gamma(\mathbf{x}) - \log |\nabla \tau_\theta(\mathbf{x})| - \log p(\tau_\theta(\mathbf{x})) \right].$$

In practice, the KL divergence is optimized via gradient-based optimization, where the gradient is estimated by Monte Carlo samples from γ.

Because evaluating the log-determinant term $\log |\nabla \tau_\theta|$ can be computationally expensive, transport maps are often parameterized so that the Jacobian is lower triangular. For example, polynomial basis expansions have been used to directly approximate the Knothe-Rosenblatt rearrangement [18], but the number of parameters grows rapidly with dimension. Normalizing flows construct the map as a composition of simple transformations, each with a triangular Jacobian. Popular examples include autoregressive flows [33], RealNVP [14], and neural spline flows [17]; see [47] for a survey. However, the structural constraints of these flows limit their expressiveness, often requiring many layers to capture complex distributions. Overparameterization can also make training unstable, and empirical studies show that very large Monte Carlo sample sizes are needed for reliable optimization via stochastic gradient descent [1, 8].

6 Numerical results

We evaluate the effectiveness of the proposed algorithm on several posterior sampling tasks. In the first three examples, we compare MFVI along principal component axes against standard MFVI. In the final two examples, we compare the iterative Gaussianization algorithm against normalizing flows. Code for reproducing the experiments is available at https://github.com/liusf15/iterative-gaussianization.git.

For MFVI, the componentwise transformations are parametrized by rational quadratic splines [17] with 10 knots, with range boundaries set to $(-8, 8)$. When selecting rotations using relative score PCA (Algorithm 2), we use a Monte Carlo sample size of 1000, and retain the top principal components that explain 95% of the variance unless otherwise noted. Optimization of variational objectives is carried out with Adam [32], using a learning rate of 0.01 and a fixed Monte Carlo sample of size 1000. For evaluation, we generate 2000 samples from the learned variational distribution to compute metrics such as maximum mean discrepancy (MMD), kernelized Stein discrepancy (KSD), effective sample size (ESS), and evidence lower bound (ELBO). When reference samples from the target distribution are needed, we generate 2000 samples using the No-U-Turn Sampler (NUTS; [27, 9]) with 20 chains, 25,000 burnin iterations and 50,000 sampling iterations with thinning, unless otherwise noted. To account for randomness, each experiment is repeated 20 times independently, with random initialization and random training/evaluation samples.

Since the performance of reverse KL-based variational inference is sensitive to initialization, we employ a Laplace approximation to obtain a good starting point. Specifically, we find the minimum $\mathbf{x}^*$ of the potential function $U(\mathbf{x}) = -\log p(\mathbf{x})$, and the corresponding inverse Hessian matrix $C^* = (\nabla^2 U(\mathbf{x}^*))^{-1}$. We then shift the target distribution by $\mathbf{x}^*$ and rescale each coordinate by $(C^*_{i,i})^{1/2}$, thereby centering the target near the origin and approximately normalizing the marginal variance. A similar initialization strategy was used in [1].

6.1 Bayesian logistic regression

The Bayesian logistic regression model is defined as

$$\begin{aligned} &\beta \sim \mathcal{N}(0, \sigma^2 I_d), \\ &y_i \mid x_i, \beta \sim \text{Bernoulli}(S(x_i^\intercal \beta)), \quad 1 \le i \le n, \end{aligned}$$

where $S(x) = (1 + e^{-x})^{-1}$ is the sigmoid function, $x_i \in \mathbb{R}^d$ represents the covariates, and $y_i \in \{0, 1\}$ represents the binary response. We take $n = 20$, $d = 10$, and fix $\sigma = 2$. The covariates are generated as $x_i \overset{iid}{\sim} \mathcal{N}(0, \Sigma_X)$, where $\Sigma_X = UDU^\intercal$, U is a random orthogonal matrix, and D is diagonal with entries equally spaced on a log scale from 10^{-1} to 10^1. The responses y_i are drawn i.i.d. from Bernoulli(0.5).

Effectiveness of relative score PCA We compare three choices of the rotation R used in MFVI:

- $R = I_d$: No rotation is performed, corresponding to the standard MFVI.
- R is a uniformly random orthogonal matrix.
- R is formed by the principal components explaining 95% of the variance (Algorithm 2).

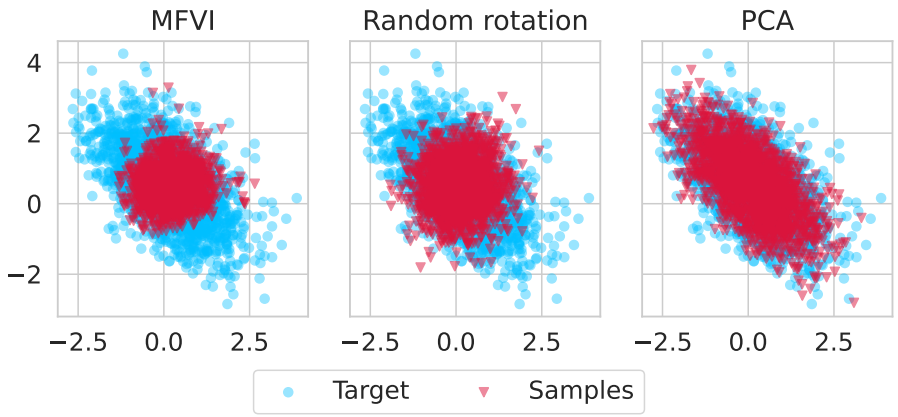

Figure 2: MFVI with different rotation choices in the Bayesian logistic regression example. Left: $R = I_d$ (no rotation). Middle: R is a random orthogonal matrix. Right: R chosen by the proposed relative score PCA method.

Figure 2 visualizes the learned variational distribution against the target distribution. The blue dots represent reference samples drawn from the target using NUTS, with 1000 iterations for burnin and 10,000 iterations for sampling, and thinning every 5 iterations. The red dots represent 2000 samples drawn from the learned variational distribution. The left and middle panels show that standard MFVI or MFVI in a random basis severely underestimates posterior variance. In contrast, the right panel shows that MFVI with PCA-based rotation yields a much closer approximation to the true posterior. This suggests that relative score PCA is able to identify important directions in the target distribution, and applying MFVI along these directions leads to a better approximation.

Iterative algorithm We next demonstrate the iterative Gaussianization algorithm with randomly sampled rotations. We denote the method by Rk, where k is the number of iterations. These are compared against one-step MFVI ($R = I_d$) and PCA-based MFVI as described above. We compute the maximum mean discrepancy (MMD) and the negative evidence lower bound ($-$ELBO) between the learned variational distribution and the reference samples.

Figure 3 summarizes the results, with error bars showing variation across 20 independent runs. As the number of iterations increases, MMD decreases and ELBO improves, confirming that iterative Gaussianization progressively improves the approximation. However, even after 11 iterations, the performance is still worse than the one-step PCA-based MFVI. This indicates that while random rotations yield gradual improvements, selecting rotations via relative score PCA is substantially more effective.

6.2 Posteriordb benchmarks

We further evaluate our method on posterior distributions from `posteriordb` [42], a repository of benchmark Bayesian models. We consider 12 low-dimensional targets with dimensions ranging from 2 to 8, and compare standard MFVI against PCA-based MFVI (using the top PCs that explain 95% of the variance).

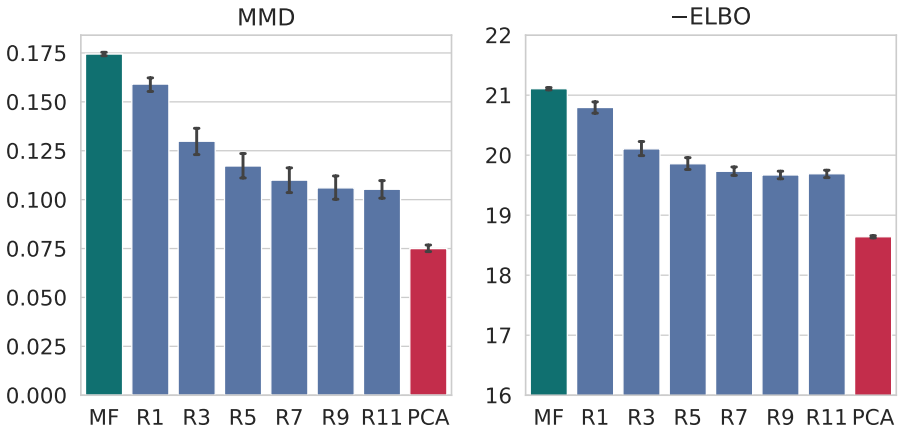

Figure 3: Comparison of MMD (left panel) and negative ELBO (right panel) on the Bayesian logistic regression example. The first bar (green) represents standard MFVI while the last one (red) is the PCA-based MFVI. The ones in between (blue) represent iterative Gaussianization with random rotations, with the number of iterations indicated on the x-axis.

For each target, we report the MMD, KSD, ELBO, and ESS of the learned variational distribution. The MMD is computed with 2000 reference samples drawn from NUTS. We report the mean and standard deviation (in parentheses) of each metric over 20 replicates. The results are shown in Table 1.

We observe that PCA-based MFVI greatly outperforms standard MFVI most of the time. For the target `hmm`, standard MFVI achieves a smaller MMD, but PCA-based MFVI achieves a smaller KSD and a much larger ESS. For the target `gp-regr`, standard MFVI is slightly better in all metrics, indicating that the original coordinate system is already a good choice for MFVI. For all other targets, PCA-based MFVI outperforms standard MFVI, sometimes by a large margin, in all metrics. Given the small computational overhead of PCA, performing MFVI in the PCA-based coordinate system is a simple yet powerful enhancement to standard MFVI.

6.3 Poisson generalized linear mixed model

Generalized linear mixed models (GLMMs) are widely used to model grouped data, yet full Bayesian inference for such models can be computationally challenging due to the high-dimensional random effects and strong dependencies between global and local parameters [19]. Appropriate model reparametrizations can markedly improve both MCMC mixing [48] and variational inference accuracy [55]. We consider the Poisson GLMM studied in [55, Section 8.1]:

$$\begin{aligned}
&\beta_0, \beta_1 \sim \mathcal{N}(0, 100), \\
&\sigma^2 \sim \text{InvGamma}(0.5, 0.5), \quad b_i \sim \mathcal{N}(0, \sigma^2), \quad 1 \le i \le n, \\
&y_{ij} \sim \text{Poisson}(e^{\eta_{ij}}), \quad \eta_{ij} = \beta_0 + \beta_1 x_{ij} + b_i, \quad 1 \le j \le m.
\end{aligned}$$

The global parameters are denoted by $\boldsymbol{\theta}_G = (\sigma, \beta_0, \beta_1)$, and the local random effects by b_i $(1 \le i \le n)$. Data are generated as in [55] with $n = 500$ and $m = 7$, giving a total dimension of 503. Following [55],

	MMD		KSD	
	MF	PCA	MF	PCA
M0	0.065 (0.005)	**0.014** (0.014)	1.269 (0.078)	**0.409** (0.154)
arK	0.241 (0.002)	**0.087** (0.004)	20.983 (2.837)	**10.557** (1.075)
garch	0.218 (0.005)	**0.146** (0.013)	1.611 (0.088)	**0.828** (0.104)
gp-regr	**0.010** (0.010)	0.015 (0.014)	**0.272** (0.079)	0.286 (0.087)
hmm	**0.016** (0.013)	0.036 (0.008)	2.962 (0.972)	**1.875** (0.785)
kidscore-interaction	0.399 (0.010)	**0.032** (0.012)	71.362 (5.679)	**50.733** (5.515)
mesquite	0.270 (0.002)	**0.092** (0.005)	5.212 (0.953)	**4.671** (0.809)
nes-logit	0.339 (0.004)	**0.015** (0.016)	15.934 (0.700)	**2.199** (1.263)
normal-mixture	0.045 (0.006)	**0.024** (0.010)	13.926 (0.752)	**4.255** (1.738)
radon	0.055 (0.004)	**0.013** (0.010)	22.615 (1.702)	**9.175** (2.992)
sesame	0.151 (0.004)	**0.018** (0.016)	9.845 (0.609)	**2.149** (0.616)
wells	0.214 (0.004)	**0.039** (0.006)	18.410 (0.884)	**2.971** (0.715)

	ELBO		ESS	
	MF	PCA	MF	PCA
M0	-243.8 (0.0)	**-243.7** (0.0)	1455.5 (270.9)	**1941.7** (35.0)
arK	88.3 (0.1)	**92.3** (0.1)	12.9 (11.5)	**257.4** (127.4)
garch	-448.4 (0.0)	**-447.8** (0.0)	106.3 (94.8)	**422.8** (230.9)
gp-regr	**-23.5** (0.0)	**-23.5** (0.0)	**1931.8** (15.9)	1874.7 (150.1)
hmm	-167.6 (0.0)	**-166.8** (0.0)	158.0 (122.5)	**1501.5** (137.9)
kidscore-interaction	-1888.3 (1.1)	**-1884.3** (1.5)	**9.1** (7.1)	7.5 (4.7)
mesquite	-317.6 (0.1)	**-311.2** (0.2)	7.5 (7.8)	**62.6** (32.8)
nes-logit	-781.8 (0.0)	**-780.7** (0.0)	90.3 (97.3)	**1630.2** (33.8)
normal-mixture	-2098.9 (0.0)	**-2098.7** (0.0)	680.9 (322.6)	**1866.3** (25.9)
radon	-18562.2 (0.0)	**-18562.1** (0.0)	1551.8 (200.4)	**1939.4** (15.4)
sesame	-107.1 (0.0)	**-106.6** (0.0)	330.6 (166.0)	**1890.0** (83.5)
wells	-1954.1 (0.0)	**-1952.6** (0.0)	56.0 (48.3)	**1610.4** (42.4)

Table 1: Average performance metrics for `posteriordb` experiments, with standard deviations over 20 independent replicates shown in parentheses.

the local effects are reparametrized as $\tilde{b}_i = (b_i - \lambda_i)/L_i$, where λ_i, L_i depend on $\boldsymbol{\theta}_G$ and are obtained from a Gaussian approximation of the conditional distribution of $b_i \mid \boldsymbol{\theta}_G, \mathbf{y}_i$; see Appendix B.1 for details.

For the reparametrized posterior, Tan [55] considers a Gaussian variational approximation of the form $q_G(\boldsymbol{\theta}_G) \cdot \prod_{i=1}^{n} q_{\tilde{b}_i}(\tilde{b}_i)$, where q_G is a full-covariance Gaussian and each $q_{\tilde{b}_i}$ is a univariate Gaussian. We apply our rotated variational inference to the same reparametrized target. To keep the number of parameters comparable and demonstrate the effectiveness of relative score PCA, we approximate the rotated target with a product Gaussian distribution.

Figure 4 shows the posterior distributions of the global parameters $(\sigma, \beta_0, \beta_1)$, and Table 2 reports the posterior means and standard deviations averaged over 20 independent replicates. The two VI methods, Gaussian VI and MFVI+PCA, are compared against MCMC. For the regression coefficients β_0 and β_1, both VI methods match the MCMC posteriors reasonably well. For the variance component σ, however, Gaussian VI severely underestimates the right tail of the posterior, leading to a biased posterior mean and underestimated variance. MFVI+PCA provides a noticeably better fit to the posterior of σ, with more accurate mean and variance estimates. It also yields a smaller average MSE in estimating the posterior means and standard deviations of the local effects b_i (Table 3). In this Poisson GLMM example, the reparametrization does not fully eliminate the dependence between the global parameters $\boldsymbol{\theta}_G$ and the local effects $\tilde{b}_i$'s. Unlike Gaussian VI, which ignores this dependence, the proposed method exploits score information to construct rotations that better capture the global-local dependence structure.

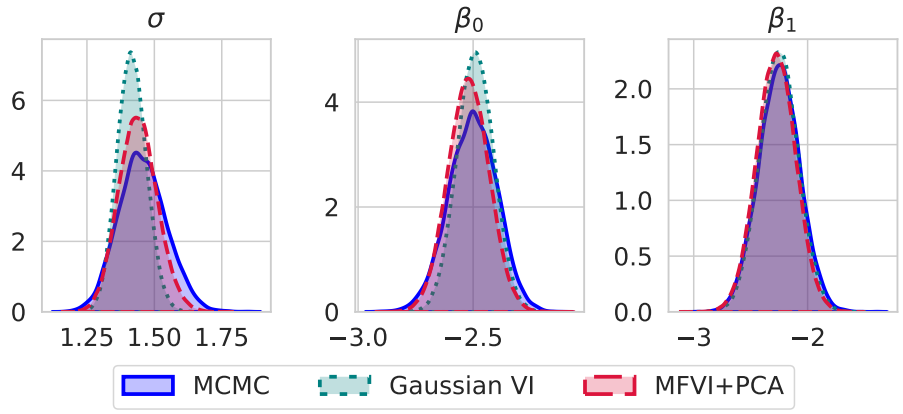

Figure 4: Posterior distributions of the global parameters in the Poisson GLMM example.

6.4 Sparse logistic regression

We consider the sparse logistic regression model given by

$$\begin{aligned} &\tau \sim \text{Gamma}(0.5,\, 0.5), \\ &\lambda_j \overset{iid}{\sim} \text{Gamma}(0.5,\, 0.5), \quad \beta_j \overset{iid}{\sim} \mathcal{N}(0,1), \quad 1 \le j \le d, \\ &y_i \sim \text{Bernoulli}(S(\tau \cdot \mathbf{x}_i^{\intercal}(\boldsymbol{\beta} \circ \boldsymbol{\lambda}))), \quad 1 \le i \le n. \end{aligned}$$

	Gaussian VI	MFVI+PCA	MCMC
σ	1.42 ± 0.05	1.44 ± 0.07	1.46 ± 0.09
β_0	-2.49 ± 0.08	-2.52 ± 0.09	-2.50 ± 0.10
β_1	-2.25 ± 0.17	-2.28 ± 0.17	-2.25 ± 0.18

Table 2: Posterior mean and posterior standard deviations for the global parameters in Poisson GLMM.

	Gaussian VI	MFVI+PCA
Mean	0.0007	0.0004
Standard deviation	0.0043	0.0032

Table 3: MSE in estimating the posterior mean and posterior standard deviations for the local effects b_i in Poisson GLMM, averaged over $1 \leq i \leq n$.

Here, $\tau \in \mathbb{R}_+$ is the global scale parameter, $\boldsymbol{\lambda} \in \mathbb{R}_+^d$ is the per-dimension scale, and $\boldsymbol{\beta} \in \mathbb{R}^d$ is the regression coefficient. We fit the model on the German credit dataset [28], which has 25 numeric covariates including an intercept. The dimension of the posterior distribution of $(\tau, \boldsymbol{\lambda}, \boldsymbol{\beta})$ is 51.

We compare the proposed method with the neural spline flow (NSF) [17], a popular normalizing flow model. To ensure fair comparison, we match the number of layers in NSF with the number of iterations in our method, and use the same number of spline knots in both. Each neural network in NSF has 1 hidden layer of size 5, resulting in about five times as many parameters as our method. Both methods are trained with 1000 samples using Adam with learning rate 0.01. NSF is optimized for 200 iterations, while our method uses 100 iterations per layer, keeping the wall-clock times on the same magnitude.

We report the KSD, MMD, ELBO, and the mean squared error (MSE) for estimating the posterior mean and variance. We do not report ESS, as the variance in its estimation is too large to be meaningful. All metrics are computed using 2000 samples from the learned variational distribution and 2000 reference samples drawn from NUTS.

The results are shown in Figure 5, with error bars indicating variation across 20 independent runs. We vary the number of layers/iterations over $2, 4, 6$ on the x-axis. We observe that the proposed method significantly outperforms NSF in terms of KSD, MMD, and MSE for estimating first and second moments, while requiring less computation time. The gap is especially pronounced when the number of layers is small.

One could potentially improve the performance of NSF by increasing the number of layers or enlarging the capacity of the neural networks. However, training a more complex model comes with higher computational cost and requires careful hyperparameter tuning. Furthermore, any change in the architecture requires retraining the entire model. In contrast, the proposed method consists only of solving multiple MFVI problems, which are much easier to optimize and require minimal tuning. Moreover, if the number of iterations is deemed insufficient, additional iterations can be added seamlessly without retraining the earlier ones.

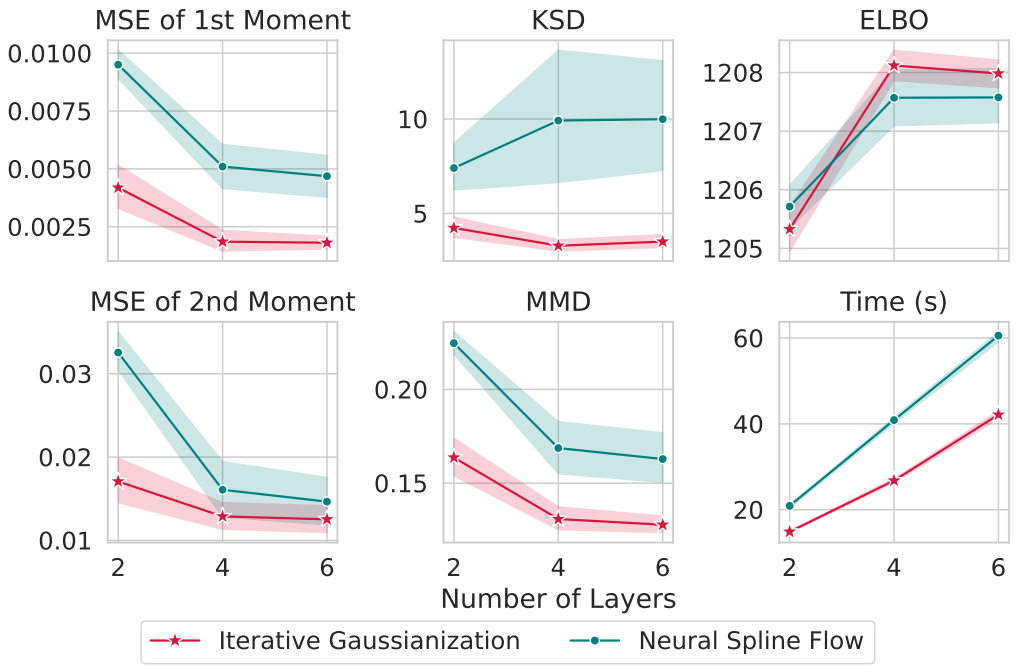

Figure 5: Average performance metrics for the sparse logistic regression experiment, with error bars showing variation across 20 independent replicates.

6.5 Item response theory model

Item response theory (IRT) is a statistical framework for analyzing responses to test items, such as exam questions or survey questionnaires. The model assumes that the probability of a correct response depends on the latent ability of the individual, the difficulty of the item, and the discrimination parameter of the item. We consider the logistic-link IRT model as defined in [52]. Specifically, for individual $i \in [I]$ and item $j \in [J]$, the correctness of the response y_{ij} is modeled as

$$y_{ij} \mid \theta_i, a_j, b_j \sim \text{Bernoulli}(S(a_j(\theta_i - b_j))),$$

where θ_i is the ability of individual i, a_j is the discrimination parameter of item j, and b_j is the difficulty parameter of item j. The prior distributions are given by

$$\begin{aligned}
a_j &\sim \text{LogNormal}(0, \sigma_a^2), \quad j \in [J], \\
\sigma_a &\sim \text{LogNormal}(0, 2^2), \\
b_j &\sim \mathcal{N}(\mu_b, \sigma_b^2) \quad j \in [J], \\
\mu_b &\sim \mathcal{N}(0, 5^2), \quad \sigma_b \sim \text{LogNormal}(0, 2^2), \\
\theta_i &\sim \mathcal{N}(0, 1), \quad i \in [I].
\end{aligned}$$

	MMD (PC-1)	MMD (PC-2)	MMD (PC-142)	MMD (PC-143)
NSF	0.799 (0.023)	0.853 (0.021)	0.224 (0.022)	0.326 (0.024)
Gaussianization	**0.312** (0.021)	**0.310** (0.035)	**0.180** (0.018)	**0.115** (0.047)
	W_2 (PC-1)	W_2 (PC-2)	W_2 (PC-142)	W_2 (PC-143)
NSF	2.022 (0.073)	2.120 (0.064)	0.096 (0.026)	0.069 (0.011)
Gaussianization	**0.714** (0.032)	**0.667** (0.057)	**0.042** (0.005)	**0.016** (0.006)

Table 4: Results for the item response theory model. Sliced MMD and sliced W_2 in the principal component directions are reported, with standard deviation in parentheses.

We use the benchmark dataset from Stan[1] with $I = 20$ and $J = 100$, resulting in a posterior distribution of dimension 143.

Due to the high dimensionality of the target distribution, we use the sliced MMD and sliced Wasserstein-2 distance (W_2) to evaluate the performance of the learned variational distribution. We consider the first two principal components (PC-1, PC-2) of the reference samples, as well as the last two principal components (PC-142, PC-143). These PCs correspond to the directions where the target distribution has the largest and smallest variance, respectively. We compute the sliced MMD and sliced W_2 along these directions, using 2000 samples from the learned variational distribution and 2000 reference samples.

The NSF uses the same architecture as in Section 6.4, with 6 layers, and is trained with Adam (learning rate 0.01) for 1000 iterations. The proposed method uses 4 iterations with PCA-based rotations that include all principal components. Each MFVI problem is optimized with Adam (learning rate 0.01) for 200 iterations.

The results are shown in Table 4. The numbers in the parentheses represent the standard deviation across 20 independent replicates. We observe that the proposed method consistently outperforms NSF across all metrics, demonstrating its ability to approximate the target distribution more accurately along both high-variance and low-variance directions.

7 Conclusions

We proposed an algorithm for sampling from unnormalized density by iteratively performing MFVI in carefully rotated coordinate systems. The rotations are designed to make the coordinates least dependent and most "non-Gaussian" in order for MFVI to bring the maximal improvement. The rotation is chosen through a PCA procedure based on the relative scores of the target distribution, which is fast to compute, cheap to store, and allows the method to capture complex distributions with minimal parametrization. This provides a computationally efficient alternative to overparametrized normalizing flows. We analyzed the algorithm through projected Fisher information, established entropy contraction rate for Gaussian targets, and demonstrated its effectiveness on posterior sampling tasks.

In practice, the transport map obtained by the algorithm is approximate, meaning that the approximation $q^{(k)}$ defined in Equation (10) does not exactly match the target. The quality of the approximation can be evaluated using diagnostics such as effective sample size, since the density

[1] https://github.com/stan-dev/stat_comp_benchmarks/tree/master/benchmarks/irt_2pl

of $q^{(k)}$ is available in closed form. If the approximation is insufficient, additional iterations can be added without modifying earlier ones. When estimating posterior expectations, variance reduction can be achieved with quasi-Monte Carlo methods [41]. Any residual bias can be corrected, either by using $q^{(k)}$ as a proposal in importance sampling or by applying Metropolis-Hastings corrections in MCMC [22]. The Gaussianization transformation can also serve as a pre-conditioning step to make the geometry of the target more amenable to MCMC [49, 26].

Acknowledgements

We thank Bob Carpenter, David Dunson, Aram-Alexandre Pooladian, Lawrence Saul, and Surya Tokdar for their valuable discussions and feedback.

A Proofs

A.1 Proof of Theorem 3.2

Proof of Theorem 3.2. Fix an orthogonal matrix R. Let the first column of $R^\intercal$ be $\theta \in \mathbb{S}^{d-1}$ and the remaining columns form $\theta^\perp$. Since γ satisfies the MFVI optimality condition for $p_R(\mathbf{x}) = p(R^\intercal \mathbf{x})$, we have

$$\mathbb{E}_{\gamma_{-1}}\left[\nabla_{x_1} \log p_R(\mathbf{x})\right] = \nabla_{x_1} \log \gamma_1(x_1).$$

Define $r(\mathbf{x}) := \log p(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}\|_2^2$, then this is equivalent to $\mathbb{E}_{\gamma_{-1}}\left[\langle \theta, \nabla r(R^\intercal \mathbf{x})\rangle\right] = 0$. Thus, for any γ-integrable function $\psi : \mathbb{R} \to \mathbb{R}$,

$$\mathbb{E}_{\gamma}\left[\langle \theta, \nabla r(R^\intercal \mathbf{x})\rangle \cdot \psi(x_1)\right] = 0.$$

By the change of variable $\mathbf{y} = R^\intercal \mathbf{x} \sim \gamma$, we have

$$\mathbb{E}_{\gamma}\left[\langle \theta, \nabla r(\mathbf{y})\rangle \cdot \psi(\theta^\intercal \mathbf{y})\right] = 0. \tag{11}$$

Write the Hermite polynomial expansion of r as

$$r(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}^d} c_{\mathbf{k}} \cdot \mathrm{He}_{\mathbf{k}}(\mathbf{x}),$$

where the sum is over all the multi-indices in $\mathbb{N}^d$ and $\mathrm{He}_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^d \mathrm{He}_{k_i}(x_i)$ is the product of univariate Hermite polynomials. We use the normalized probabilist's Hermite polynomials, which satisfy

$$\mathbb{E}_{\gamma}\left[\mathrm{He}_{\mathbf{k}}(\mathbf{x}) \cdot \mathrm{He}_{\mathbf{k}'}(\mathbf{x})\right] = \delta_{\mathbf{k},\mathbf{k}'},$$
$$\mathrm{He}'_m(x) = \sqrt{m} \cdot \mathrm{He}_{m-1}(x).$$

For $\mathbf{k} \in \mathbb{N}^d$, let $|\mathbf{k}| = \sum_{i=1}^d k_i$, $\mathbf{k}! = \prod_{i=1}^d k_i!$, and $\theta^{\mathbf{k}} = \prod_{i=1}^d \theta_i^{k_i}$.

The derivative of r with respect to x_i can be expressed as

$$\nabla_{x_i} r(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}^d} c_{\mathbf{k}} \sqrt{k_i} \cdot \mathrm{He}_{\mathbf{k}-\mathbf{e}_i}(\mathbf{x}),$$

thus

$$\langle \theta, \nabla r(\mathbf{x}) \rangle = \sum_{i=1}^{d} \theta_i \sum_{\mathbf{k} \in \mathbb{N}^d} c_{\mathbf{k}} \sqrt{k_i} \cdot \mathrm{He}_{\mathbf{k}-\mathbf{e}_i}(\mathbf{x}).$$

Setting $\psi = \mathrm{He}_m$ in Equation (11) gives

$$\begin{aligned} 0 &= \mathbb{E}_\gamma \left[\langle \theta, \nabla r(\mathbf{x}) \rangle \cdot \mathrm{He}_m(\theta^\intercal \mathbf{x}) \right] \\ &= \sum_{i=1}^{d} \theta_i \sum_{\mathbf{k} \in \mathbb{N}^d} c_{\mathbf{k}} \sqrt{k_i} \cdot \mathbb{E}_\gamma \left[\mathrm{He}_{\mathbf{k}-\mathbf{e}_i}(\mathbf{x}) \cdot \mathrm{He}_m(\theta^\intercal \mathbf{x}) \right]. \end{aligned}$$

By the addition formula of Hermite polynomials (https://dlmf.nist.gov/18.18#ii)

$$\mathrm{He}_m(\theta^\intercal \mathbf{x}) = \sum_{|\mathbf{j}|=m} \sqrt{\frac{m!}{\mathbf{j}!}} \theta^{\mathbf{j}} \cdot \mathrm{He}_{\mathbf{j}}(\mathbf{x}),$$

the last equation becomes

$$0 = \sum_{i=1}^{d} \theta_i \sum_{\mathbf{k} \in \mathbb{N}^d} c_{\mathbf{k}} \sqrt{k_i} \sum_{|\mathbf{j}|=m} \sqrt{\frac{m!}{\mathbf{j}!}} \theta^{\mathbf{j}} \cdot \mathbb{E}_\gamma \left[\mathrm{He}_{\mathbf{k}-\mathbf{e}_i}(\mathbf{x}) \cdot \mathrm{He}_{\mathbf{j}}(\mathbf{x}) \right].$$

Since the expectation vanishes unless $\mathbf{k} - \mathbf{e}_i = \mathbf{j}$, it reduces to

$$\begin{aligned} 0 &= \sum_{i=1}^{d} \theta_i \sum_{\mathbf{k}:|\mathbf{k}-\mathbf{e}_i|=m} c_{\mathbf{k}} \sqrt{k_i} \sqrt{\frac{m!}{(\mathbf{k}-\mathbf{e}_i)!}} \theta^{\mathbf{k}-\mathbf{e}_i} \\ &= \sum_{i=1}^{d} \sum_{\mathbf{k}:|\mathbf{k}|=m+1} c_{\mathbf{k}} k_i \sqrt{\frac{m!}{\mathbf{k}!}} \cdot \theta^{\mathbf{k}} \\ &= (m+1)\sqrt{m!} \sum_{\mathbf{k}:|\mathbf{k}|=m+1} \frac{c_{\mathbf{k}}}{\sqrt{\mathbf{k}!}} \cdot \theta^{\mathbf{k}}. \end{aligned}$$

Since this holds for almost every $\theta \in \mathbb{S}^{d-1}$ and any $m \in \mathbb{N}$, it follows that $c_{\mathbf{k}} = 0$ for all $|\mathbf{k}| \geq 1$. This proves that r is a constant function, and hence $p = \gamma$.

□

A.2 Proof of Theorem 3.3

Proof of Theorem 3.3. The proof largely follows the proof of [35, Theorem 2.5]. Using the Fokker-Planck equation, one can show that

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathrm{KL}(q_t \,\|\, p) = -\tilde{I}(q_t, p), \tag{12}$$

With the definition of $h_{t,i}$, the projected Fisher information $\tilde{I}(q_t, p)$ can be expressed as

$$\tilde{I}_t := \tilde{I}(q_t, p) = \mathbb{E}_{q_t} \left[\|h_t\|_2^2 \right], \tag{13}$$

Differentiating both sides of Equation (13) and following the calculation in [35, Section 6.4], we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{I}_t = -2 \cdot \mathbb{E}_{q_t}\left[\|\nabla h_t\|_{\mathrm{Frob}}^2 + \langle \nabla h_t, \nabla^2 U \nabla h_t \rangle\right],$$

where ∇h_t is a diagonal matrix. Denote $J_t = \mathbb{E}_{q_t}\left[\|\nabla h_t\|_{\mathrm{Frob}}^2\right]$. By the assumption that $\lambda I_d \preceq \nabla^2 U \preceq L I_d$, we have

$$\lambda \cdot \mathbb{E}_{q_t}\left[\|h_t\|_2^2\right] \le \mathbb{E}_{q_t}\left[\langle h_t, \nabla^2 U h_t\right] \le L \cdot \mathbb{E}_{q_t}\left[\|h_t\|_2^2\right].$$

Therefore,

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{I}_t + 2\lambda \tilde{I}_t \le -2 \cdot J_t, \quad \frac{\mathrm{d}}{\mathrm{d}t}\tilde{I}_t + 2L\tilde{I}_t \ge -2 \cdot J_t,$$

which leads to

$$e^{-2Lt}\tilde{I}_0 - 2\int_0^t e^{-2L(t-s)} J_s \mathrm{d}s \le \tilde{I}_t \le e^{-2\lambda t}\tilde{I}_0 - 2\int_0^t e^{-2\lambda(t-s)} J_s \mathrm{d}s.$$

Integrating t from 0 to ∞, and using Equation (12), we have

$$\frac{1}{2L}\tilde{I}_0 - \frac{1}{L}\int_0^\infty J_s \mathrm{d}s \le \mathrm{KL}(\gamma \,\|\, p) - \lim_{t\to\infty} \mathrm{KL}(q_t \,\|\, p) \le \frac{1}{2\lambda}\tilde{I}_0 - \frac{1}{\lambda}\int_0^\infty J_s \mathrm{d}s.$$

The proof is completed by the fact that $\lim_{t\to\infty} \mathrm{KL}(q_t \,\|\, p) = \inf_{q\in\mathcal{P}_{\mathrm{prod}}} \mathrm{KL}(q \,\|\, p)$ [35, Theorem 2.4]. □

A.3 Proof of Corollary 4.2

Proof of Corollary 4.2. Since R is uniformly distributed, we have

$$\mathbb{E}_R\left[\sum_{i=1}^d (RHR^\intercal)_{ii}^2\right] = d \cdot \mathbb{E}_{\mathbf{r}\sim\mathrm{Unif}(\mathbb{S}^{d-1})}\left[(\mathbf{r}^\intercal H \mathbf{r})^2\right].$$

Because the distribution of $\mathbf{r}$ is invariant to orthogonal transformations, we have

$$\mathbb{E}\left[(\mathbf{r}^\intercal H \mathbf{r})^2\right] = \mathbb{E}\left[(\mathbf{r}^\intercal D \mathbf{r})^2\right] = \mathbb{E}\left[(\sum_{i=1}^d \nu_i r_i^2)^2\right] = \sum_{i,j=1}^d \mathbb{E}\left[\nu_i \nu_j r_i^2 r_j^2\right].$$

Define $\mathbf{y} = (r_1^2, \ldots, r_d^2)$. Note that $\mathbf{y} \overset{d}{=} \frac{\mathbf{x}}{\|\mathbf{x}\|_2^2}$, where $\mathbf{x} \sim \mathcal{N}(0, I_d)$. Since $x_i^2 \overset{iid}{\sim} \mathrm{Gamma}(1/2,\, 2)$ for $1 \le i \le d$, we have $\mathbf{y} \sim \mathrm{Dirichlet}(1/2, \ldots, 1/2)$. Then it is known that (see e.g. [34, Chapter 49])

$$\mathbb{E}\left[r_i^4\right] = \frac{3}{d(d+2)}, \quad \mathbb{E}\left[r_i^2 r_j^2\right] = \frac{1}{d(d+2)} \quad \text{for } i \ne j.$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[(\mathbf{r}^\intercal H \mathbf{r})^2\right] &= \sum_{i=1}^d \nu_i^2 \frac{3}{d(d+2)} + \sum_{i\ne j} \nu_i \nu_j \frac{1}{d(d+2)} \\ &= \frac{1}{d(d+2)}\left[2\sum_{i=1}^d \nu_i^2 + (\sum_{i=1}^d \nu_i)^2\right].\end{aligned}$$

Combining this with Theorem 3.4 completes the proof. □

A.4 Proof of Theorem 5.1

Proof of Theorem 5.1. Because the MF approximation of $\mathcal{N}(0, \Sigma)$ is the standard Gaussian, we have $(\Sigma^{-1})_{ii} = 1$ for $1 \le i \le d$. Write the eigendecomposition of Σ as $\Sigma = U\Lambda U^\intercal$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$. Denote $\nu_i = 1/\lambda_i$, then they must satisfy

$$\sum_{i=1}^d \nu_i = \sum_{i=1}^d \frac{1}{\lambda_i} = d.$$

The initial KL divergence is equal to

$$\mathcal{L}_0 = \mathrm{KL}(\mathcal{N}(0, I_d) \,\|\, \mathcal{N}(0, \Sigma)) = \frac{1}{2}\sum_{i=1}^d (\log \lambda_i + \frac{1}{\lambda_i} - 1) = \frac{1}{2}\sum_{i=1}^d \log \lambda_i.$$

For an orthogonal matrix R, the MF approximation of $\mathcal{N}(0, R\Sigma R^\intercal)$ is $\mathcal{N}(0, S)$, where S is diagonal with $S_{ii}^{-1} = ((R\Sigma R^\intercal)^{-1})_{ii}$. Write $\tilde{R} = RU$, then

$$\frac{1}{S_{ii}} = (\tilde{R}\Lambda^{-1}\tilde{R}^\intercal)_{ii} = \sum_{j=1}^d \frac{\tilde{R}_{ij}^2}{\lambda_j}.$$

The KL divergence after MFVI is

$$\begin{aligned}
\mathcal{L}_1 &= \mathrm{KL}(\mathcal{N}(0, S) \,\|\, \mathcal{N}(0, R\Sigma R^\intercal)) \\
&= \frac{1}{2}\Big[\,\mathrm{tr}(S(R\Sigma R^\intercal)^{-1}) + \log|\Sigma| - \log|S| - d\Big] \\
&= \frac{1}{2}\Big[\,\mathrm{tr}(S\tilde{R}\Lambda^{-1}\tilde{R}^\intercal) + \sum_{i=1}^d \log\lambda_i - \sum_{i=1}^d \log S_{ii} - d\Big].
\end{aligned}$$

Note that

$$\mathrm{tr}(S\tilde{R}\Lambda^{-1}\tilde{R}^\intercal) = \sum_{i=1}^d S_{ii}\sum_{j=1}^d \frac{\tilde{R}_{ij}^2}{\lambda_j} = \sum_{i=1}^d S_{ii}\cdot\frac{1}{S_{ii}} = d.$$

Thus

$$\mathcal{L}_1 = \frac{1}{2}\Big[\sum_{i=1}^d \log\lambda_i - \sum_{i=1}^d \log S_{ii}\Big] = \frac{1}{2}\Big[\sum_{i=1}^d \log\lambda_i + \sum_{i=1}^d \log\sum_{j=1}^d \frac{\tilde{R}_{ij}^2}{\lambda_j}\Big].$$

Note that $\tilde{R} = RU$ is a uniformly distributed orthogonal matrix. Applying Lemma A.1 with $\nu_i = 1/\lambda_i$, we have

$$\mathbb{E}_R\Big[\sum_{i=1}^d \log\sum_{j=1}^d \tilde{R}_{ij}^2\nu_j\Big] \le \frac{2}{(d+2)\kappa^2}\sum_{i=1}^d \log\nu_i.$$

Therefore,

$$\begin{aligned}\mathbb{E}_R\left[\mathcal{L}_1\right] &\le \frac{1}{2}\left[\sum_{i=1}^d \log\lambda_i - \frac{2}{(d+2)\kappa^2}\sum_{i=1}^d \log\lambda_i\right] \\ &= \frac{1}{2}\Big(1 - \frac{2}{(d+2)\kappa^2}\Big)\sum_{i=1}^d \log\lambda_i \\ &= \Big(1 - \frac{2}{(d+2)\kappa^2}\Big)\cdot\mathcal{L}_0.\end{aligned}$$

□

Lemma A.1. *Suppose $\nu_i > 0$ and $\sum_{i=1}^d \nu_i = d$. For $\nu_{\max} = \max\limits_{1\le i\le d}\nu_i$ and $\nu_{\min} = \min\limits_{1\le i\le d}\nu_i$, assume $\frac{\nu_{\max}}{\nu_{\min}} \le \kappa$. Then*

$$\mathbb{E}_R\left[\sum_{i=1}^d \log\sum_{j=1}^d R_{ij}^2\nu_j\right] \le \frac{2}{(d+2)\kappa^2}\sum_{i=1}^d \log\nu_i,$$

where the expectation is taken over uniformly distributed orthogonal matrix $R \in \mathbb{R}^{d\times d}$.

Proof. Note that each row of R is uniformly distributed on $\mathbb{S}^{d-1} = \{\mathbf{v}\in\mathbb{R}^d : \mathbf{v}^\intercal\mathbf{v} = 1\}$. Denote $\mathbf{y} = (R_{i1}^2, \ldots, R_{id}^2)$, so we have $\sum_{i=1}^d y_i = 1$ and $\mathbb{E}\left[y_i\right] = 1/d$. The expectation on the left-hand-side is equal to

$$\mathbb{E}_R\left[\sum_{i=1}^d \log\sum_{j=1}^d R_{ij}^2\nu_j\right] = d\cdot\mathbb{E}_{\mathbf{y}}\left[\log\sum_{i=1}^d \nu_i y_i\right].$$

Define the random variable $W = \sum_{i=1}^d \nu_i y_i$, which satisfies

$$\nu_{\min} \le W \le \nu_{\max}, \quad \mathbb{E}\left[W\right] = \frac{1}{d}\sum_{i=1}^d \nu_i = 1.$$

Applying Lemma A.2 with $g(x) = -\log x$ and the random variable W, we have

$$\frac{1}{2\nu_{\max}^2}\operatorname{Var}\left[W\right] \le -\mathbb{E}\left[\log W\right] \le \frac{1}{2\nu_{\min}^2}\operatorname{Var}\left[W\right].$$

Similarly, define Z as the random variable that is uniformly distributed on $\{\nu_1, \ldots, \nu_d\}$. Then Z is also bounded between $\nu_{\min}$ and $\nu_{\max}$, and satisfies $\mathbb{E}\left[Z\right] = \frac{1}{d}\sum_{i=1}^d \nu_i = 1$. Applying Lemma A.2 again with $g(x) = -\log x$ and the random variable Z, we have

$$\frac{1}{2\nu_{\max}^2}\operatorname{Var}\left[Z\right] \le -\mathbb{E}\left[\log Z\right] \le \frac{1}{2\nu_{\min}^2}\operatorname{Var}\left[Z\right].$$

By Lemma A.3, we have $\mathrm{Var}[W] = \frac{2}{d+2}\mathrm{Var}[Z]$. Therefore,

$$\begin{aligned}\mathbb{E}[\log W] &\le -\frac{1}{2\nu_{\max}^2}\mathrm{Var}[W] \\ &= -\frac{1}{2\nu_{\max}^2}\frac{2}{d+2}\mathrm{Var}[Z] \\ &\le \frac{1}{2\nu_{\max}^2}\frac{2}{d+2}\cdot 2\nu_{\min}^2\mathbb{E}[\log Z] \\ &\le \frac{2}{(d+2)\kappa^2}\frac{1}{d}\sum_{i=1}^{d}\log\nu_i.\end{aligned}$$

This proves that $d\cdot\mathbb{E}\left[\log\sum_{i=1}^{d}\nu_i y_i\right] \le \frac{2}{(d+2)\kappa^2}\sum_{i=1}^{d}\log\nu_i$. □

Lemma A.2. *Let X be a random variable that is bounded between $[a,b]$. Let $g:[a,b]\to\mathbb{R}$ be twice continuously differentiable. Then*

$$\frac{1}{2}\inf_{x\in[a,b]}g''(x)\,\mathrm{Var}[X] \le \mathbb{E}[g(X)] - g(\mathbb{E}[X]) \le \frac{1}{2}\sup_{x\in[a,b]}g''(x)\,\mathrm{Var}[X].$$

Proof of Lemma A.2. Denote $\mu = \mathbb{E}[X]\in[a,b]$ as the mean of X. Taking the Taylor expansion of $g(X)$ around μ gives

$$\frac{1}{2}\inf_{x\in[a,b]}g''(x)(X-\mu)^2 \le g(X) - g(\mu) - g'(\mu)(X-\mu) \le \frac{1}{2}\sup_{x\in[a,b]}g''(x)(X-\mu)^2.$$

Taking expectation of both sides proves the result. □

Lemma A.3. *Suppose $\mathbf{r}$ is uniformly distributed on the sphere $\mathbb{S}^{d-1}$ and $y_i = r_i^2$. Suppose $\nu_1,\dots,\nu_d \ge 0$ and satisfy $\sum_{i=1}^{d}\nu_i = d$. Then*

$$\mathrm{Var}\left[\sum_{i=1}^{d}\nu_i y_i\right] = \frac{2}{(d+2)d}\sum_{i=1}^{d}(\nu_i-1)^2.$$

Proof of Lemma A.3. Because $\mathbf{y} = (y_1,\dots,y_d)\sim\mathrm{Dirichlet}(1/2,\dots,1/2)$, we have from [34, Chapter 49] that

$$\mathrm{Cov}[y_i,y_j] = \frac{\delta_{i,j}\frac{1}{d} - \frac{1}{d^2}}{\frac{d}{2}+1}.$$

The variance of $\sum_{i=1}^{d} \nu_i y_i$ can be computed as

$$\begin{aligned}
\operatorname{Var}\left[\sum_{i=1}^{d} \nu_i y_i\right] &= \sum_{i=1}^{d} \operatorname{Var}\left[\nu_i y_i\right] + \sum_{i \neq j} \operatorname{Cov}\left[\nu_i y_i, \nu_j y_j\right] \\
&= \sum_{i=1}^{d} \nu_i^2 \frac{\frac{1}{d} - \frac{1}{d^2}}{\frac{d}{2}+1} + \sum_{i \neq j} \nu_i \nu_j \frac{-\frac{1}{d^2}}{\frac{d}{2}+1} \\
&= \sum_{i=1}^{d} \nu_i^2 \frac{\frac{1}{d}}{\frac{d}{2}+1} + \sum_{i,j=1}^{d} \nu_i \nu_j \frac{-\frac{1}{d^2}}{\frac{d}{2}+1} \\
&= \frac{2}{d(d+2)} \sum_{i=1}^{d} \nu_i^2 - \frac{2}{d^2(d+2)} (\sum_{i=1}^{d} \nu_i)^2 \\
&= \frac{2}{d(d+2)} \sum_{i=1}^{d} \nu_i^2 - \frac{2}{d+2} \\
&= \frac{2}{d+2} (\frac{1}{d} \sum_{i=1}^{d} \nu_i^2 - 1) \\
&= \frac{2}{d+2} \cdot \frac{1}{d} \sum_{i=1}^{d} (\nu_i - 1)^2.
\end{aligned}$$

□

B Additional details for the experiments

B.1 Poisson GLMM

We use the first approach introduced in [55] to reparametrize the posterior of the Poisson GLMM. The conditional distribution of $b_i \mid \theta_G, \mathbf{y}_i$ is proportional to

$$p(b_i \mid \theta_G, \mathbf{y}_i) \propto \varphi(b_i; 0, \sigma^2) \cdot \prod_{j=1}^{m} p(y_{ij} \mid \eta_{ij}), \quad \eta_{ij} = \beta_0 + \beta_1 x_{ij} + b_i.$$

The log likelihood $\log p(y_{ij} \mid \eta_{ij}) = y_{ij}\eta_{ij} - e^{\eta_{ij}}$ is approximated by its second-order Taylor expansion at the MLE $\hat{\eta}^{\mathrm{MLE}}$:

$$\log p(y_{ij} \mid \eta_{ij}) \approx -\frac{1}{2} h_{ij} (\eta_{ij} - \hat{\eta}_{ij}^{\mathrm{MLE}})^2, \quad h_{ij} = e^{\hat{\eta}_{ij}^{\mathrm{MLE}}},$$

and $\log p(b_i \mid \theta_G, \mathbf{y}_i)$ is approximated by the quadratic function

$$-\frac{1}{2\sigma^2} b_i^2 - \frac{1}{2} \sum_{j=1}^{m} h_{ij} (\beta_0 + \beta_1 x_{ij} + b_i - \hat{\eta}_{ij}^{\mathrm{MLE}})^2.$$

The reparametrization $\tilde{b}_i = (b_i - \lambda_i)/L_i$ is determined by this Gaussian approximation. When $y_{ij} = 0$, we follow [55] and define the MLE as $\hat{\eta}_{ij}^{\mathrm{MLE}} = \psi(y_{ij} + 0.5)$ where ψ is the digamma function.

A mean-field Gaussian approximation is first fitted to the reparametrized posterior, and its mean and standard deviation are used to center the target. This initialization step runs for 200 Adam iterations with a learning rate of 0.1. Both Gaussian VI and MFVI+PCA are then optimized for 100 iterations using the same learning rate 0.1. Gradients are computed using 5000 Monte Carlo samples.

References

[1] Agrawal, A. and Domke, J. (2025). Disentangling impact of capacity, objective, batchsize, estimators, and step-size on flow VI. In *International Conference on Artificial Intelligence and Statistics*, pages 325–333. PMLR.

[2] Arnese, M. and Lacker, D. (2024). Convergence of coordinate ascent variational inference for log-concave measures via optimal transport. *arXiv preprint arXiv:2404.08792*.

[3] Balasubramanian, K., Chewi, S., Erdogdu, M. A., Salim, A., and Zhang, S. (2022). Towards a theory of non-log-concave sampling: first-order stationarity guarantees for Langevin Monte Carlo. In *Conference on Learning Theory*, pages 2896–2923. PMLR.

[4] Bhattacharya, A., Pati, D., and Yang, Y. (2025). On the convergence of coordinate ascent variational inference. *The Annals of Statistics*, 53(3):929–962.

[5] Bickel, P., Choi, D., Chang, X., and Zhang, H. (2013). Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics*, 41(4).

[6] Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern Recognition and Machine Learning*, volume 4. Springer.

[7] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

[8] Blessing, D., Jia, X., Esslinger, J., Vargas, F., and Neumann, G. (2024). Beyond ELBOs: a large-scale evaluation of variational methods for sampling. In *Proceedings of the 41st International Conference on Machine Learning*, pages 4205–4229.

[9] Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, 76:1–32.

[10] Che, B., Chen, Y., Huan, Z., Huang, D. Z., and Wang, W. (2025). Stable derivative free Gaussian mixture variational inference for Bayesian inverse problems. *SIAM Journal on Scientific Computing*, 47(5):A2583–A2608.

[11] Chen, S. and Gopinath, R. (2000). Gaussianization. *Advances in Neural Information Processing Systems*, 13.

[12] Constantine, P. G., Dow, E., and Wang, Q. (2014). Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524.

[13] Cui, T., Martin, J., Marzouk, Y. M., Solonen, A., and Spantini, A. (2014). Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30(11):114015.

[14] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using Real NVP. In *International Conference on Learning Representations*.

[15] Draxler, F., Kühmichel, L., Rousselot, A., Müller, J., Schnörr, C., and Köthe, U. (2023). On the convergence rate of Gaussianization with random rotations. In *International Conference on Machine Learning*, pages 8449–8468. PMLR.

[16] Du, Q., Wang, K., Zhang, E., and Zhong, C. (2024). A particle algorithm for mean-field variational inference. *arXiv preprint arXiv:2412.20385*.

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

[18] El Moselhy, T. A. and Marzouk, Y. M. (2012). Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815–7850.

[19] Fong, Y., Rue, H., and Wakefield, J. (2010). Bayesian inference for generalized linear mixed models. *Biostatistics*, 11(3):397–412.

[20] Friedman, J. and Tukey, J. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 100(SLAC-PUB-1312).

[21] Friedman, J. H. (1987). Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266.

[22] Gabrié, M., Rotskoff, G. M., and Vanden-Eijnden, E. (2022). Adaptive Monte Carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10):e2109420119.

[23] Gorham, J. and Mackey, L. (2015). Measuring sample quality with Stein's method. *Advances in Neural Information Processing Systems*, 28.

[24] Han, S., Liao, X., Dunson, D., and Carin, L. (2016). Variational Gaussian copula inference. In *Artificial Intelligence and Statistics*, pages 829–838. PMLR.

[25] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97.

[26] Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D., and Vasudevan, S. (2019). Neutra-lizing bad geometry in Hamiltonian Monte Carlo using neural transport. *arXiv preprint arXiv:1903.03704*.

[27] Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.

[28] Hofmann, H. (1994). Statlog (German Credit Data). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NC77.

[29] Huber, P. J. (1985). Projection pursuit. *The Annals of Statistics*, pages 435–475.

[30] Jiang, Y., Chewi, S., and Pooladian, A.-A. (2025). Algorithms for mean-field variational inference via polyhedral optimization in the Wasserstein space. *Foundations of Computational Mathematics*, pages 1–52.

[31] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

[32] Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[33] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*, 29.

[34] Kotz, S., Balakrishnan, N., and Johnson, N. L. (2019). *Continuous Multivariate Distributions, Volume 1: Models and Applications*, volume 1. John Wiley & Sons.

[35] Lacker, D. (2023). Independent projections of diffusions: Gradient flows for variational inference and optimal mean field approximations. *arXiv preprint arXiv:2309.13332*.

[36] Lacker, D., Mukherjee, S., and Yeung, L. C. (2024). Mean field approximations via log-concavity. *International Mathematics Research Notices*, 2024(7):6008–6042.

[37] Laparra, V., Camps-Valls, G., and Malo, J. (2011). Iterative Gaussianization: from ICA to random rotations. *IEEE Transactions on Neural Networks*, 22(4):537–549.

[38] Lavenant, H. and Zanella, G. (2024). Convergence rate of random scan coordinate ascent variational inference under log-concavity. *SIAM Journal on Optimization*, 34(4):3750–3761.

[39] Lin, W., Khan, M. E., and Schmidt, M. (2019). Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In *International Conference on Machine Learning*, pages 3992–4002. PMLR.

[40] Liu, Q. and Wang, D. (2016). Stein variational gradient descent: A general purpose Bayesian inference algorithm. *Advances in Neural Information Processing Systems*, 29.

[41] Liu, S. (2024). Transport quasi-Monte Carlo. *arXiv preprint arXiv:2412.16416*.

[42] Magnusson, M., Torgander, J., Bürkner, P.-C., Zhang, L., Carpenter, B., and Vehtari, A. (2024). posteriordb: Testing, benchmarking and developing Bayesian inference algorithms. In *The 28th International Conference on Artificial Intelligence and Statistics*.

[43] Meng, C., Song, Y., Song, J., and Ermon, S. (2020). Gaussianization flows. In *International Conference on Artificial Intelligence and Statistics*, pages 4336–4345. PMLR.

[44] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

[45] Neville, S. E., Ormerod, J. T., and Wand, M. (2014). Mean field variational Bayes for continuous sparse signal shrinkage: Pitfalls and remedies. *Electronic Journal of Statistics*, 8:1113–1151.

[46] Opper, M. and Archambeau, C. (2009). The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792.

[47] Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64.

[48] Papaspiliopoulos, O., Roberts, G. O., and Sköld, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science*, pages 59–73.

[49] Parno, M. D. and Marzouk, Y. M. (2018). Transport map accelerated Markov chain Monte Carlo. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):645–682.

[50] Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR.

[51] Sklar, M. (1959). Fonctions de répartition à n dimensions et leurs marges. In *Annales de l'ISUP*, volume 8, pages 229–231.

[52] Stan Development Team (2011). *Stan User's Guide*. https://mc-stan.org/docs/stan-users-guide/regression.html#multilevel-2pl-model.

[53] Tabak, E. G. and Turner, C. V. (2013). A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164.

[54] Tabak, E. G. and Vanden-Eijnden, E. (2010). Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233.

[55] Tan, L. S. (2021). Use of model reparametrization to improve variational Bayes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(1):30–57.

[56] Tran, M.-N., Tseng, P., and Kohn, R. (2023). Particle mean field variational Bayes. *arXiv preprint arXiv:2303.13930*.

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

[58] Wang, B. and Titterington, D. M. (2005). Inadequacy of interval estimates corresponding to variational Bayesian approximations. In *International workshop on artificial intelligence and statistics*, pages 373–380. PMLR.

[59] Zahm, O., Cui, T., Law, K., Spantini, A., and Marzouk, Y. (2022). Certified dimension reduction in nonlinear Bayesian inverse problems. *Mathematics of Computation*, 91(336):1789–1835.

[60] Zhai, S., Zhang, R., Nakkiran, P., Berthelot, D., Gu, J., Zheng, H., Chen, T., Bautista, M. Á., Jaitly, N., and Susskind, J. M. (2025). Normalizing flows are capable generative models. In *Forty-second International Conference on Machine Learning*.

[61] Zhang, F. and Gao, C. (2020). Convergence rates of variational posterior distributions. *The Annals of Statistics*, 48(4):2180–2207.

[62] Zhang, Y. and Yang, Y. (2024). Bayesian model selection via mean-field variational approximation. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(3):742–770.