Unconstrained optimizers for ICA learning on oblique manifold using Parzen density estimation

S. Easter Selvan[a], Umberto Amato[b], Chunhong Qi[c], Kyle A. Gallivan[c], M. Francesca Carfora[b], Michele Larobina[d], Bruno Alfano[d]

[a]*Department of Mathematical Engineering, Université Catholique de Louvain, 1348 Louvain-la-Neuve, Belgium (e-mail: easterselvans@gmail.com).*
[b]*Istituto per le Applicazioni del Calcolo 'Mauro Picone'–Sede di Napoli, Consiglio Nazionale delle Ricerche, Via Pietro Castellino 111, Napoli 80131, Italy (e-mail: {u.amato, f.carfora}@iac.cnr.it).*
[c]*Department of Mathematics, Florida State University, Tallahassee, FL, 32306, USA (e-mail: {cqi, gallivan}@math.fsu.edu).*
[d]*Istituto di Biostrutture e Bioimmagini, Consiglio Nazionale delle Ricerche, Via Pansini 5–Edificio 10, Napoli 80131, Italy (e-mail: {michele.larobina, bruno.alfano}@ibb.cnr.it).*

Abstract

A Riemannian manifold optimization strategy is proposed to facilitate the relaxation of the orthonormality constraint in a more natural way in the course of performing the independent component analysis (ICA) that employs a mutual information (MI)-based source adaptive contrast function. Despite the extensive development of manifold techniques catering to the orthonormality constraint, we are devoid of adequate oblique manifold ($\mathcal{OB}$) algorithms to intrinsically handle the normality constraint. Essentially, imposing the normality constraint implicitly, in line with the ICA definition, guarantees a substantial improvement in the solution accuracy, by way of increased degrees of freedom while searching an optimal unmixing ICA matrix, in contrast with the orthonormality constraint. Towards this, a design of the steepest descent (SD), conjugate gradient (CG) with Hager-Zhang (HZ) or a hybrid update parameter, quasi-Newton (QN) and cost effective quasi-Newton (QN-CE) methods, intended for $\mathcal{OB}$, is presented in the paper; their performance is validated using natural images and audio signals, and compared with the popular state-of-the-art approaches. We surmount the computational challenge associated with the direct estimation of the source densities using the improved fast Gauss transform (IFGT)-based function and gradient evaluation. The proposed $\mathcal{OB}$ schemes may find applicability in the offline image/signal analysis, wherein on the one hand the computational overhead can be tolerated and on the other hand the solution quality

holds paramount interest.

Keywords:
Conjugate gradient, Oblique manifold, Parzen window, Quasi-Newton, Retraction, Steepest descent, Vector transport

---

1. Introduction

Linear independent component analysis (ICA) linearly transforms multivariate data such that the transformed data components remain as independent as possible. The efficiency of an ICA approach to separate the mixed real world data relies on the choice of a contrast function and an optimization strategy imposing necessary constraints. The supreme importance in selecting the contrast function is its capability to adaptively estimate the source densities. As a consequence, use of a fixed nonlinear function to measure negentropy approximation in the case of popular fastICA produces inferior separation results, compared to the methods employing source adaptive contrast functions [1]. On the other hand, moment-based ICA schemes such as JADE measure independence using higher-order cross-cumulants, which may not be well-suited for real data in many applications [2], for example, they are not robust to outliers [3]. In contrast, the non-parametric density estimation does not assume any statistical model of the sources besides being source adaptive. Even though the parametric models achieve source adaptivity, their performance degrades when the sources do not follow the assumed parametric model [1]. Therefore for applications where the major emphasis is the estimation accuracy of the ICA unmixing matrix, and the computational burden can be tolerated, the non-parametric density estimation is more apt.

Traditionally, the ICA estimation is treated as a constrained optimization of a contrast function, so that it becomes plausible for the algorithm to find an optimal solution in the restricted search space. The constraint imposition between subsequent iterations, as exemplified by the Gram-Schmidt (GS) orthogonalization procedure, is a well-known constraint handling strategy. Alternatively, the contrast function is modified by adding a penalty term such that the function has a minimum when the constraint is satisfied [4]. Another possibility is to restrict changes to the unmixing matrix to eliminate any deviation from the constraint [5]. Nevertheless, these attempts do not guarantee the constraint always, as the tendency of the unmixing matrix to "drift" away from the constraint surface has to be continually tackled [6]. Therefore, instead of seeking for an optimal unmixing matrix

on the Euclidean space, the Riemannian optimization is formulated in such a way that the search is always restricted to the constraint set admitting a manifold structure, which enjoys the following merits.

1. It does not necessitate any extra effort to "coax" the solutions to follow the constraint, since the optimization takes place on the constraint surface.
2. All the iterates (trial solutions) during the optimization procedure remain "locked" to the curved space, which ensures good convergence characteristics.

Consequently, in the pursuit to impose the constraint implicitly and simultaneously preserve the convergence characteristics, manifold optimization has been a topic of interest in ICA applications during recent years.

1.1. Is normality constraint a viable assumption in ICA estimation?

Indeed a constellation of ICA learning algorithms on the Stiefel, Grassmann and flag manifolds [4, 7–11] have already been proposed, which are premised on the hypothesis that the independent components (ICs) are mutually orthogonal to each other. Nevertheless, we consider a collection of unconstrained deterministic optimization algorithms on the oblique manifold ($\mathcal{OB}$)—the set of all matrices with normalized columns—by relaxing the orthonormality constraint for the following reasons.

1. An insightful definition of the ICA as postulated by Lee *et al.* [12, p. 867] goes like this: ICA is a way of finding a linear non-orthogonal co-ordinate system in multivariate data that minimizes mutual information (MI) among the axial projections of the data.
2. Despite the orthogonality constraint being well-founded in the case of Principal Component Analysis (PCA), there is no underlying theoretical justification for this constraint to be maintained in the ICA. Hyvärinen *et al.* [13, pp. 223, 275–276] reiterated the key difference between using the MI and an ICA contrast based on the non-Gaussianity; we force the estimates of the ICs to be uncorrelated while maximizing the sum of their non-Gaussianities, whereas, this is not necessary when minimizing the MI.
3. Intuitively, relying on the orthonormality constraint implies a smaller subset of unitary matrices which simplifies the optimization, however to the detriment of solution accuracy. As a consequence of imposing the orthonormality constraint on an unmixing matrix of order $d \times d$, the degrees of freedom are restricted to $d(d-1)/2$; on the contrary, the

choice of normality constraint will allow one to increase the degrees of freedom to $d(d-1)$.

4. Lewicki and Sejnowski emphasized in their work that the ICA being an extension of the PCA allows the learning of non-orthogonal (oblique) bases for data with non-Gaussian distributions. It has been stressed with an illustration [14, pp. 343–344] that if the data have non-Gaussian structure and the ICA model insists on the orthogonality constraint, then the model poses a risk of underestimating the likelihood of data in the dense regions and overestimating it in the sparse regions. By Shannon's theorem, this will limit the efficiency of the representation. Therefore it is recommended that the ICA assumes the coefficients have non-Gaussian structure and allows the components to be non-orthogonal.
5. To further support the applicability of unit-norm constraint in the ICA problem, Douglas *et al.* [15] described self-stabilized, gradient-based, one-unit ICA algorithms to implicitly maintain the normality constraint to estimate the minimum-kurtosis ICs. Furthermore, of late, a conjugate gradient (CG) algorithm was suggested on the unit sphere by Shen *et al.* [16] for the non-whitened one-unit ICA problem.
6. Even though relaxing the constraint from orthonormality to normality provides a good avenue in the quest for better minimization of an ICA contrast function, and designing the Riemannian optimization algorithms suited for $\mathcal{OB}$ is relatively simpler compared to their counterparts on the Stiefel or Grassmann manifold, as far as we are aware, the problem of ICA learning on $\mathcal{OB}$ has been considered only in [17] and [18].

Since the MI is invariant to changes in the scale of the sources, the minimization problem admits infinite number of solutions corresponding to any real value of the scaling coefficient; precisely, the problem is ill-posed. Therefore, it becomes essential to enforce a constraint either on the norm of the unmixing matrix or its columns to have a unique solution. For simplicity's sake, we opt for the unit-norm column constraint of the unmixing matrix. The naive implementation is to relax the orthonormality constraint, and to search in a more exhaustive space such as the set of all normalized column matrices, by imposing the normality constraint between the iterations. However, such an attempt does adversely affect the convergence properties of the optimizers, in the vein of ICA approaches enforcing an orthogonalization procedure during the course of optimization, and in turn warrants the design of Riemannian algorithms. We envisage that allowing

more degrees of freedom for the unmixing matrix by designing a suitable Riemannian unconstrained optimizer, and making use of a source adaptive contrast function which is robust in the absence of *a priori* statistical information about the sources, will definitely improve the solution quality in an ICA algorithm.

1.2. Related work

Sengupta *et al.* employed kernel density estimation using smoothing methods to estimate the underlying probability density distributions of the sources; this algorithm was reported to perform the source separation satisfactorily in the presence of outliers and in non-Gaussian zero-kurtotic signal mixture [3]. A generalized form of Shannon's entropy, called Rényi's entropy, was estimated using Parzen windowing with a Gaussian kernel by Hild *et al.* in the blind source separation (BSS) algorithm; the Givens rotation ensures the orthonormality constraint in the unmixing matrix [19]. Even though this algorithm claimed superior performance when experimented with audio sources in terms of the signal-to-distortion ratio (SDR), compared to the fastICA, infomax and Comon's minimum mutual information (MMI), in a later work by Pham *et al.*, the Rényi-entropy-based criterion was proved to be risky in the context of BSS [20]. Pham provided low-cost algorithms to perform ICA through the minimization of the MI criterion with the help of a binning technique and cardinal splines; a new estimator of the score function for calculating the gradient of the estimated criterion was also introduced [21]. Chen demonstrated a fast kernel density ICA (FastKDE) algorithm by choosing the Laplacian kernel for kernel density estimation; the computational and statistical efficiencies were reported to be promising [22]. A noteworthy contribution was made by Boscolo *et al.* towards a nonparametric ICA approach (NPICA) that estimates the kernel density using the fast Fourier transform (FFT)-based technique; it outperformed the state-of-the-art ICA techniques in terms of the signal-to-interference ratio (SIR) [2]. Even though the orthonormality constraint among the ICs was relaxed in the quasi-Newton (QN) method, the optimization scheme in the NPICA does not follow the framework of the Riemannian-manifold-based approach endowed with the super-linear convergence characteristics. Shwartz *et al.* presented an analogous ICA algorithm based on accelerated kernel entropy estimation utilizing FFT-based fast convolution to manage the computationally intense task [23]. In the recent past, Xue *et al.* proposed a source adaptive ICA algorithm (GEKD-ICA) hinged on iteratively solving the gradient equation and simultaneously estimating the density by kernel density method [1]; in most of the test cases, a comparable performance of this

scheme with the NPICA, in terms of the SIR, performance index (PI) and computational complexity, was evident from the experimental results. Tsai and Lai attempted a method that combines binning-principle-based density estimation and a particle swarm optimizer (PSO) for background subtraction in indoor surveillance applications [24]; nonetheless, its robustness in the camouflage foreground and the extension of this method to higher data dimensions are questionable.

Learning algorithms on manifolds, stemming from differential geometry techniques, have been of primary interest for the machine learning community over the past few decades. Gabay discussed search algorithms by treating the Riemannian optimization to be locally equivalent to the smoothly constrained Euclidean space optimization [25]. Edelman *et al.* developed a Newton and a CG algorithm on the Grassmann and Stiefel manifolds which represent orthogonality constraint [26]. Fiori customized the differential geometry concepts to the orthogonal group, and demonstrated the non-negative ICA through a deterministic- and a diffusion-type-gradient algorithm [7]. In [27], retraction-based ICA algorithms tailored for the orthogonal matrix manifold optimization were experimentally concluded better in performance than the fastICA. Plumbley solved the non-negative ICA problem by optimizing over the space of orthogonal matrices with a steepest descent (SD) and a CG method in conjunction with techniques involving a Lie group and its corresponding Lie algebra [4]. Nishimori *et al.* investigated a Riemannian optimization method on the flag manifold, by adapting a geodesic formula to the Stiefel manifold in the context of the independent subspace analysis [11]. Yamada and Ezaki adopted the dual Cayley parameterization technique to decompose the orthogonal matrix optimization problem into a pair of simple constraint-free optimization problems, and validated its applicability to the ICA [28]. Abrudan *et al.* derived an SD [8] and a CG algorithm [9] with geodesic search methods to minimize the JADE criterion on the Lie group of unitary matrices. Selvan *et al.* hybridized a Riemannian QN approach and a PSO employing Lie group techniques, in an attempt to produce a near-global-optimum solution, while minimizing a non-convex ICA contrast function [10]. Absil and Gallivan broke with tradition in proposing a strategy on $\mathcal{OB}$ for the ICA based on joint diagonalization by employing a Riemannian trust-region (RTR) approach; the efficacy of this non-orthogonal ICA for high accuracy source separation was evidenced by the numerical experiments [17]. Shen and Hüper followed suit to optimize joint diagonalization cost functions on $\mathcal{OB}$ by devising a family of block Jacobi-type methods [18].

1.3. Contribution

The scope of this paper is to design Riemannian unconstrained optimizers—SD, CG and QN—which exploit the differential geometry methods associated with $\mathcal{OB}$, to optimize a source adaptive contrast function that can be evaluated with an efficient non-parametric density estimation tool. We were inspired by the work in [17] where an $\mathcal{OB}$ optimization, namely, RTR approach, is introduced for the ICA implementation via joint diagonalization, that involves the techniques to project the Euclidean gradient of the contrast function onto a tangent space and to establish a correspondence between tangent vectors and points on the manifold using a retraction. Of late, there has been a growing interest in many real applications to seek for a non-orthogonal ICA unmixing matrix to better separate the observed data; face recognition [29], hyperspectral image classification [30], anomaly detection in hyperspectral imagery [31] and noisy data separation [32] are a few examples. Albeit the potential applications of non-orthogonal ICA, to the best of our knowledge, but for the RTR approach on $\mathcal{OB}$, other unconstrained optimizers tailored to $\mathcal{OB}$ were not reported in the literature. Even though RTR methods address some of the shortcomings of the Newton method such as lack of global convergence and prohibitive numerical cost of solving the Newton equation, they suffer from algorithmic complexity and may not perform ideally on all problems [33]. We were motivated to propose an SD method on $\mathcal{OB}$, since it always converges to a local minimum when the step-size selection is subject to the Armijo's rule [34], although asymptotically the rate of convergence is only linear. Moreover, since obtaining the second-order information through the evaluation of Hessian is very expensive for our contrast function, we have developed $\mathcal{OB}$ optimization methods, namely, CG with Hager-Zhang (HZ) or a hybrid update parameter, QN—in the spirit of [35], adapted to the ICA problem—and cost effective QN (QN-CE), that provide lower-cost numerical iterations and stronger global convergence properties than the Newton iteration by approximating the second-order properties to obtain super-linear local convergence. The original contributions of our paper are consolidated below.

1. We have designed an SD algorithm with the Armijo's step-size on $\mathcal{OB}$ for the ICA application that employs the notions of gradient vector projection onto the manifold tangent space and a retraction.
2. For the same task, super-linear algorithms—CG and QN—on $\mathcal{OB}$ are developed with the following ingredients associated with the manifold of interest:

(a) use of a vector transport on $\mathcal{OB}$ which has computationally similar expense, compared to its equivalent classical concept in the manifold optimization termed as parallel transport;
(b) derivation and application of a corresponding inverse vector transport;
(c) appropriate operations to the Hessian that involve vector transport operation back and forth between the subsequent iterates.

2. Ingredients of oblique manifold optimization

Formally $\mathcal{OB}(n, d)$ is defined as the set of all $n \times d$ matrices whose columns have unit Euclidean norm

$$\mathcal{OB}(n,d) = \left\{ \mathbf{X} \in \mathbb{R}^{n \times d} : \mathrm{ddiag}(\mathbf{X}^{\mathrm{T}}\mathbf{X}) = \mathbf{I}_d, \mathrm{rk}(\mathbf{X}) = d \right\}, \tag{1}$$

where $\mathrm{ddiag}(\cdot)$ represents the diagonal matrix whose diagonal elements are those of the matrix in the argument, $\mathrm{rk}(\cdot)$ is the rank of the matrix of interest and $\mathbf{I}_d$ is the $d \times d$ identity matrix. For $n = d$, Eq. (1) coincides with the definition of $\mathcal{OB}$ found in [36]; it is of particular interest to our ICA application where the assumption that the number of ICs is equal to the number of observed mixtures holds. Clearly, $\mathcal{OB}$ can be viewed as an embedded Riemannian manifold of $\mathbb{R}^{n \times d}$, which allows one to define the canonical inner product

$$\langle \mathbf{Z}_1, \mathbf{Z}_2 \rangle := \mathrm{trace}(\mathbf{Z}_1^{\mathrm{T}} \mathbf{Z}_2) \tag{2}$$

in $\mathbb{R}^{n \times d}$, where $\mathrm{trace}(\cdot)$ is the square Frobenius norm defined as the sum of the squares of the elements of the matrix under consideration. For completeness sake, we briefly present the orthogonal projection onto a tangent space to $\mathcal{OB}$ and a retraction, which are introduced in [17]. The tangent space is defined as

$$T_{\mathbf{X}}\mathcal{OB} = \left\{ \mathbf{Z} : \mathrm{ddiag}(\mathbf{X}^{\mathrm{T}}\mathbf{Z}) = 0 \right\}; \tag{3}$$

it means that ${\mathbf{x}^{(i)}}^{\mathrm{T}} \mathbf{z}^{(i)} = 0$, $i = 1, 2, \ldots, d$, where $\mathbf{x}^{(i)}$ and $\mathbf{z}^{(i)}$ are the ith columns of $\mathbf{X}$ and $\mathbf{Z}$, respectively. The orthogonal projection of $\mathbf{Z} \in T_{\mathbf{X}}\mathbb{R}^{n \times d}$ onto the tangent space to $\mathcal{OB}$ at $\mathbf{X}$, $T_{\mathbf{X}}\mathcal{OB}$, can be expressed as

$$\mathrm{P}_{T_{\mathbf{X}}}(\mathbf{Z}) = \mathbf{Z} - \mathbf{X}\mathrm{ddiag}(\mathbf{X}^{\mathrm{T}}\mathbf{Z}). \tag{4}$$

Notice that $\mathcal{OB}$ is isometric to the product of d unit spheres in $\mathbb{R}^n$, thanks to the fact that $\mathcal{OB}$ is endowed with a Riemannian metric. Therefore, the

various ingredients of optimization on $\mathcal{OB}$ can be conveniently derived first for the unit sphere $\mathcal{S}^{n-1}$ embedded in $\mathbb{R}^n$ and then extended to $\mathcal{OB}(n,d)$.

Riemannian optimization approaches are currently being investigated to assess the interplay between the choice of retraction/transport and the efficiency and effectiveness of the resulting algorithm. For some manifolds, e.g., the Stiefel manifold, parallel transport and the exponential map are more costly computationally than many vector transports and their associated retractions. In this case, vector transport-based algorithms can be more efficient as long as their convergence does not degrade significantly compared to the convergence of the parallel transport-based algorithm. For some manifolds, e.g., $\mathcal{S}^{n-1}$, parallel transport and the exponential map are only moderately more costly than efficient implementations of some vector transports and retractions. However, it is becoming more apparent in the literature that some vector transport/retraction-based algorithms have superior convergence behavior and the moderate computational savings on each iteration are amplified to significant savings overall due to fewer iterations (see [33] and [37] for details). We have observed this phenomenon in our experiments.

While optimizing on $\mathcal{S}^{n-1}$, the kth iterate $\mathbf{x}_k$ must move along some direction, $\boldsymbol{\xi}_k$, e.g., the steepest descent direction, such that the next iterate $\mathbf{x}_{k+1}$ is "locked" to the manifold; this is achieved by a retraction

$$R_{\mathbf{x}_k}(\boldsymbol{\xi}_k) := \frac{\mathbf{x}_k + \alpha_k \boldsymbol{\xi}_k}{\|\mathbf{x}_k + \alpha_k \boldsymbol{\xi}_k\|}, \tag{5}$$

where $\|\cdot\|$ denotes the Euclidean norm. For the sake of simplicity, the steplength α_k is assumed to be 1 in the sequel. A special case of retraction for $\mathcal{S}^{n-1}$ is the standard Riemannian exponential map [38]

$$\exp|_{\mathbf{x}_k}(\boldsymbol{\xi}_k) := \mathbf{x}_k \cos\|\boldsymbol{\xi}_k\| + \frac{\boldsymbol{\xi}_k}{\|\boldsymbol{\xi}_k\|} \sin\|\boldsymbol{\xi}_k\|; \tag{6}$$

notice that there is similar computational complexity compared to the retraction given in Eq. (5). By generalizing either Eq. (5) or (6) for the product of d unit spheres i.e., $\mathcal{OB}(n,d)$, one can establish a correspondence between the tangent vectors and points on $\mathcal{OB}$.

During each iterative step of the optimization process, the evaluation of the next steepest descent direction involves the tangent vectors pertaining to the current and the subsequent iterations; this necessitates the notion of vector transport proposed in [33], which provides a mechanism to move a tangent vector between the tangent spaces corresponding to two different

iterates. The conventional Levi-Civita parallel transport [39] of $\boldsymbol{\lambda}_k \in T_{\mathbf{x}_k}\mathcal{S}$ along the geodesic, ζ, from $\mathbf{x}_k$ in the direction of $\boldsymbol{\xi}_k \in T_{\mathbf{x}_k}\mathcal{S}$ is given by

$$P_{\zeta}^{t\leftarrow 0}\boldsymbol{\lambda}_k = \left\{\mathbf{I} + [\cos(\|\boldsymbol{\xi}_k\|t) - 1]\,\frac{\boldsymbol{\xi}_k\boldsymbol{\xi}_k^{\mathrm{T}}}{\|\boldsymbol{\xi}_k\|^2} - \sin(\|\boldsymbol{\xi}_k\|t)\frac{\mathbf{x}_k\boldsymbol{\xi}_k^{\mathrm{T}}}{\|\boldsymbol{\xi}_k\|}\right\}\boldsymbol{\lambda}_k. \tag{7}$$

Since the manifold is endowed with a retraction R, one possible choice of vector transport[1] for $\mathcal{S}^{n-1}$ is given by

$$\mathcal{T}_{\boldsymbol{\xi}_k}(\boldsymbol{\lambda}_k) = \left[\mathbf{I} - \frac{(\mathbf{x}_k + \boldsymbol{\xi}_k)(\mathbf{x}_k + \boldsymbol{\xi}_k)^{\mathrm{T}}}{\|\mathbf{x}_k + \boldsymbol{\xi}_k\|^2}\right]\boldsymbol{\lambda}_k, \tag{8}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix, $\mathbf{x}_k \in \mathcal{S}^{n-1}$ and $\boldsymbol{\xi}_k, \boldsymbol{\lambda}_k \in T_{\mathbf{x}_k}\mathcal{S}$; it is straightforward to extend the formula for $\mathcal{OB}$. Notice that the computational complexity of parallel transport on $\mathcal{S}^{n-1}$ is similar to the cost of vector transport on $\mathcal{S}^{n-1}$.

By defining $\mathbf{v}_k = \mathbf{x}_k + \boldsymbol{\xi}_k$, the retraction in Eq. (5) is written as

$$R_{\mathbf{x}_k}(\boldsymbol{\xi}_k) = \frac{\mathbf{v}_k}{\|\mathbf{v}_k\|} = \mathbf{x}_{k+1}. \tag{9}$$

Now applying the vector transport operator on $\boldsymbol{\lambda}_k \in T_{\mathbf{x}_k}\mathcal{S}$ from Eq. (8) yields

$$\begin{aligned}\mathcal{T}_{\boldsymbol{\xi}_k}(\boldsymbol{\lambda}_k) &= \left(\mathbf{I} - \frac{\mathbf{v}_k\mathbf{v}_k^{\mathrm{T}}}{\|\mathbf{v}_k\|^2}\right)\boldsymbol{\lambda}_k \\ &= (\mathbf{I} - \mathbf{x}_{k+1}\mathbf{x}_{k+1}^{\mathrm{T}})\boldsymbol{\lambda}_k.\end{aligned}$$

It can easily be verified that $\boldsymbol{\lambda}_k$ is transported to the tangent space $T_{\mathbf{x}_{k+1}}\mathcal{S}$ due to the operation in Eq. (8) as follows:

$$\begin{aligned}\mathbf{x}_{k+1}^{\mathrm{T}}\mathcal{T}_{\boldsymbol{\xi}_k}(\boldsymbol{\lambda}_k) &= \mathbf{x}_{k+1}^{\mathrm{T}}(\mathbf{I} - \mathbf{x}_{k+1}\mathbf{x}_{k+1}^{\mathrm{T}})\boldsymbol{\lambda}_k \\ &= \mathbf{x}_{k+1}^{\mathrm{T}}\boldsymbol{\lambda}_k - \mathbf{x}_{k+1}^{\mathrm{T}}\mathbf{x}_{k+1}\mathbf{x}_{k+1}^{\mathrm{T}}\boldsymbol{\lambda}_k = 0.\end{aligned}$$

Therefore, we conclude that $\mathcal{T}_{\boldsymbol{\xi}_k}$ is a map $T_{\mathbf{x}_k} \rightarrow T_{\mathbf{x}_{k+1}}$.

When a vector transport exists, the inverse of the linear map $\mathcal{T}_{\boldsymbol{\xi}_k}$ is denoted by $\mathcal{T}_{\boldsymbol{\xi}_k}^{-1}$; in order to make sense, both a vector and an inverse vector

[1] Eq. (8) means that any arbitrary element $\boldsymbol{\lambda}_k$ given in the argument, belonging to the tangent space $T_{\mathbf{x}_k}\mathcal{S}$ to the manifold $\mathcal{S}$ at the incumbent iterate $\mathbf{x}_k$, is transported by the operator $\mathcal{T}$ to the tangent space $T_{\mathbf{x}_{k+1}}\mathcal{S}$ at the next iterate $\mathbf{x}_{k+1}$; note that $\mathbf{x}_{k+1}$ is obtained via a retraction involving $\mathbf{x}_k$ and another element $\boldsymbol{\xi}_k$ in $T_{\mathbf{x}_k}\mathcal{S}$ which is denoted in the subscript.

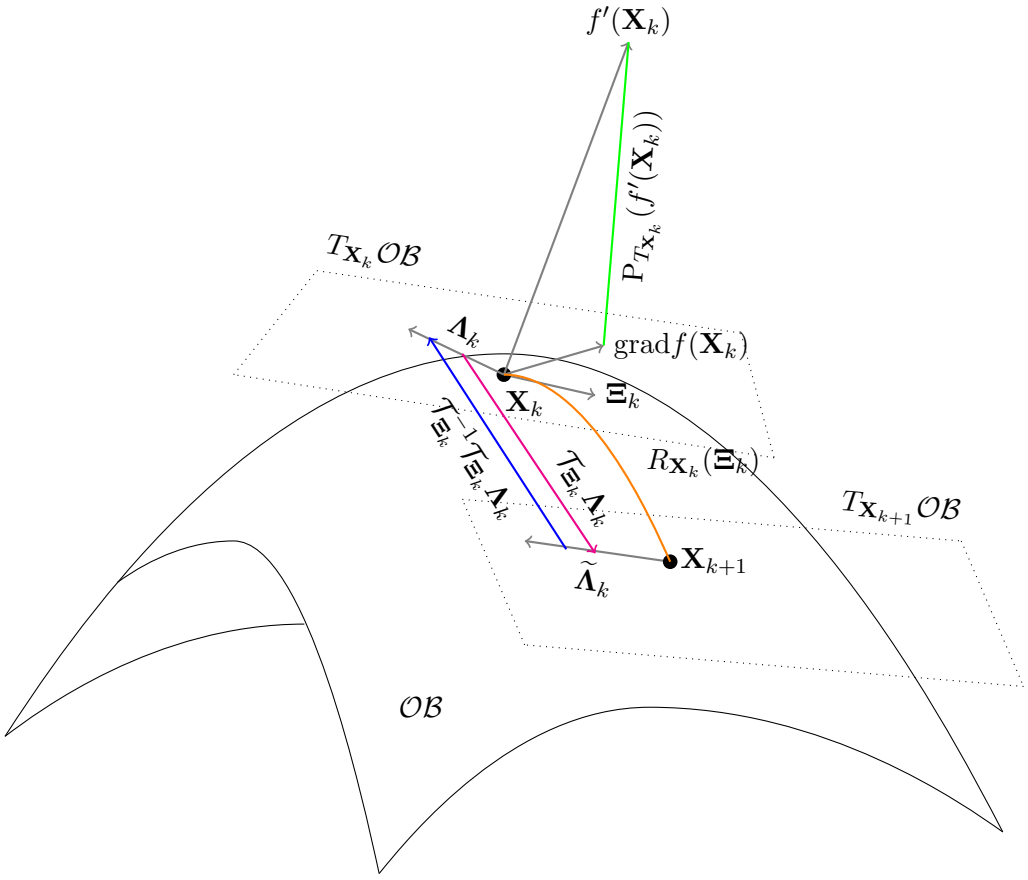

Figure 1: Various mechanisms involved in $\mathcal{OB}$ optimization algorithms. The Euclidean gradient evaluated at $\mathbf{X}_k = \left[\mathbf{x}_k^{(1)}\mathbf{x}_k^{(2)}\cdots\mathbf{x}_k^{(d)}\right] \in \mathcal{OB}$, $f'(\mathbf{X}_k) \in T_{\mathbf{X}_k}\mathbb{R}^{n\times d}$, is projected onto the tangent space to $\mathcal{OB}$ at $\mathbf{X}_k$, $T_{\mathbf{X}_k}\mathcal{OB}$, represented by the green line. Given the search direction $\boldsymbol{\Xi}_k = \left[\boldsymbol{\xi}_k^{(1)}\boldsymbol{\xi}_k^{(2)}\cdots\boldsymbol{\xi}_k^{(d)}\right]$, a retraction $R_{\mathbf{X}_k}(\boldsymbol{\Xi}_k)$, designated by the orange arc, "locks" the subsequent iterate $\mathbf{X}_{k+1}$ to the constraint surface $\mathcal{OB}$. The magenta arrow typifies the role of a vector transport $\mathcal{T}_{\boldsymbol{\Xi}_k}$ which moves an arbitrary tangent vector $\boldsymbol{\Lambda}_k = \left[\boldsymbol{\lambda}_k^{(1)}\boldsymbol{\lambda}_k^{(2)}\cdots\boldsymbol{\lambda}_k^{(d)}\right]$ from $T_{\mathbf{X}_k}\mathcal{OB}$ to $T_{\mathbf{X}_{k+1}}\mathcal{OB}$; the reverse operation, namely, inverse vector transport $\mathcal{T}_{\boldsymbol{\Xi}_k}^{-1}$ is a map $T_{\mathbf{X}_{k+1}} \to T_{\mathbf{X}_k}$ that enables the transport of $\widetilde{\boldsymbol{\Lambda}}_k = \left[\widetilde{\boldsymbol{\lambda}}_k^{(1)}\widetilde{\boldsymbol{\lambda}}_k^{(2)}\cdots\widetilde{\boldsymbol{\lambda}}_k^{(d)}\right] \in T_{\mathbf{X}_{k+1}}\mathcal{OB}$ to $T_{\mathbf{X}_k}\mathcal{OB}$ as symbolized by the blue arrow.

transport operator must be considered together. It is essential to derive the corresponding inverse vector transport for realizing a QN algorithm on $\mathcal{OB}$. One may refer to Appendix A for a formal derivation of an inverse vector transport, and Fig. 1 for comprehending the role played by each $\mathcal{OB}$ optimization ingredient. A suitable choice for the inverse vector transport on $\mathcal{S}_{n-1}$ is

$$\mathcal{T}^{-1}_{\boldsymbol{\xi}_k}(\widetilde{\boldsymbol{\lambda}}_k) = \left[\mathbf{I} - \frac{(\mathbf{x}_k + \boldsymbol{\xi}_k)\mathbf{x}_k^{\mathrm{T}}}{\mathbf{x}_k^{\mathrm{T}}(\mathbf{x}_k + \boldsymbol{\xi}_k)}\right] \widetilde{\boldsymbol{\lambda}}_k, \tag{10}$$

where $\widetilde{\boldsymbol{\lambda}}_k \in T_{\mathbf{x}_{k+1}}\mathcal{S}$, which can be justified in a manner similar to the vector transport and easily adapted for $\mathcal{OB}$. We can express the operation of the inverse vector transport on $\widetilde{\boldsymbol{\lambda}}_k \in T_{\mathbf{x}_{k+1}}\mathcal{S}$ as follows[2]:

$$\begin{aligned}
\mathcal{T}^{-1}_{\boldsymbol{\xi}_k}(\widetilde{\boldsymbol{\lambda}}_k) &= \left[\mathbf{I} - (\mathbf{x}_k^{\mathrm{T}}\mathbf{v}_k)^{-1}\mathbf{v}_k\mathbf{x}_k^{\mathrm{T}}\right] \widetilde{\boldsymbol{\lambda}}_k \\
&= \left[\mathbf{I} - (\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k+1})^{-1}\mathbf{x}_{k+1}\mathbf{x}_k^{\mathrm{T}}\right] \widetilde{\boldsymbol{\lambda}}_k.
\end{aligned}$$

It is trivial to demonstrate that $\widetilde{\boldsymbol{\lambda}}_k$ is transported back to $T_{\mathbf{x}_k}\mathcal{S}$:

$$\begin{aligned}
\mathbf{x}_k^{\mathrm{T}}\mathcal{T}^{-1}_{\boldsymbol{\xi}_k}(\widetilde{\boldsymbol{\lambda}}_k) &= \mathbf{x}_k^{\mathrm{T}}\left[\mathbf{I} - (\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k+1})^{-1}\mathbf{x}_{k+1}\mathbf{x}_k^{\mathrm{T}}\right] \widetilde{\boldsymbol{\lambda}}_k \\
&= \mathbf{x}_k^{\mathrm{T}}\widetilde{\boldsymbol{\lambda}}_k - (\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k+1})^{-1}\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k+1}\mathbf{x}_k^{\mathrm{T}}\widetilde{\boldsymbol{\lambda}}_k \\
&= \mathbf{x}_k^{\mathrm{T}}\widetilde{\boldsymbol{\lambda}}_k - \mathbf{x}_k^{\mathrm{T}}\widetilde{\boldsymbol{\lambda}}_k = 0,
\end{aligned}$$

which implies that $\mathcal{T}^{-1}_{\boldsymbol{\xi}_k}$ is a map $T_{\mathbf{x}_{k+1}} \to T_{\mathbf{x}_k}$. We have chosen a particular vector/inverse vector transport pair; admittedly, there are other potential choices such as two oblique projectors, which may have more satisfactory theoretical properties, and are under investigation.

3. Evaluation of contrast function and its derivative

Assuming the conventional generative ICA model with d independent and stationary sources $a^{(1)}, a^{(2)}, \ldots, a^{(d)}$ mixed by an unknown, full-rank mixing matrix $\mathbf{W}$, the observed mixtures are expressed as $\mathbf{m} = \mathbf{W}\mathbf{a}$. The original sources are recovered by minimizing the MI between the estimated sources $\mathbf{b} = \mathbf{X}^{\mathrm{T}}\mathbf{m}$, given by

$$f(\mathbf{X}) = \sum_{i=1}^{d} H\left(b^{(i)}\right) - H\left(b^{(1)}, b^{(2)}, \ldots, b^{(d)}\right), \tag{11}$$

[2]Throughout this paper, the tilde symbolizes that the vector has already been transported from one tangent space of the manifold to another.

where $H\left(b^{(i)}\right)$ stands for the Shannon entropy of $b^{(i)}$, $H\left(b^{(1)}, b^{(2)}, \ldots, b^{(d)}\right)$ denotes the joint entropy of $b^{(1)}, b^{(2)}, \ldots, b^{(d)}$ or precisely the entropy of the vector $\mathbf{b}$ and $\mathbf{X}$ is the unmixing matrix to be determined. Since $H(\mathbf{b}) = H(\mathbf{m}) + \log|\det \mathbf{X}|$ and the term $H(\mathbf{m})$ does not depend on $\mathbf{X}$, the contrast function can be stated as follows

$$\begin{aligned} f(\mathbf{X}) &= \sum_{i=1}^{d} H\left(b^{(i)}\right) - \log|\det \mathbf{X}| \\ &= -\sum_{i=1}^{d} E\left\{\log p^{(i)}\left(b^{(i)}\right)\right\} - \log|\det \mathbf{X}|, \end{aligned} \tag{12}$$

which does not involve $H(\mathbf{m})$. Here $E\{\cdot\}$ is the expectation operator, the symbol log stands for the natural logarithm and $p^{(i)}(\eta)$ is the marginal probability density function (PDF) that can be estimated using the Parzen window

$$p^{(i)}(\eta) \simeq \frac{1}{N}\sum_{v=1}^{N} \frac{1}{h\sqrt{2\pi}} \exp\left[\frac{-\left(\eta - b_v^{(i)}\right)^2}{2h^2}\right], \tag{13}$$

where $\mathbf{b}^{(i)} = \left[b_1^{(i)}, b_2^{(i)}, \ldots, b_N^{(i)}\right] = \mathbf{x}^{(i)^{\mathrm{T}}}\mathbf{M}$, $i = 1, 2, \ldots, d$, with $\mathbf{M}$ being the $d \times N$ matrix containing the observed data, and the bandwidth h is optimally selected as a function of the sample size N [40]

$$h = 1.06N^{-\frac{1}{5}} \tag{14}$$

assuming unit standard deviation[3] of $b^{(i)}$, thereby simplifying the design of the kernel density estimator. Observe that this criterion obviates the need for estimating the joint density [21].

In many offline source separation applications involving images or audio signals the major emphasis is on the accurate estimation of the unmixing matrix, despite the computational overhead. Therefore, we have resorted to the direct estimation of the statistical independence with the aid of Parzen window density estimation for evaluating the contrast function and its first derivative. This method adapts itself fully to the particularity of the source

[3]Eq. (14) can be generalized for any standard deviation as $h = 1.06\hat{\sigma}N^{-\frac{1}{5}}$, where an estimate of the standard deviation, $\hat{\sigma}$, can easily be obtained as $\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{v=1}^{N} b_v^2}$; note that the dependence on the component index is omitted for the sake of simplicity.

distributions, since everything is estimated from the data [21]. Unlike the approaches whose computational complexity is ducked by implementing a simple gradient algorithm, wherein some "surrogate" functions substitute the unknown score functions in this gradient (refer, e.g., [13, p. 185]), the Parzen window density estimation uses the recently proposed improved fast Gauss transform (IFGT) to efficiently evaluate the sum of Gaussians in higher dimensions, compared to the fast Gauss transform (FGT) in which case the computational complexity and storage grow exponentially with dimension. The crux of the IFGT algorithm is subdividing the d-dimensional space using a k-center-clustering-based geometric data structure, followed by building a truncated representation of kernels inside each cluster using a set of decaying basis functions. In our approach, direct evaluation of the contrast function and its derivative is simultaneously carried out; in consequence, the overall computational load is lessened by performing the k-center clustering only once at each iterative step. Additionally, akin to the work of Boscolo *et al.* [2], it precludes the need to separate the optimization step from the step involving the re-estimation of the score functions as in [41]. As claimed in [42], the IFGT reduces the computational complexity into linear time. Moreover, as opposed to the kernel density estimators based on the FFT which is limited to evaluating the density estimates of the gridded data, the IFGT does not require the data to be on grids. Given N source data-points, the direct evaluation of densities at M target points takes only $O(N)$ time for the IFGT, where $M = N$ specific to our case applies; it is significantly less compared to the $O(N \log N)$ time taken by the FFT. For a detailed treatment on the IFGT for efficient kernel density estimation, one may refer to [42] and [43]. The closed-form expression for the gradient of

the contrast function, derived in Appendix B, is furnished below:

$$\begin{aligned}\frac{\partial f}{\partial x_{rs}} =& \frac{1}{Nh^2}\sum_{u=1}^{N}\frac{1}{\sum_{v=1}^{N}\exp\left[\frac{-\left(b_u^{(r)}-b_v^{(r)}\right)^2}{2h^2}\right]} \\ & \left\{ b_u^{(r)} m_u^{(s)} \sum_{v=1}^{N}\exp\left[\frac{-\left(b_u^{(r)}-b_v^{(r)}\right)^2}{2h^2}\right]\right. \\ & - b_u^{(r)}\sum_{v=1}^{N} m_v^{(s)}\exp\left[\frac{-\left(b_u^{(r)}-b_v^{(r)}\right)^2}{2h^2}\right] \\ & - m_u^{(s)}\sum_{v=1}^{N} b_v^{(r)}\exp\left[\frac{-\left(b_u^{(r)}-b_v^{(r)}\right)^2}{2h^2}\right] \\ & \left. + \sum_{v=1}^{N} b_v^{(r)} m_v^{(s)}\exp\left[\frac{-\left(b_u^{(r)}-b_v^{(r)}\right)^2}{2h^2}\right]\right\} \\ & - [(\mathbf{X}^{\mathrm{T}})^{-1}]_{rs}.\end{aligned} \tag{15}$$

4. ICA learning on oblique manifold

4.1. Steepest descent on oblique manifold

Albeit its linear convergence rate, the reason for attempting the SD algorithm is that if coupled with the Armijo's step-size rate [44], it almost always converges to a local minimum. In the following, the ICA learning on $\mathcal{OB}$ employing an SD algorithm is discussed in detail; one may refer to Table 1 for the implementation steps. A non-orthogonal unmixing matrix $\mathbf{X}_0$ of size $d \times d$, where d is the dimension of the multivariate data being unmixed with its columns normalized, is randomly chosen to be the initial seed for the optimization algorithm. The cost $f(\mathbf{X}_0)$ and the Euclidean gradient $f'(\mathbf{X}_0)$ are evaluated at $\mathbf{X}_0$ using Eqs. (12) and (15), respectively. Unlike the Euclidean SD algorithm, $f'(\mathbf{X}_0)$ has to be projected onto the tangent space to $\mathcal{OB}$ at the current iterate $\mathbf{X}_0$, which is computed as

$$\mathrm{grad} f(\mathbf{X}_0) = \mathrm{P}_{T_{\mathbf{X}_0}}(f'(\mathbf{X}_0)) = f'(\mathbf{X}_0) - \mathbf{X}_0 \mathrm{ddiag}(\mathbf{X}_0^{\mathrm{T}} f'(\mathbf{X}_0)). \tag{16}$$

At the initial iteration, the step-size α_0 and the search direction $\boldsymbol{\Xi}_0$ are set to be 1 and $-\text{grad}f(\mathbf{X}_0)$, respectively, provided that the convergence criterion is not met. If the iteration number $k > 0$, the step-size α_k is to be determined by successively reducing $\alpha_{k_{\text{guess}}} = 1$ by a factor γ, termed as the backtracking line-search, until it satisfies the Armijo step-size rule [44]

$$f(\mathbf{X}_{k+1}) - f(\mathbf{X}_k) \leq -\frac{1}{2}\alpha_k \|\text{vec}(\text{grad}f(\mathbf{X}_k))\|^2. \tag{17}$$

Here, $\text{vec}(\cdot)$ denotes generating a column vector by concatenating the columns of the matrix in the argument, and the reverse operation is $\text{unvec}(\cdot)$. The next iterate $\mathbf{X}_{k+1}$ always lies on $\mathcal{OB}$, since we use a retraction to determine $\mathbf{X}_{k+1}$ as

$$R_{\mathbf{X}_k}(\boldsymbol{\Xi}_k) = (\mathbf{X}_k + \alpha_k \boldsymbol{\Xi}_k)(\text{ddiag}((\mathbf{X}_k + \alpha_k \boldsymbol{\Xi}_k)^{\mathrm{T}}(\mathbf{X}_k + \alpha_k \boldsymbol{\Xi}_k)))^{-\frac{1}{2}}. \tag{18}$$

The algorithm is terminated when

$$\|\text{vec}(\text{grad}f(\mathbf{X}_{k+1}))\|_\infty < \epsilon(1 + \|\text{vec}(\text{grad}f(\mathbf{X}_0))\|_\infty), \tag{19}$$

where $\|\cdot\|_\infty$ denotes the infinity norm, and $\epsilon > 0$ is a preset threshold value. Even though in most of the experimented cases the SD algorithm with the values of $\alpha_0, \alpha_{k_{\text{guess}}} = 1$ performs satisfactorily, as cautioned in [44, p. 58], it may happen that the SD and CG methods do not produce well-scaled search directions; nonetheless, an expedient to circumvent this drawback is to use the current information about the problem and the algorithm to make the initial guess for the step-size

$$\alpha_0 = \frac{1}{\|\text{vec}(\text{grad}f(\mathbf{X}_0))\|} \tag{20}$$

as recommended in [45], and

$$\alpha_{k_{\text{guess}}} = \frac{\alpha_{k-1}((\text{vec}(\text{grad}f(\mathbf{X}_{k-1})))^{\mathrm{T}}\text{vec}(\boldsymbol{\Xi}_{k-1}))}{(\text{vec}(\text{grad}f(\mathbf{X}_k)))^{\mathrm{T}}\text{vec}(\boldsymbol{\Xi}_k)}. \tag{21}$$

4.2. Conjugate gradient on oblique manifold

In our ICA formulation, evaluating the second-order information through a closed-form Hessian is prohibitively expensive. Therefore, instead of merely being content with the SD algorithm having linear convergence, it is possible to approximate the second derivatives by "comparing" the first-order information—tangent vectors—at distinct points on $\mathcal{OB}$ [33, p. 169]. Since we

Table 1: Steepest descent algorithm on oblique manifold.

1. Select a random initial seed $\mathbf{X}_0$ with the normality constraint, and evaluate $f(\mathbf{X}_0)$ and $f'(\mathbf{X}_0)$ found in Eqs. (12) and (15), respectively; project $f'(\mathbf{X}_0)$ onto $T_{\mathbf{X}_0}\mathcal{OB}$ as in Eq. (16) to obtain $\mathrm{grad}f(\mathbf{X}_0)$ and form $\mathbf{\Xi}_0$.
2. Accept α_k which satisfies the Armijo step-size rule in Eq. (17) using the backtracking line-search.
3. Make use of a retraction in Eq. (18) to generate $\mathbf{X}_{k+1}$.
4. Compute $f(\mathbf{X}_{k+1})$ and $f'(\mathbf{X}_{k+1})$.
5. Evaluate $\mathrm{grad}f(\mathbf{X}_{k+1})$ and decide $\mathbf{\Xi}_{k+1}$.
6. Subject to the convergence criterion in Eq. (19), either terminate or divert to Step 2.

are endowed with a vector transport on $\mathcal{OB}$, we can transport an arbitrary tangent vector $\mathbf{\Lambda}$ from a point $\mathbf{X} \in \mathcal{OB}$ to another point $R_{\mathbf{X}}(\mathbf{\Xi}) \in \mathcal{OB}$. This information is sufficient to design a CG algorithm on $\mathcal{OB}$, which is characterized by low memory requirement and strong local and global convergence properties [46]. A CG algorithm on $\mathcal{OB}$ to be applicable for our ICA problem is detailed below, along with the step-wise realization procedure as summarized in Table 3.

The initial seed $\mathbf{X}_0$, as described earlier, is randomly selected to evaluate $f(\mathbf{X}_0)$ and $f'(\mathbf{X}_0)$ based on Eqs. (12) and (15), respectively. As shown in Eq. (16), the tangent space gradient $\mathrm{grad}f(\mathbf{X}_0)$ on $\mathcal{OB}$ at $\mathbf{X}_0$ is computed, that enables the initial search direction to be set as $\mathbf{\Xi}_0 = -\mathrm{grad}f(\mathbf{X}_0)$. Due to the concern raised by the failure to produce well-scaled search directions, the initial step-size guesses α_0 and $\alpha_{k_{\mathrm{guess}}}$ are calculated as furnished in Eqs. (20) and (21), respectively. In contrast to the SD algorithm, in the case of CG, it is not trivial to determine the search directions $\mathbf{\Xi}_{k+1}$ which are always descent; conversely, to ensure global convergence, the determination of α_k demands an exact line-search. We have cautiously adopted the update parameter β_{k+1}—HZ [47] or a hybrid value [48] derived from Dai-Yuan (DY) [49] and Hestenes-Stiefel (HS) [50]—that appears in the expression for $\mathbf{\Xi}_{k+1}$, in which case it is sufficient if α_k satisfies the weak Wolfe conditions [44, p. 37]

$$f(\mathbf{X}_{k+1}) \leq f(\mathbf{X}_k) + c_1\alpha_k(\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_k)))^{\mathrm{T}}\mathrm{vec}(\mathbf{\Xi}_k) \tag{22}$$

$$(\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1})))^{\mathrm{T}}\widetilde{\mathbf{\Xi}}_k \geq c_2(\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_k)))^{\mathrm{T}}\mathrm{vec}(\mathbf{\Xi}_k), \tag{23}$$

where $0 < c_1 < c_2 < 1$. We perform an inexact line-search iterative technique employing the cubic/quadratic polynomial method as proposed in [51] on $\mathcal{OB}$ to estimate α_k. Note that while evaluating the product of $\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1}))$ and $\mathbf{\Xi}_k$ to verify the curvature condition (Eq. (23)) by Wolfe, we confront a foreseen difficulty since $\mathrm{grad}f(\mathbf{X}_{k+1})$ corresponds to

the tangent space to $\mathcal{OB}$ at $\mathbf{X}_{k+1}$, $T_{\mathbf{X}_{k+1}}\mathcal{OB}$, whereas $\boldsymbol{\Xi}_k$ is an element belonging to the tangent space to $\mathcal{OB}$ at $\mathbf{X}_k$, $T_{\mathbf{X}_k}\mathcal{OB}$. Favorably, the vector transport mechanism defined on $\mathcal{OB}$ renders a better means to deal with such situations, wherein a mathematical operation between the quantities pertaining to two different points on the manifold has to be carried out. As a result, we first transport the tangent vector $\boldsymbol{\Xi}_k$ from $T_{\mathbf{X}_k}\mathcal{OB}$ to $T_{\mathbf{X}_{k+1}}\mathcal{OB}$ following

$$\mathcal{T}_{\alpha_k\boldsymbol{\xi}_k^{(i)}}(\boldsymbol{\xi}_k^{(i)}) = \left[\mathbf{I} - \frac{(\mathbf{x}_k^{(i)} + \alpha_k\boldsymbol{\xi}_k^{(i)})(\mathbf{x}_k^{(i)} + \alpha_k\boldsymbol{\xi}_k^{(i)})^{\mathrm{T}}}{\|\mathbf{x}_k^{(i)} + \alpha_k\boldsymbol{\xi}_k^{(i)}\|^2}\right]\boldsymbol{\xi}_k^{(i)} \tag{24}$$

and then multiply $\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1})) \in T_{\mathbf{X}_{k+1}}\mathcal{OB}$ with $\widetilde{\boldsymbol{\Xi}}_k = \mathrm{vec}(\mathcal{T}_{\alpha_k\boldsymbol{\Xi}_k}(\boldsymbol{\Xi}_k)) \in T_{\mathbf{X}_{k+1}}\mathcal{OB}$. Similarly, in the Euclidean case expression for determining $\boldsymbol{\Xi}_{k+1}$

$$\boldsymbol{\Xi}_{k+1} = -f'(\mathbf{X}_{k+1}) + \beta_{k+1}\boldsymbol{\Xi}_k \tag{25}$$

the use of $\mathcal{T}_{\alpha_k\boldsymbol{\Xi}_k}(\boldsymbol{\Xi}_k) \in T_{\mathbf{X}_{k+1}}\mathcal{OB}$ instead of $\boldsymbol{\Xi}_k \in T_{\mathbf{X}_k}\mathcal{OB}$ is warranted to adapt to the manifold case, stated as

$$\boldsymbol{\Xi}_{k+1} = -\mathrm{grad}f(\mathbf{X}_{k+1}) + \beta_{k+1}\mathrm{unvec}(\widetilde{\boldsymbol{\Xi}}_k). \tag{26}$$

Clearly, in Eq. (26) all the quantities are defined in $T_{\mathbf{X}_{k+1}}\mathcal{OB}$, and thus compatible to be operated with.

As pointed out earlier, it is crucial to decide β_{k+1} that guarantees good convergence characteristics. Popular choices for β_{k+1} such as the Fletcher-Reeves (FR) and Polak-Ribière-Polyak (PRP) require a line-search of sufficient accuracy to ensure that the search directions yield descent [52]. For instance, as evidenced in [53], even the strong Wolfe condition

$$|(\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1})))^{\mathrm{T}}\widetilde{\boldsymbol{\Xi}}_k| \leq c_2|(\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_k)))^{\mathrm{T}}\mathrm{vec}(\boldsymbol{\Xi}_k)| \tag{27}$$

may not produce a descent direction unless $c_2 \leq 1/2$ for the FR scheme, which is very restrictive; as opposed to this stringent choice, in most of the practical implementations, c_2 close to 1 results in efficient performance. For the PRP method, any choice of $c_2 \in (0, 1)$ in Eq. (27) may not result in a direction of descent [47]. Therefore, in our implementation, β_{k+1} proposed by Hager and Zhang with due alterations, $\beta_{k+1}^{\mathrm{HZ}}$, is considered as specified in Table 2. Notice that $\mathbf{y}_k$ expressed as

$$\mathbf{y}_k = \mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1}) - \mathcal{T}_{\alpha_k\boldsymbol{\Xi}_k}(\mathrm{grad}f(\mathbf{X}_k))) \tag{28}$$

and $\overline{\beta}_{k+1}^{\mathrm{HZ}}$ to deduce $\beta_{k+1}^{\mathrm{HZ}}$ involve the vector transport of $\mathrm{grad}f(\mathbf{X}_k)$ and $\boldsymbol{\Xi}_k$ from $T_{\mathbf{X}_k}\mathcal{OB}$ to $T_{\mathbf{X}_{k+1}}\mathcal{OB}$. Alternatively, a hybrid method comprising DY

Table 2: Possible choices of β_{k+1} for conjugate gradient algorithm on oblique manifold.

Scheme	β_{k+1}
Hager and Zhang	$\beta_{k+1}^{\mathrm{HZ}} = \max\left\{\overline{\beta}_{k+1}^{\mathrm{HZ}}, \varphi_k\right\}$, where $\overline{\beta}_{k+1}^{\mathrm{HZ}} = \dfrac{1}{\widetilde{\boldsymbol{\Xi}}_k^{\mathrm{T}}\mathbf{y}_k}\left(\mathbf{y}_k - 2\widetilde{\boldsymbol{\Xi}}_k\dfrac{\|\mathbf{y}_k\|^2}{\widetilde{\boldsymbol{\Xi}}_k^{\mathrm{T}}\mathbf{y}_k}\right)^{\mathrm{T}}\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1}))$; $\varphi_k = \dfrac{-1}{\|\mathrm{vec}(\boldsymbol{\Xi}_k)\|\min\{\varphi, \|\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_k))\|\}}$; $\varphi > 0$.
Hybrid	$\beta_{k+1}^{\mathrm{hybrid}} = \max\{0, \min\{\beta_{k+1}^{\mathrm{HS}}, \beta_{k+1}^{\mathrm{DY}}\}\}$, where $\beta_{k+1}^{\mathrm{HS}} = \dfrac{(\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1})))^{\mathrm{T}}\mathbf{y}_k}{\widetilde{\boldsymbol{\Xi}}_k^{\mathrm{T}}\mathbf{y}_k}$; $\beta_{k+1}^{\mathrm{DY}} = \dfrac{\|\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1}))\|^2}{\widetilde{\boldsymbol{\Xi}}_k^{\mathrm{T}}\mathbf{y}_k}$.

and HS update parameters is adopted to compute β_{k+1} in our approach; the rationale behind the surpassing performance of such hybridization is briefed below. The available choices for β_{k+1} in the literature can be broadly classified into two groups: their numerator being either $\|\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1}))\|^2$ or $(\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1})))^{\mathrm{T}}\mathbf{y}_k$. The first group of methods—the FR, DY and conjugate descent (CD)—have strong convergence properties; however if they encounter a bad search direction and a tiny step, then the direction and the step corresponding to the next iterate are likely to be poor as well. This phenomenon known as "jamming" [54] will adversely affect the performance of these methods. In contrast, the second group of methods—the PRP, HS and Liu-Storey (LS)—although often perform better than the first category, suffer from poor convergence. Therefore to exploit the desirable features from each set, hybrid methods are investigated. We employ $\beta_{k+1}^{\mathrm{hybrid}}$, following [48], after introducing the necessary modifications to suit the manifold case, as found in Table 2. Given $\mathbf{X}_k$, α_k and $\boldsymbol{\Xi}_k$, the following iterate $\mathbf{X}_{k+1}$ that "lives" on $\mathcal{OB}$ can be computed by the retraction in Eq. (18). The algorithm proceeds iteratively until the solution fails to improve further, which is indicated by a relatively small value of $\|\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1}))\|_\infty$.

4.3. Quasi-Newton on oblique manifold

The QN algorithm makes use of both gradient and curvature information about the optimization landscape for exploring the solution space [55], and possesses several other advantages listed by Yang *et al.* in [56], namely, use

Table 3: Conjugate gradient algorithm on oblique manifold.

1. Start with a random initial guess $\mathbf{X}_0$ abiding by the normality constraint, and evaluate $f(\mathbf{X}_0)$ and $f'(\mathbf{X}_0)$ stated in Eqs. (12) and (15), respectively; using $\mathrm{grad}f(\mathbf{X}_0)$ in Eq. (16), determine $\boldsymbol{\Xi}_0$ and set α_0 as in Eq. (20).
2. If $k > 0$, initialize α_k as in Eq. (21).
3. Vector transport $\boldsymbol{\Xi}_k$ from $T_{\mathbf{X}_k}\mathcal{OB}$ to $T_{\mathbf{X}_{k+1}}\mathcal{OB}$ with the aid of $\mathcal{T}_{\alpha_k\boldsymbol{\Xi}_k}(\boldsymbol{\Xi}_k)$ in Eq. (24), which yields $\widetilde{\boldsymbol{\Xi}}_k$.
4. Select α_k such that the weak Wolfe conditions in Eqs. (22) and (23), are met by employing the inexact line-search.
5. Obtain $\mathbf{X}_{k+1}$ as expressed in Eq. (18), and compute $f(\mathbf{X}_{k+1})$ and $f'(\mathbf{X}_{k+1})$.
6. Evaluate $\mathrm{grad}f(\mathbf{X}_{k+1})$, and deduce $\mathbf{y}_k$ as per Eq. (28).
7. Determine β_{k+1} by either Hager-Zhang or hybrid approach as specified in Table 2.
8. Find $\boldsymbol{\Xi}_{k+1}$ in Eq. (26) for the next iteration.
9. Proceed to Step 2, unless the convergence criterion in Eq. (19) is reached.

of only gradient information to obtain the Hessian update, small computational load, super-linear convergence and superior performance while the SD methods face convergence difficulty. Furthermore, it is noteworthy to mention that even if the Hessian matrix happens to incorrectly estimate the curvature in the objective function, the Hessian approximation will tend to correct itself within a few steps [44, p. 200]. Due to the aforementioned reasons, it is worthwhile to design the optimizer suitable for $\mathcal{OB}$ minimization. The implementation issues are discussed here for a QN scheme restructured to optimize on $\mathcal{OB}$, and a concise description of the algorithm is provided in Table 4.

The random initial unmixing matrix $\mathbf{X}_0$ with unit-norm columns is used to evaluate the cost $f(\mathbf{X}_0)$ (Eq. (12)) and the Euclidean gradient $f'(\mathbf{X}_0)$ of the contrast function (Eq. (15)). Using Eq. (16), the tangent space gradient on $\mathcal{OB}$ at $\mathbf{X}_0$, $\mathrm{grad}f(\mathbf{X}_0)$ is computed, and the corresponding Hessian $\mathbf{H}_0$ is assumed to be the identity matrix of order $d^2 \times d^2$. The search direction at $\mathbf{X}_0$ is given by

$$\begin{aligned} \boldsymbol{\Xi}_0 &= \mathrm{unvec}(-\mathbf{B}_0 \mathrm{vec}(\mathrm{grad}f(\mathbf{X}_0))) \\ &= \mathrm{unvec}(-\mathbf{H}_0^{-1} \mathrm{vec}(\mathrm{grad}f(\mathbf{X}_0))), \end{aligned} \tag{29}$$

where $\mathbf{B}_0 = \mathrm{I}_{d^2}$ is the initial inverse-Hessian approximation. According to [51], the initial step-size guesses are set as

$$\alpha_{k_{\mathrm{guess}}} = \begin{cases} \min\left\{\dfrac{1}{\|\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_0))\|_\infty}, 1\right\} & \text{if } k = 0 \\ 1 & \text{if } k \geq 1. \end{cases} \tag{30}$$

Since the Hessian $\mathbf{H}_k$ acts on subspaces of the embedding space, care must

be taken to guarantee efficiency and rigor in its estimation relative to invertibility on the appropriate subspaces. One possible approach is that, instead of evaluating the inverse of $\mathbf{H}_k$, its pseudo-inverse can be computed via a rank factorization technique such as the singular value decomposition (SVD). While updating the Hessian as per the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula modified to the manifold setting

$$\mathbf{H}_{k+1} = \widetilde{\mathbf{H}}_k + \frac{\mathbf{y}_k \mathbf{y}_k^{\mathrm{T}}}{\mathbf{y}_k^{\mathrm{T}} \mathbf{s}_k} - \frac{\widetilde{\mathbf{H}}_k \mathbf{s}_k \mathbf{s}_k^{\mathrm{T}} \widetilde{\mathbf{H}}_k}{\mathbf{s}_k^{\mathrm{T}} \widetilde{\mathbf{H}}_k \mathbf{s}_k} \tag{31}$$

care should be taken to ensure whether the tangent vectors belong to the same tangent space prior to mathematically operating them. Here,

$$\mathbf{s}_k = \mathrm{vec}(\mathcal{T}_{\alpha_k \boldsymbol{\Xi}_k}(\alpha_k \boldsymbol{\Xi}_k)) \tag{32}$$

is the input change which is transported from $T_{\mathbf{X}_k}\mathcal{OB}$ to $T_{\mathbf{X}_{k+1}}\mathcal{OB}$ and $\mathbf{y}_k$ is given in Eq. (28). Since $\mathbf{H}_k$ is defined at a point $\mathbf{X}_k \in \mathcal{OB}$, it only acts on the vectors in the tangent space $T_{\mathbf{X}_k}\mathcal{OB}$. In order to have well-defined operations between the terms in Eq. (31), all the quantities should necessarily be defined at the same point $\mathbf{X}_k$. Besides, the resulting sum ought to define an operator at the next iterate $\mathbf{X}_{k+1}$. Incidentally, the vectors $\mathbf{s}_k$ and $\mathbf{y}_k$ are derived by transporting $\alpha_k \boldsymbol{\Xi}_k$ and $\mathrm{grad} f(\mathbf{X}_k)$ to $T_{\mathbf{X}_{k+1}}\mathcal{OB}$, and the ensuing result is that the linear operator $\mathbf{H}_k$ can no longer be applied on these vectors owing to their confinement to $T_{\mathbf{X}_{k+1}}\mathcal{OB}$. Therefore, it becomes imperative to replace $\mathbf{H}_k$ in the original BFGS update expression [44, p. 198] with the transported Hessian $\widetilde{\mathbf{H}}_k$ which is defined at the iterate $\mathbf{X}_{k+1}$ as

$$\widetilde{\mathbf{H}}_k = \mathcal{T} \mathbf{H}_k \mathcal{T}^{-1}, \tag{33}$$

where

$$\mathcal{T} = \begin{bmatrix} \mathcal{T}_{\alpha_k \boldsymbol{\xi}_k^{(1)}} & 0 & \dots & 0 \\ 0 & \mathcal{T}_{\alpha_k \boldsymbol{\xi}_k^{(2)}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathcal{T}_{\alpha_k \boldsymbol{\xi}_k^{(d)}} \end{bmatrix} \tag{34}$$

$$\mathcal{T}^{-1} = \begin{bmatrix} \mathcal{T}^{-1}_{\alpha_k \boldsymbol{\xi}_k^{(1)}} & 0 & \dots & 0 \\ 0 & \mathcal{T}^{-1}_{\alpha_k \boldsymbol{\xi}_k^{(2)}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathcal{T}^{-1}_{\alpha_k \boldsymbol{\xi}_k^{(d)}} \end{bmatrix}. \tag{35}$$

The tangent vectors from $T_{\mathbf{X}_{k+1}}\mathcal{OB}$ are hence transported with the block diagonal matrix constructed with an inverse vector transport operator defined as

$$\mathcal{T}^{-1}_{\alpha_k\boldsymbol{\xi}_k^{(i)}} = \mathbf{I} - \frac{(\mathbf{x}_k^{(i)} + \alpha_k\boldsymbol{\xi}_k^{(i)})(\mathbf{x}_k^{(i)})^{\mathrm{T}}}{(\mathbf{x}_k^{(i)})^{\mathrm{T}}(\mathbf{x}_k^{(i)} + \alpha_k\boldsymbol{\xi}_k^{(i)})} \tag{36}$$

to $T_{\mathbf{X}_k}\mathcal{OB}$ on which $\mathbf{H}_k$ is defined. Consequently, $\mathbf{H}_k$ transforms the transported vectors on $T_{\mathbf{X}_k}\mathcal{OB}$ and then forwards the result with the block diagonal matrix, whose diagonal elements are the vector transport operators, to $T_{\mathbf{X}_{k+1}}\mathcal{OB}$. Another desirable option to handle the issue concerning the Hessian update is to straightaway estimate the inverse-Hessian using the Sherman-Morrison-Woodbury formula [57]

$$\begin{aligned}\mathbf{B}_{k+1} =& \widetilde{\mathbf{B}}_k + \left(1 + \frac{\mathbf{y}_k^{\mathrm{T}}\widetilde{\mathbf{B}}_k\mathbf{y}_k}{\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k}\right)\frac{\mathbf{s}_k\mathbf{s}_k^{\mathrm{T}}}{\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k} \\ &- \frac{1}{\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k}(\mathbf{s}_k\mathbf{y}_k^{\mathrm{T}}\widetilde{\mathbf{B}}_k + \widetilde{\mathbf{B}}_k\mathbf{y}_k\mathbf{s}_k^{\mathrm{T}})\end{aligned} \tag{37}$$

that allows the direct calculation of the search direction without the need for matrix inversion; $\widetilde{\mathbf{B}}_k$ is the transported inverse-Hessian at the iterate $\mathbf{X}_{k+1}$, that can be denoted by

$$\widetilde{\mathbf{B}}_k = \mathcal{T}\mathbf{B}_k\mathcal{T}^{-1}. \tag{38}$$

Thus the Hessian and inverse-Hessian operators are to be transported back and forth between the two different iterates $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$ in the manifold optimization [58]. The subsequent search direction by taking into account either the inverse-Hessian or Hessian update can be reformulated as

$$\begin{aligned}\boldsymbol{\Xi}_{k+1} &= \mathrm{unvec}(-\mathbf{B}_{k+1}\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1}))) \\ &= \mathrm{unvec}(-\mathbf{H}_{k+1}^{-1}\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1}))).\end{aligned} \tag{39}$$

The performance of the QN method can degrade if the line-search is not based on the Wolfe conditions [44, p. 201]; therefore, we select α_k that satisfies the strong Wolfe conditions mentioned in Eqs. (22) and (27) by employing the inexact line-search algorithm presented in [51]. In practice, to meet the curvature condition $\mathbf{y}_k^{\mathrm{T}}\mathbf{s}_k > 0$ at a chosen step, the Hessian/inverse-Hessian update is skipped if the condition

$$\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k \geq \theta\mathbf{s}_k^{\mathrm{T}}\mathbf{H}_{k+1}\mathbf{s}_k \tag{40}$$

Table 4: Quasi-Newton algorithm on oblique manifold.

1. Randomly select $\mathbf{X}_0$ with the normality constraint, and evaluate $f(\mathbf{X}_0)$ (Eq. (12)) and $f'(\mathbf{X}_0)$ (Eq. (15)); compute $\mathrm{grad}f(\mathbf{X}_0)$ in Eq. (16) to set $\boldsymbol{\Xi}_0$ and α_0 given by Eqs. (29) and (30), respectively.
2. With the initial guess, $\alpha_k = 1$, vector transport $\boldsymbol{\Xi}_k$ to the tangent space $T_{\mathbf{X}_{k+1}}\mathcal{OB}$ to produce $\widetilde{\boldsymbol{\Xi}}_k$ using Eq. (24).
3. Determine α_k such that the strong Wolfe conditions in Eqs. (22) and (27) are satisfied by performing the inexact line-search.
4. Obtain $\mathbf{X}_{k+1}$ with Eq. (18) and evaluate $f(\mathbf{X}_{k+1})$; compute $\mathrm{grad}f(\mathbf{X}_{k+1})$ from $f'(\mathbf{X}_{k+1})$.
5. By transporting $\alpha_k\boldsymbol{\Xi}_k$ and $\mathrm{grad}f(\mathbf{X}_k)$ to $T_{\mathbf{X}_{k+1}}\mathcal{OB}$, calculate $\mathbf{s}_k$ and $\mathbf{y}_k$ as in Eqs. (32) and (28), respectively.
6. Choose an inverse vector transport operator defined in Eq. (36) to formulate either an inverse-Hessian transport (Eq. (38)) or Hessian transport (Eq. (33)).
7. Update the inverse-Hessian/Hessian following Eq. (37) or (31).
8. Determine $\boldsymbol{\Xi}_{k+1}$ for the subsequent iteration as mentioned in Eq. (39).
9. If the convergence criterion in Eq. (19) is not satisfied, return to Step 2.

is not adhered to, where $\theta > 0$ (typical θ value is 10^{-2} [44, p. 538]). For the reasons outlined in Section 2, since the retraction in Eq. (5) and vector transport in Eq. (8) are a particular choice, a Riemannian QN algorithm can also be realized by replacing them with the exponential map in Eq. (6) and Levi-Civita parallel transport in Eq. (7), respectively.

In [59], Manton argued that it may be an overkill to perform parallel transportation along geodesics in an optimization problem formulated on either the Stiefel or Grassmann manifold; as a consequence, it may suffice to utilize a "simple transport". This idea is reinforced in the improved BFGS algorithm proposed by Brace and Manton [60], which conveniently precludes the Hessian transport mechanism. In addition, Depczynski and Stöckler implemented a Riemannian BFGS algorithm on a sphere by merely adopting the inverse-Hessian update formula for the Euclidean case and omitting the inverse-Hessian transport [38]. It is relevant to remark that one can however use an efficient but rigorously defined vector transport to get efficiency, in some cases, compared to parallel transport [35]. Anyway, in agreement with Manton's notion, we have formulated an equivalent procedure on $\mathcal{OB}$ to update the Hessian/inverse-Hessian using either Eq. (31) or (37) by replacing $\widetilde{\mathbf{H}}_k$ and $\widetilde{\mathbf{B}}_k$ with $\mathbf{H}_k$ and $\mathbf{B}_k$, respectively; the objective here is to relieve the computational burden, due to the Hessian/inverse-Hessian transport calculation, which is not at the expense of efficiency. One may refer to Table 5 for the steps involved in the determination of $\boldsymbol{\Xi}_{k+1}$ in the proposed cost effective QN scheme well-suited for $\mathcal{OB}$. The halting criterion for the QN scheme is also based on $\|\mathrm{vec}(\mathrm{grad}f(\mathbf{X}_{k+1}))\|_\infty$, so that it would be congruent

Table 5: Deciding $\boldsymbol{\Xi}_{k+1}$ in cost effective quasi-Newton algorithm on oblique manifold.

1. Determine $\mathbf{s}_k$ and $\mathbf{y}_k$ expressed in Eqs. (32) and (28), respectively, using the vector transported $\alpha_k \boldsymbol{\Xi}_k$ and $\text{grad} f(\mathbf{X}_k)$.
2. Update the inverse-Hessian/Hessian as per Eq. (37) or (31), where $\widetilde{\mathbf{B}}_k$ and $\widetilde{\mathbf{H}}_k$ are replaced by $\mathbf{B}_k$ and $\mathbf{H}_k$, respectively.
3. Find $\boldsymbol{\Xi}_{k+1}$ using Eq. (39).

with the rest of the algorithms discussed in this paper.

A systematic analysis of the Riemannian BFGS method can be found in [37]. The study includes convergence theory generalizing the standard Euclidean results of QN methods, discussion of choices of transport and retraction, and tradeoffs relative to efficient implementation.

5. Evaluation

5.1. Algorithms evaluated

In this section, we evaluate the efficiency and effectiveness of algorithms designed using various combinations of the ideas discussed above: Riemannian vs. Euclidean framework and normality vs. orthonormality constraints.

We evaluate a set of Riemannian manifold algorithms:

- **Riemannian QN (specifically Riemannian BFGS) on $\mathcal{OB}$.** These enforce normality in a Riemannian framework. Three versions are tested that vary the choice of the retraction/transport mechanism.
  - OM-QN-PT is Riemannian BFGS using the exponential map as the retraction to maintain normality and parallel transport to move tangent vectors and linear operators on tangent spaces between tangent spaces.
  - OM-QN-VT is Riemannian BFGS using a simple scaling to unit length as the retraction to maintain normality and a nonisometric vector transport to move tangent vectors and linear operators on tangent spaces between tangent spaces.
  - OM-QN-CE is Brace and Manton's Riemannian BFGS algorithm that ignores transport of the linear operators between tangent spaces.
- **Riemannian QN on the Stiefel manifold**. SM-QN-LG enforces orthonormality in a Riemannian framework by working on the Stiefel

manifold and making use of a Lie Group perspective on the computations [10].

- **Alternate methods on $\mathcal{OB}$.** Three line search-based methods on $\mathcal{OB}$ are employed to assess using the Riemannian framework, normality and an alternate to the QN class of optimization methods.
  - OM-SD is a generalization of steepest descent.
  - Two generalizations of conjugate gradients that differ based on the strategy of choosing their update parameter, OM-CG-HZ and OM-CG-hybrid.

All of these Riemannian algorithms, of course, use the Riemannian notions of gradients, linear operators on tangent spaces and the Riemannian metric defined as a function of the tangent spaces to define distance on the manifold.

We also evaluate two Euclidean algorithms that use the Euclidean form of the gradients, linear operators and distance. They are both a QN method (BFGS) and differ in the enforcement of normality and orthonormality. Specifically, we have

- E-QN-GS that enforces orthogonality on each iteration using GS procedure.
- E-QN-NI that enforces normality on each iteration by scaling the individual vectors to unit length.

5.2. Performance improvement attributed to normality constraint and manifold learning

To validate the performance improvement of the proposed optimization scheme, OM-QN-VT, which accentuates the relaxation of the orthonormality constraint and manifold learning, we have performed a simulation with a set of natural images using the E-QN-GS, SM-QN-LG and E-QN-NI. The aforestated three strategies, which have already been reported in the ICA literature, optimize the same contrast function in Eq. (12) by making use of the analytical gradient in Eq. (15) to be consistent in the quantitative comparison of the source separation results with the OM-QN-VT.

A set of 12 natural images was taken from the MATLAB Image Processing Toolbox, and resized to have 50×50 pixels each. To begin with, the image data were converted into a multispectral data by concatenating the pixel values along the columns in each image, and stacking d such image data to form an input data of size $d \times 2500$. Observe that for 9-D and 11-D

simulations, we have exhaustively supplied the image combinations arising from a pool of 11 and 12 images, i.e., C_9^{11} and C_{11}^{12} different data inputs, respectively. Each multispectral data was first mixed by a random non-orthogonal mixing matrix as in [61], and then whitened for the following reasons, although our approach does not strictly require this preprocessing step.

1. Primarily with relevance to $\mathcal{OB}$ optimization, preprocessing the data by whitening can be regarded as a good initialization for the search, which will therefore speed up the convergence. In other words, mere whitening brings the data globally close to independence, and we can proceed further by locally finding a non-orthogonal unmixing matrix which reduces the MI further [32]. On top of that, it has been argued in [62] that whitening reduces the complexity of the problem in finding an optimal unmixing matrix, despite it being very simple and a standard procedure.
2. Interestingly, the whitening process helps to simplify the design of kernel density estimator due to the fact that such a constraint enables the reconstructed signals to be treated as zero-mean and unit variance random variables [2]; resultantly, the bandwidth h can be safely chosen as in Eq. (14).

The parameter values for the algorithms—the E-QN-GS, SM-QN-LG and E-QN-NI—were set as reported in [2] and [10], whereas in the design of OM-QN-VT, the values conform to those followed conventionally in the equivalent Euclidean setting, as recorded below: $\epsilon = 10^{-6}$, $c_1 = 0.01$ and $c_2 = 0.9$. Various parameters in the $\mathcal{OB}$ line-search routine were assigned values as follows, which are recommended in [51]: τ_1 to determine the size of the jumps of the iterates in the bracketing phase is 9; τ_2 and τ_3 to restrict a trial point from being arbitrarily close to the extremes of the interval in the sectioning phase are 0.1 and 0.5, respectively; the line search threshold is set to be 2.2204×10^{-15}. The software system was implemented in MATLAB 7.5 on a PC (Pentium D 3-GHz CPU, 2-GB DDR2 RAM) using Windows XP Professional SP3.

The unmixing matrices from the experimented schemes were used to reconstruct the separated source images $\hat{I}^{(i)}$, $i = 1, 2, \ldots, d$; since we already have the original images $I^{(i)}$, after necessary reordering of $\hat{I}^{(i)}$, $i = 1, 2, \ldots, d$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{d} \sum_{l,c} \left(I_{l,c}^{(i)} - \hat{I}_{l,c}^{(i)} \right)^2}{\sum_{i=1}^{d} \sum_{l,c} \left(I_{l,c}^{(i)} \right)^2}}, \tag{41}$$

Table 6: Comparison of MI values from E-QN-GS, SM-QN-LG, E-QN-NI and OM-QN-VT in ICA learning.

Optimizer	$d = 9$			$d = 11$		
minimizing MI	MI mean	MI std. dev.	least MI %	MI mean	MI std. dev.	least MI %
E-QN-GS	12.212951	0.269491	0	14.911829	0.225228	0
SM-QN-LG	11.603528	0.124083	0	14.170216	0.099226	0
E-QN-NI	11.512988	0.197401	3.64	14.053028	0.173112	0
OM-QN-VT	**11.425382**	0.172612	96.36	**13.905114**	0.135201	100

Table 7: Comparison of RMSE in ICA learning using E-QN-GS, SM-QN-LG, E-QN-NI and OM-QN-VT.

Optimizer under	$d = 9$			$d = 11$		
investigation	RMSE mean	RMSE std. dev.	least RMSE %	RMSE mean	RMSE std. dev.	least RMSE %
E-QN-GS	0.219962	0.040694	0	0.250308	0.023857	0
SM-QN-LG	0.090414	0.025770	7.27	0.121046	0.038539	0
E-QN-NI	0.115623	0.039241	5.45	0.145189	0.050567	0
OM-QN-VT	**0.066644**	0.022887	87.27	**0.081939**	0.022371	100

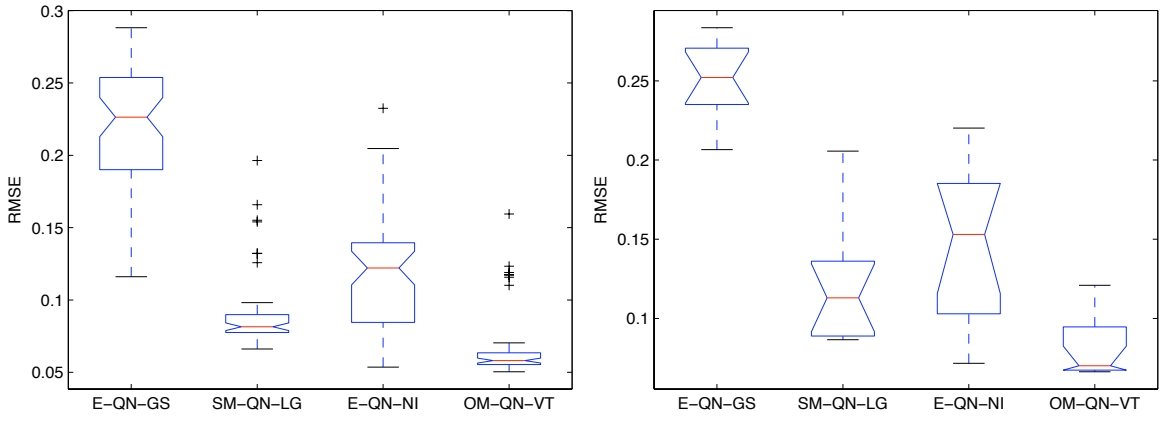

Figure 2: Boxplots depicting the RMSE values resulted in the simulations involving the E-QN-GS, SM-QN-LG, E-QN-NI and OM-QN-VT, which make use of an input image data of dimension (a) 9 and (b) 11.

where l and c are the row and column indices of the pixels, respectively, in $I^{(i)}$ and $\hat{I}^{(i)}$, $i = 1, 2, \ldots, d$. Additionally, the minimized MI values by the algorithms under study were recorded. Since the RMSE measure is always non-negative, scale invariant and does not suffer from the permutation ambiguity due to the reordering of the estimated sources, it possesses the virtues of the well-known performance measures such as the SIR and PI. Furthermore, unlike the SIR and PI, which compare the original and estimated unmixing matrices, the RMSE measure we intend to work with takes the original and reconstructed sources into account. Therefore, the RMSE lends itself to consider any random non-orthogonal mixing matrix in the simulation, whereas the retrieved unmixing matrices are either subject to orthonormality or normality constraint.

To assess the performance of the OM-QN-VT with regard to the rest of the optimization methods, the outcome of the experiment, namely, the mean, standard deviation, and percentage of cases with the minimum value of MI and RMSE are provided in Table 6 and 7, respectively. We infer from the empirical studies that it is beneficial to stay on $\mathcal{OB}$ and enjoy the superior convergence characteristics rather than enforcing the normality constraint by mere post-normalization. This result has been illustrated using the E-QN-NI as a counterexample, by incorporating an Euclidean QN—identical to the NPICA—which reimposes the constraint by post-normalization after each iteration. Indeed, it is of interest to investigate whether the increased degrees of freedom offered by the normality constraint, compared to the conventional orthonormality constraint, substantially contributes to the solution quality in the ICA implementation. Keeping this in mind, we have replaced the routine for post-normalization with the popular GS orthogonalization procedure in the E-QN-NI to maintain the orthonormality constraint in the E-QN-GS. Since it has been reported in a recent work [10] that learning the ICA unmixing matrix on the Stiefel manifold, by accounting for the manifold's curved geometry in a more natural way, using a Lie group method improves the IC estimation accuracy, a QN algorithm in conjunction with Lie group techniques, SM-QN-LG, is also included.

It is straightforward to conclude that the OM-QN-VT can better minimize the MI, since in 96.36% and 100% of the total test cases its minimization performance surpassed the other approaches for the data of dimension 9 and 11, respectively. Even though the differences in the MI values between the investigated algorithms may be seemingly inappreciable, the importance of manifold learning and relaxing the orthonormality constraint is corroborated by the noticeable reduction in the RMSE values resulted from the

OM-QN-VT. Observe from the boxplots[4] in Fig. 2 that the true medians of the RMSE values corresponding to the OM-QN-VT for both the 9-D and 11-D data sets do differ from the medians of the remaining methods with 95% confidence, since the notches in the boxes do not overlap. Conspicuously, the proposed algorithm outperformed the rest of the methods in terms of the RMSE values in 87.27% and 100% of the total trials. We attribute the superior performance of the OM-QN-VT, particularly in comparison with the E-QN-NI wherein the minimization is subject to post-normalization, to its ability to preserve the asymptotic super-linear convergence thanks to manifold learning.

5.3. Results in simulations with natural images and audio signals

To subjectively assess the performance of the ICA algorithms on $\mathcal{OB}$ in relation to the methods—the JADE [63], infomax [64], fastICA [65], GEKD-ICA [1] and NPICA [2]—well-known among the ICA community, a simulation was performed using nine natural images, each having 200×200 pixels. The multispectral data of size 9×40000, constructed using the image pixel values, were mixed by a random non-orthogonal matrix, followed by whitening, and then supplied as the input for all the investigated approaches. The source estimates were reconstructed with the unmixing matrices estimated by the various schemes under study and are furnished in Fig. 3 for visual scrutiny along with the original source and mixed images. Notably, the methods built on the source adaptive contrast functions excelled the ones relying on a "surrogate" function to measure independence. The separation performance was quantitatively evaluated using Eq. (41) and the RMSE value for the OM-QN-VT—0.030149—is remarkably lower than the JADE, infomax and fastICA values, which are 0.191902, 0.196403 and 0.213413, respectively. Although the SM-QN-LG and E-QN-NI were implemented with the contrast function in Eq. (12), due to the orthonormality constraint introduced by the Stiefel manifold learning and the enforcement of the normality constraint, their performances are limited as evident from the RMSE values of 0.065568 and 0.128786, respectively. Of particular interest is the GEKD-ICA RMSE value—0.041362—which is higher than the OM-QN-VT, even if

[4]In this paper, the boxplots are defined as follows: the thin horizontal line passing through the notches in the box stands for the median of the data, the upper and lower ends of the box correspond to the upper and lower quartiles, respectively, and the whiskers extending from the box refer to points within a standard range of the data, defined as 1.5 times the interquartile range; beyond the ends of the whiskers, outliers are displayed with a plus sign.

Figure 3: Simulation results using nine natural images of size 200×200 for subjective assessment. (a) Original source images. (b) Mixed images. (c)–(j) Reconstructed sources using the ICA unmixing matrices estimated by the JADE, infomax, fastICA, E-QN-NI, SM-QN-LG, GEKD-ICA, NPICA and OM-QN-VT with RMSE values of 0.191902, 0.196403, 0.213413, 0.128786, 0.065568, 0.041362, 0.030149 and 0.030149, respectively.

it claims to outmatch a class of popular source adaptive ICA algorithms.

Unlike the multimodal density distributions of the pixel values in natural images, the amplitudes of the audio signal samples follow unimodal density distributions which can trivially be estimated using parametric techniques. Moreover, some of the existing ICA algorithms, which will make use of a prior assumption about the density distributions of the underlying sources, may be befitting for the audio signal separation task. However, to evaluate the robustness of the proposed method over a wider range of applications, an experiment was set up with nine audio signals in the WAVE format comprising speech, song, music and mechanical sound, which are sampled at 8 KHz with a sample size of 8 bits. Since the duration of each signal is roughly 6 seconds, they all have a length of 50000 time points. The same set of ICA algorithms investigated with the simulation involving natural images were supplied with the mixed and whitened multispectral data of size 9×50000; similar to the previous experiment, the mixing matrix was considered to be a random and non-orthogonal one. The retrieved source estimates from various approaches were reordered and quantitatively compared with the original source signals. For visual comparison of the source separation performance, the amplitude error corresponding to each source sample is plotted for all the algorithms in Fig. 4; to clearly distinguish between the approaches whose performance distinctly differ from the rest—more than an RMSE difference of 0.001—the plots are shown in different colors. Aside from the case of audio source 1, the OM-QN-VT and NPICA could retrieve the sources with the least RMSE measure. In comparison with the "surrogate"-function-based techniques, the SM-QN-LG and GEKD-ICA managed to reproduce the sources with an increased accuracy. With respect to the separation results, the tested algorithms are graded in the following order: OM-QN-VT/NPICA, GEKD-ICA, SM-QN-LG, fastICA, infomax, JADE and E-QN-NI with the RMSE values of 0.014899, 0.015991, 0.017998, 0.022889, 0.023890, 0.044443 and 0.058083, respectively. The worst RMSE value yielded by the E-QN-NI emphasizes the fact that the post-normalization performed on the ICA unmixing matrix, in the pretext of ascertaining its columns to be of unit-norm by naive methods, will significantly deteriorate the source separation quality, in defiance of a robust contrast function grounded on non-parametric density estimation tools.

It is candidly stated that the NPICA performs on par with the $\mathcal{OB}$ algorithms, because unlike approaches where the constraint is imposed between the iterations, Boscolo *et al.* carefully adopt the normality constraint in the gradient calculation, which turns out to be an unconstrained optimization task. Since the underlying supposition in [2] regarding the constraint

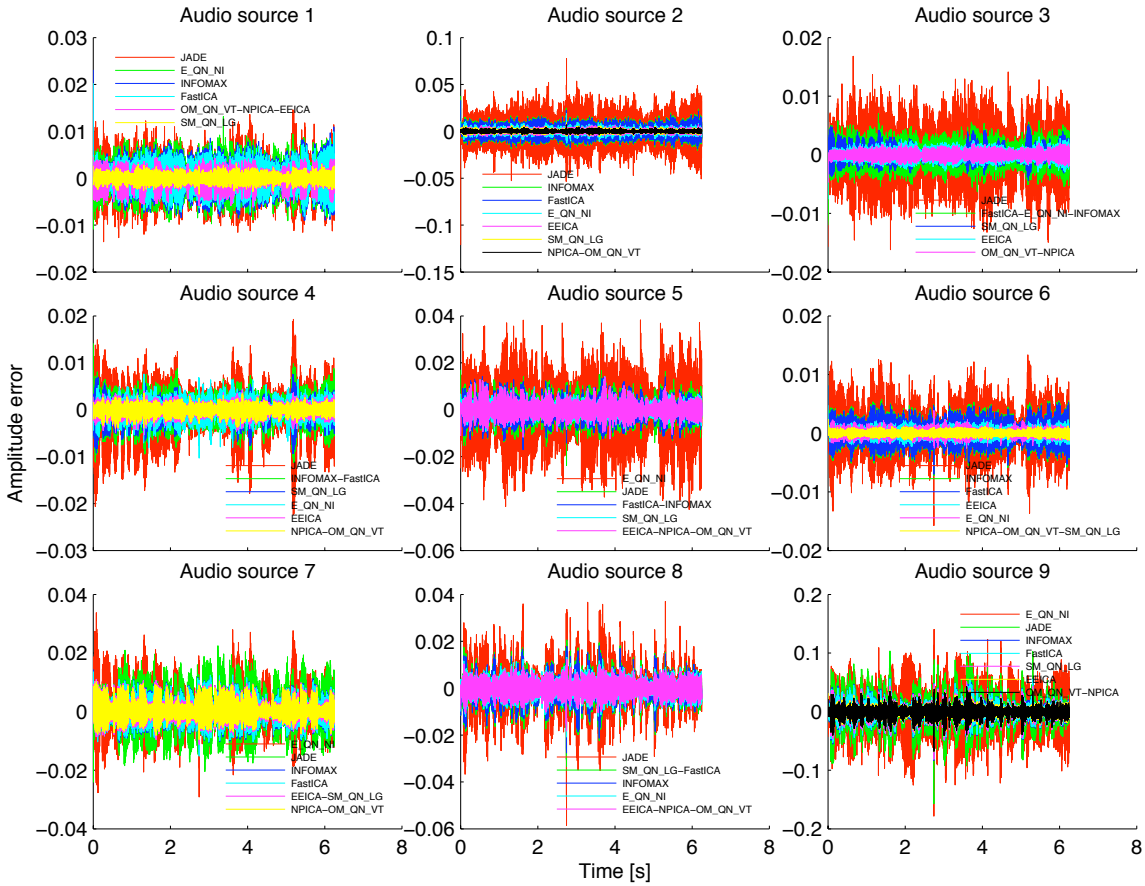

Figure 4: The amplitude error between each estimated and original source sample while retrieving the nine audio signals of approximately 6 seconds duration, sampled at 8 KHz and mixed with a non-orthogonal matrix, using various ICA algorithms. To appreciate the differences in performance subjectively, the methods whose RMSE values differ below a threshold of 0.001 are clubbed together. For grading the performance quantitatively, the RMSE values of OM-QN-VT/NPICA, GEKD-ICA, SM-QN-LG, fastICA, infomax, JADE and E-QN-NI are listed as 0.014899, 0.015991, 0.017998, 0.022889, 0.023890, 0.044443 and 0.058083, respectively, in the increasing order.

Table 8: Convergence of OM-SD, OM-CG-HZ, OM-CG-hybrid, OM-QN-VT and OM-QN-CE algorithms, experimented with 3, 6 and 9 natural images of size 50×50, by supplying 10 random initial input matrices. Standard deviations of MI are less than 10^{-7}, implying consistency in finding a local minimum. As a benchmark for the empirical study, the MI values from OM-QN-PT are tabulated in the last column.

d	Mean MI Value					
	OM-SD	OM-CG-HZ	OM-CG-hybrid	OM-QN-VT	OM-QN-CE	OM-QN-PT
3	3.517011	3.517011	3.517011	3.517011	3.517011	3.517011
6	7.586613	7.586613	7.586613	7.586613	7.586613	7.586613
9	11.553152	11.553152	11.553152	11.553152	11.553152	11.553152

imposition is practically equivalent to ours, where the major emphasis is to allow the iterations to proceed unperturbed by intrinsic preservation of the normality constraint, it is not surprising to infer from the results that they converge to the same solution. Nevertheless, this paper specifically concerns itself with a manifold optimization strategy—a more general optimization framework founded on the notions of differential geometry—which can handle any ICA contrast function intended to work with the normality constraint. Importantly, the recently proposed source adaptive contrast functions such as the non-parametric likelihood ratio (NLR) [66] and the MI based on Jensen's inequality [67], which are the preferred choices over the NPICA for speech separation, can readily be applied in our manifold techniques without any modification; whereas, it is not always guaranteed that the Euclidean gradients of the contrast functions can be tailored to account for the normality constraint for the optimization problem to be treated as an unconstrained one, as in the case of NPICA.

5.4. Convergence and scalability issues of oblique manifold algorithms

To better understand the optimization landscape and to verify the convergence behavior of $\mathcal{OB}$ algorithms—the OM-SD, OM-CG-HZ, OM-CG-hybrid, OM-QN-VT and OM-QN-CE—in practical problems, the mixed and prewhitened multispectral data of size $d \times 2500$, where $d = 3, 6, 9$, were allowed to be unmixed by the aforementioned methods. Specified below are the parameter settings for the OM-SD, OM-CG-HZ and OM-CG-hybrid: $\gamma = 0.5$, $c_1 = 0.01$, $c_2 = 0.1$, $\varphi = 0.01$ and $\epsilon = 10^{-6}$. Due to the uniqueness of the unmixing matrix apart from the scaling and permutation indeterminacies, convergence to a local minimum was feared to end up in a suboptimal solution. Optimistically, as discussed in [68], it may be quite sufficient in practice if the convergence to a local minimum can be attained for an arbi-

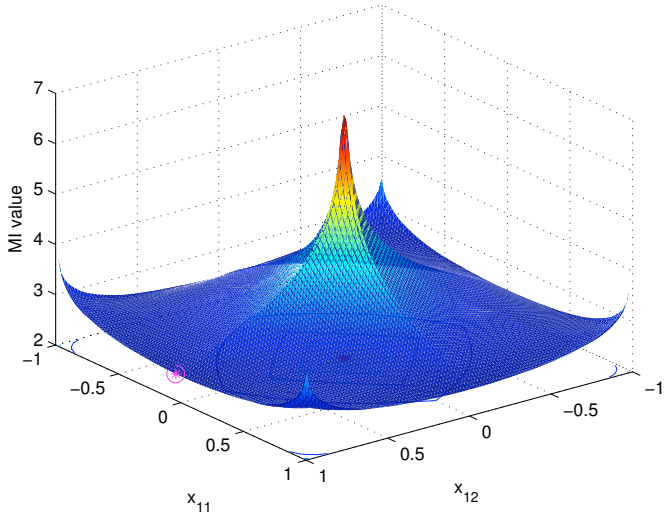

Figure 5: Optimization landscape for the contrast function in Eq. (12), evaluated using a 50×50 sized, 2-D natural image data; the global minimum is marked by a pink circled asterisk. x_{11} and x_{12} are the independent unmixing matrix elements, while $\mathbf{X}$ is subject to the normality constraint.

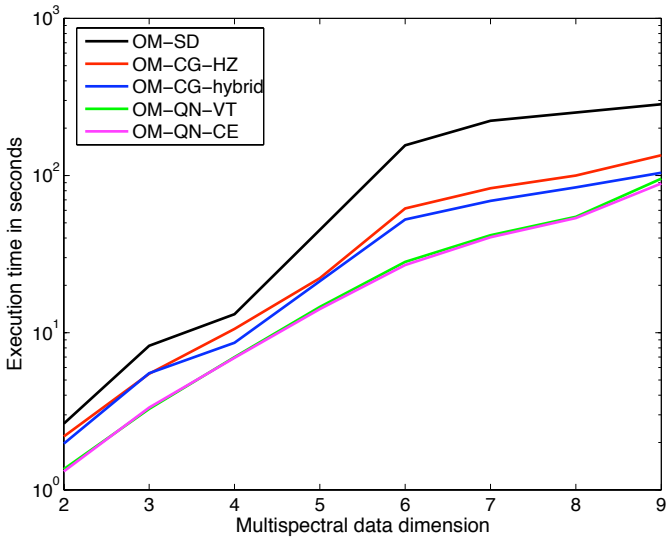

Figure 6: Relationship between the average CPU time (on a PC, Pentium D 3-GHz CPU, 2-GB DDR2 RAM) taken by the collection of $\mathcal{OB}$ algorithms, implemented with the same stopping criterion and data dimension. Notice that the y-axis is logarithmic.

trary initial guess. We surmise that the above requirement is satisfied by the methods designed for $\mathcal{OB}$, because for each data dimension, the MI corresponding to the optimal solution remains unaltered regardless of 10 different random initial inputs as enlisted in Table 8. By way of explanation, the standard deviations of the MI lie well below 10^{-7} in all the experiments, which is due to the convergence thresholds set for the optimization algorithms; it implies that the proposed methods consistently locate the local minimum. In addition, these MI values coincide with the minimization results, included in the last column of Table 8, from the OM-QN-PT algorithm built with the exponential map in Eq. (6) and Levi-Civita parallel transport in Eq. (7). This highlights the fact that replacing the conventional Riemannian optimization ingredients in Eqs. (6) and (7) with a retraction in Eq. (5) and vector transport in Eq. (8), respectively, is practically equivalent. Although the retraction update appears to be simple in our case, $\mathcal{OB}$ optimization does not amount to mere post-normalization. We recall that for ensuring convergence, suitable mechanisms to project the gradient onto a tangent space, transport of various elements between different tangent spaces, and transport of an approximate Hessian back and forth between the tangent spaces at subsequent iterates are to be incorporated in the algorithm design. To discern the structure of the optimization landscape generated by our contrast function, we have portrayed the simplest case concerning a 2-D natural image data of size 50×50 in Fig. 5, where one of the four possible global minima is indicated using a pink circled asterisk.

To examine the scalability issue, 10 trial executions were performed using the proposed $\mathcal{OB}$ techniques with random initial starts for each data dimension varying from two to nine, and the average CPU time consumption in seconds for each technique was recorded. In order to be fair in comparison, all the approaches were supplied with an identical set of initial solution guesses and were subject to the same stopping criterion. It is plausible to conjecture that the execution time for the proposed collection of optimizers increases quadratically with the data dimension; in the scalability graph depicted in Fig. 6, a quadratic fit is more appropriate for the data-points representing the execution time taken by each approach, with 95% confidence bounds. In essence, this investigation ascertains that the proposed schemes are computationally scalable for the multispectral offline data analysis.

6. Conclusion

This paper describes a collection of geometric optimization algorithms—the OM-SD, OM-CG-HZ, OM-CG-hybrid, OM-QN-VT, OM-QN-PT and OM-QN-CE—meant to convert the ICA optimization task subject to the normality constraint into an unconstrained one by staying on $\mathcal{OB}$. The present work has been motivated by the rationale that follows. Even though a considerable effort has recently been focused by the ICA community on the development of Riemannian methods, optimizing effectively on the Stiefel, Grassmann and flag manifolds to account for the orthonormality constraint, manifold algorithms with an intent to handle the normality constraint have not been widely investigated. Ironically, as conceded in the literature, the ICA methods grounded on the normality constraint yield a more accurate solution in comparison with the ones insisting on the orthonormality constraint due to the following reasons: (1) the former methods rely on the contrast functions which go for a direct estimation of source densities and (2) they enjoy more degrees of freedom during the course of optimization.

The OM-SD algorithm, though simple in construction, in conjunction with the Armijo's step-size rule assures convergence to a local minimum. However, the linear convergence inherent to the OM-SD approach outweighs its potential use to unmix the higher dimensional data. To resolve this, we ventured in the design of OM-CG algorithms with the choice of HZ and a hybrid update parameter; owing to the strong convergence properties and the ability to overcome "jamming", these update parameters did supersede the traditional choices. Inspite of the super-linear convergence rate offered by the OM-CG, since the QN is the favored choice in many real world applications, we have incorporated all the $\mathcal{OB}$ optimization ingredients—gradient projection onto a tangent space, retraction, vector transport, inverse vector transport and Hessian transport—to develop a OM-QN-VT algorithm and its slight variation, OM-QN-CE, which evades the Hessian transport. We have systematically compared these Riemannian algorithms and two Euclidean QN algorithms, E-QN-GS and E-QN-NI, to evaluate the influence on efficiency and effectiveness of the choices of Riemannian and Euclidean frameworks for the problem and the enforcing of normality and orthonormality.

It turns out that the accurate estimation of the unmixing matrix is of prime concern in certain dedicated applications, for instance, offline medical image segmentation [69]. To be pertinent to such an application, we have demonstrated the techniques on $\mathcal{OB}$ by opting for a source adaptive contrast function, where the non-parametric density estimation is considerably

expedited thanks to the IFGT. In order to give insight into the influence of manifold learning on the solution accuracy with regard to both normality and orthonormality constraints, an experimental validation using natural images and audio signals is provided. For a visual assessment, the ICA separation results from prominent ICA techniques as well as the OM-QN-VT are furnished; apparently, as predicted by the RMSE values, learning on $\mathcal{OB}$ is conclusively better than either orthogonalizing the IC estimates between the iterations or implicitly handling the orthonormality constraint through manifold learning or enforcing the normality constraint. To verify whether ICA algorithms with the normality constraint reach the lower Cramér-Rao bound is deferred to future work. However, what is stressed in the present paper is that the MI-based estimation of ICs by relaxing the unnecessary orthogonality constraint is more accurate than the ones obtained with the same constraint, regardless of whether the optimal lower Cramér-Rao bound is reached or not. A possible future direction is to examine the suitability of optimizing contrast functions such as the NLR criterion or the MI based on Jensen's inequality by staying on $\mathcal{OB}$ for quantifying the brain tissues in magnetic resonance image (MRI) data. Further research is underway to speed up the function and gradient evaluation step by alternatively estimating the densities by way of a Gaussian mixture model.

Appendix A. Proof for the choice of oblique projector as inverse vector transport operator

We are given $\mathbf{x}_k \in \mathcal{S}^{n-1}$ and we choose a direction $\boldsymbol{\xi}_k \in T_{\mathbf{x}_k}\mathcal{S}$ to define the tangent space corresponding to the next iterate $T_{\mathbf{x}_{k+1}}\mathcal{S}$, where

$$\mathbf{x}_{k+1} = R_{\mathbf{x}_k}(\boldsymbol{\xi}_k) = \frac{\mathbf{x}_k + \boldsymbol{\xi}_k}{\|\mathbf{x}_k + \boldsymbol{\xi}_k\|}.$$

Theorem 1. *Given a vector transport on* $\mathcal{S}^{n-1}$, $\mathcal{T}_{\boldsymbol{\xi}_k} = \mathbf{I} - \dfrac{(\mathbf{x}_k + \boldsymbol{\xi}_k)(\mathbf{x}_k + \boldsymbol{\xi}_k)^{\mathrm{T}}}{\|\mathbf{x}_k + \boldsymbol{\xi}_k\|^2}$, *an inverse vector transport is* $\mathcal{T}_{\boldsymbol{\xi}_k}^{-1} = \mathbf{I} - \dfrac{(\mathbf{x}_k + \boldsymbol{\xi}_k)\mathbf{x}_k^{\mathrm{T}}}{\mathbf{x}_k^{\mathrm{T}}(\mathbf{x}_k + \boldsymbol{\xi}_k)}$.

Proof: A. Linear algebra interpretation
Note that both operators are linear maps on $\mathbb{R}^n$, and $\mathcal{T}_{\boldsymbol{\xi}_k}^{-1}\mathcal{T}_{\boldsymbol{\xi}_k} = \mathbf{I}$ when restricted to $T_{\mathbf{x}_k}\mathcal{S}$. From the definitions of tangent space and retraction, we can write

$$\begin{aligned} \mathbf{x}_k^{\mathrm{T}}\boldsymbol{\xi}_k &= 0; \\ \mathbf{v}_k &= \mathbf{x}_k + \boldsymbol{\xi}_k = \mathbf{x}_{k+1}\|\mathbf{v}_k\|. \end{aligned}$$

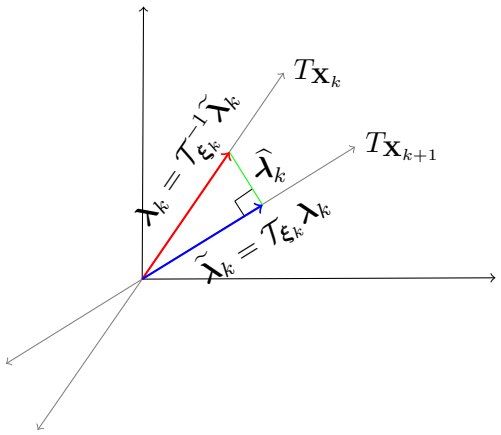

Figure A.7: Geometric approach to explicate that if the chosen vector transport on $\mathcal{OB}$ is an orthogonal projector, the corresponding inverse vector transport should be an oblique one.

Substituting the expressions for $\mathcal{T}_{\boldsymbol{\xi}_k}$ and $\mathcal{T}^{-1}_{\boldsymbol{\xi}_k}$ in $\mathcal{T}^{-1}_{\boldsymbol{\xi}_k}\mathcal{T}_{\boldsymbol{\xi}_k}\boldsymbol{\xi}_k$ yields $\boldsymbol{\xi}_k$ as shown below:

$$\begin{aligned}
\mathcal{T}^{-1}_{\boldsymbol{\xi}_k}\mathcal{T}_{\boldsymbol{\xi}_k}\boldsymbol{\xi}_k &= \left[\mathbf{I} - (\mathbf{x}_k^{\mathrm{T}}\mathbf{v}_k)^{-1}\mathbf{v}_k\mathbf{x}_k^{\mathrm{T}}\right]\left[\mathbf{I} - \mathbf{x}_{k+1}\mathbf{x}_{k+1}^{\mathrm{T}}\right]\boldsymbol{\xi}_k \\
&= \boldsymbol{\xi}_k - \mathbf{x}_{k+1}\mathbf{x}_{k+1}^{\mathrm{T}}\boldsymbol{\xi}_k - (\mathbf{x}_k^{\mathrm{T}}\mathbf{v}_k)^{-1}\mathbf{v}_k\mathbf{x}_k^{\mathrm{T}}\boldsymbol{\xi}_k + \\
&\quad (\mathbf{x}_k^{\mathrm{T}}\mathbf{v}_k)^{-1}\mathbf{v}_k(\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k+1})(\mathbf{x}_{k+1}^{\mathrm{T}}\boldsymbol{\xi}_k) \\
&= \boldsymbol{\xi}_k - \mathbf{x}_{k+1}(\mathbf{x}_{k+1}^{\mathrm{T}}\boldsymbol{\xi}_k) + \\
&\quad \mathbf{v}_k\frac{(\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k+1})(\mathbf{x}_{k+1}^{\mathrm{T}}\boldsymbol{\xi}_k)}{(\mathbf{x}_k^{\mathrm{T}}\mathbf{v}_k)} \\
&= \boldsymbol{\xi}_k - \mathbf{x}_{k+1}(\mathbf{x}_{k+1}^{\mathrm{T}}\boldsymbol{\xi}_k) + \\
&\quad \mathbf{x}_{k+1}\frac{\|\mathbf{v}_k\|(\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k+1})(\mathbf{x}_{k+1}^{\mathrm{T}}\boldsymbol{\xi}_k)}{(\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k+1})\|\mathbf{v}_k\|} \\
&= \boldsymbol{\xi}_k - \mathbf{x}_{k+1}(\mathbf{x}_{k+1}^{\mathrm{T}}\boldsymbol{\xi}_k) + \mathbf{x}_{k+1}(\mathbf{x}_{k+1}^{\mathrm{T}}\boldsymbol{\xi}_k) \\
&= \boldsymbol{\xi}_k.
\end{aligned}$$

Hence $\mathcal{T}^{-1}_{\boldsymbol{\xi}_k}$ is the related inverse vector transport of $\mathcal{T}_{\boldsymbol{\xi}_k}$.

B. Geometric interpretation

A general projection setup uses two spaces of dimension $1 \leq k \leq n$, $\mathbb{K}$ and

$\mathbb{L}$. A projector $\mathbf{P}$ can then be defined such that for any vector $\mathbf{z} \in \mathbb{R}^n$

$$\begin{aligned} \mathbf{P}_{\mathbf{z}} &\in \mathbb{K} \\ \mathbf{z} - \mathbf{P}_{\mathbf{z}} &\perp \mathbb{L}. \end{aligned}$$

It is possible to show that if we have bases $\mathbf{K}$ and $\mathbf{L}$ for $\mathbb{K}$ and $\mathbb{L}$, respectively, and suppose $\mathbf{L}^{\mathrm{T}}\mathbf{K}$ is nonsingular then

$$\mathbf{P} = \mathbf{K}(\mathbf{L}^{\mathrm{T}}\mathbf{K})^{-1}\mathbf{L}^{\mathrm{T}}.$$

Furthermore, if $\mathbb{K} = \mathbb{L}$ we may take $\mathbf{K} = \mathbf{L} = \mathbf{Q}$ with $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}_k$, and the symmetric matrix

$$\mathbf{P} = \mathbf{Q}\mathbf{Q}^{\mathrm{T}}$$

is an orthogonal projector. If $\mathbb{K} \neq \mathbb{L}$, the projector is called oblique. These ideas are applied to define $\mathcal{T}_{\boldsymbol{\xi}_k}$ and $\mathcal{T}_{\boldsymbol{\xi}_k}^{-1}$; Fig. A.7 illustrates that if $\mathcal{T}_{\boldsymbol{\xi}_k}$ is an orthogonal projector then $\mathcal{T}_{\boldsymbol{\xi}_k}^{-1}$ must be an oblique one. Since $\mathcal{T}_{\boldsymbol{\xi}_k}$ is an orthogonal projector, $\widetilde{\boldsymbol{\lambda}}_k = \mathcal{T}_{\boldsymbol{\xi}_k}\boldsymbol{\lambda}_k$ is the best approximation in $T_{\mathbf{x}_{k+1}}$ to $\boldsymbol{\lambda}_k$. In terms of the general projector notion, we have $\mathbb{K} = \mathbb{L} = T_{\mathbf{x}_{k+1}}$. Therefore

$$\begin{aligned} \mathbf{I} &= \begin{bmatrix}\mathbf{Q}_{\mathbf{x}_{k+1}} & \mathbf{x}_{k+1}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_{\mathbf{x}_{k+1}} & \mathbf{x}_{k+1}\end{bmatrix}^{\mathrm{T}} \\ &= \mathbf{Q}_{\mathbf{x}_{k+1}}\mathbf{Q}_{\mathbf{x}_{k+1}}^{\mathrm{T}} + \mathbf{x}_{k+1}\mathbf{x}_{k+1}^{\mathrm{T}} \\ \mathcal{T}_{\boldsymbol{\xi}_k} &= \mathbf{Q}_{\mathbf{x}_{k+1}}\mathbf{Q}_{\mathbf{x}_{k+1}}^{\mathrm{T}} = \mathbf{I} - \mathbf{x}_{k+1}\mathbf{x}_{k+1}^{\mathrm{T}}, \end{aligned}$$

where $\mathbf{Q}_{\mathbf{x}_{k+1}} \in \mathbb{R}^{n \times n-1}$ is the orthonormal basis of $T_{\mathbf{x}_{k+1}}$. From Fig. A.7, it is clear that the angular relationship of the residual $\boldsymbol{\lambda}_k - \widetilde{\boldsymbol{\lambda}}_k$ and $T_{\mathbf{x}_k}$ is not perpendicular; therefore to map $\widetilde{\boldsymbol{\lambda}}_k$ to $\boldsymbol{\lambda}_k$, we must use an oblique projector. Let $\widehat{\boldsymbol{\lambda}}_k = \widetilde{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}_k$ be the residual that results from the oblique projection defining $\mathcal{T}_{\boldsymbol{\xi}_k}^{-1}$. We seek a projector $\mathbf{P}$ such that $\widehat{\boldsymbol{\lambda}}_k = \mathbf{P}\widetilde{\boldsymbol{\lambda}}_k$; if we succeed it follows that

$$\begin{aligned} \boldsymbol{\lambda}_k &= \widetilde{\boldsymbol{\lambda}}_k - \widehat{\boldsymbol{\lambda}}_k = \widetilde{\boldsymbol{\lambda}}_k - \mathbf{P}\widetilde{\boldsymbol{\lambda}}_k \\ &= (\mathbf{I} - \mathbf{P})\widetilde{\boldsymbol{\lambda}}_k = \mathcal{T}_{\boldsymbol{\xi}_k}^{-1}\widetilde{\boldsymbol{\lambda}}_k. \end{aligned}$$

Since we have

$$\begin{aligned} \widehat{\boldsymbol{\lambda}}_k &= \widetilde{\boldsymbol{\lambda}}_k - \boldsymbol{\lambda}_k \perp T_{\mathbf{x}_{k+1}} \rightarrow \widehat{\boldsymbol{\lambda}}_k \in \mathcal{R}(\mathbf{x}_{k+1}) = \mathbb{K} \\ \boldsymbol{\lambda}_k &= \widetilde{\boldsymbol{\lambda}}_k - \mathbf{P}\widetilde{\boldsymbol{\lambda}}_k \in T_{\mathbf{x}_k} \rightarrow \boldsymbol{\lambda}_k \perp \mathcal{R}(\mathbf{x}_k) = \mathbb{L}, \end{aligned}$$

where $\mathcal{R}(\cdot)$ is the span of the vector under consideration, the projector can be written as

$$\mathbf{P} = \mathbf{x}_{k+1}(\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k+1})^{-1}\mathbf{x}_k^{\mathrm{T}}.$$

It implies that the related inverse vector transport is

$$\mathcal{T}_{\boldsymbol{\xi}_k}^{-1} = \mathbf{I} - \frac{\mathbf{x}_{k+1}\mathbf{x}_k^{\mathrm{T}}}{\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k+1}}.$$

□

Appendix B. First derivative of contrast function

The contrast function is given by

$$f(\mathbf{X}) = \sum_{i=1}^{d} H_i - \log|\det \mathbf{X}|,$$

where

$$\begin{aligned} H_i &:= -E\left\{\log p^{(i)}\left(b^{(i)}\right)\right\} \\ &\simeq -\frac{1}{N}\sum_{u=1}^{N}\log p^{(i)}\left(b_u^{(i)}\right) \end{aligned}$$

and $p^{(i)}\left(b_u^{(i)}\right)$ is expressed in Eq. (13). Here N and d are the total number of data-points and dimension of the multispectral data input for the ICA algorithm, respectively, h is the kernel bandwidth and the superscript indices signify that the density function is not the same for all components in general. In what follows, for ease of notation, we resort to the same symbols f and H to refer to the contrast function and entropy, respectively, that have been approximated by means of the estimators of kernel density and expectation. The first derivative of the contrast function can be derived as follows:

$$\frac{\partial H_i}{\partial x_{rs}} = -\frac{1}{N}\sum_{u=1}^{N}\frac{1}{p^{(i)}\left(b_u^{(i)}\right)}\frac{\partial}{\partial x_{rs}}p^{(i)}\left(b_u^{(i)}\right)$$

$$\frac{\partial}{\partial x_{rs}}p^{(i)}\left(b_u^{(i)}\right) = \frac{1}{Nh\sqrt{2\pi}}\sum_{v=1}^{N}\frac{\partial}{\partial x_{rs}}\exp\left[\frac{-\left(b_u^{(i)} - b_v^{(i)}\right)^2}{2h^2}\right]$$

$$\frac{\partial}{\partial x_{rs}} \exp\left[\frac{-\left(b_u^{(i)} - b_v^{(i)}\right)^2}{2h^2}\right] = \exp\left[\frac{-\left(b_u^{(i)} - b_v^{(i)}\right)^2}{2h^2}\right] \times$$
$$\frac{-2\left(b_u^{(i)} - b_v^{(i)}\right)}{2h^2} \frac{\partial}{\partial x_{rs}} \left(b_u^{(i)} - b_v^{(i)}\right)$$
$$= -\frac{1}{h^2} \exp\left[\frac{-\left(b_u^{(i)} - b_v^{(i)}\right)^2}{2h^2}\right] \times \left(b_u^{(i)} - b_v^{(i)}\right)\left(m_u^{(s)} - m_v^{(s)}\right)\delta_{ri}.$$

From $b_u^{(i)} = \sum_{t=1}^{d} x_{it} m_u^{(t)}$, it follows that

$$\frac{\partial b_u^{(i)}}{\partial x_{rs}} = \begin{cases} 0 & \text{if } r \neq i \\ m_u^{(s)} & \text{if } r = i. \end{cases}$$

Therefore,

$$\frac{\partial}{\partial x_{rs}} p^{(i)}\left(b_u^{(i)}\right) = \frac{1}{Nh\sqrt{2\pi}} \sum_{v=1}^{N} \left(-\frac{1}{h^2}\right) \times$$
$$\exp\left[\frac{-\left(b_u^{(i)} - b_v^{(i)}\right)^2}{2h^2}\right] \left(b_u^{(i)} - b_v^{(i)}\right)\left(m_u^{(s)} - m_v^{(s)}\right)\delta_{ri}$$

$$\frac{\partial H_i}{\partial x_{rs}} = \frac{1}{N^2 h^3 \sqrt{2\pi}} \sum_{u=1}^{N} \frac{1}{p^{(i)}\left(b_u^{(i)}\right)} \sum_{v=1}^{N} \exp\left[\frac{-\left(b_u^{(i)} - b_v^{(i)}\right)^2}{2h^2}\right]$$
$$\times \left(b_u^{(i)} - b_v^{(i)}\right)\left(m_u^{(s)} - m_v^{(s)}\right)\delta_{ri}$$

$$\frac{\partial H}{\partial x_{rs}} = \sum_{i=1}^{d} \frac{\partial H_i}{\partial x_{rs}} = \frac{1}{N^2 h^3 \sqrt{2\pi}} \sum_{u=1}^{N} \frac{1}{p^{(r)}\left(b_u^{(r)}\right)} \times$$
$$\sum_{v=1}^{N} \exp\left[\frac{-\left(b_u^{(r)} - b_v^{(r)}\right)^2}{2h^2}\right] \left(b_u^{(r)} - b_v^{(r)}\right)\left(m_u^{(s)} - m_v^{(s)}\right).$$

After substituting $p^{(r)}\left(b_u^{(r)}\right)$ into $\dfrac{\partial H}{\partial x_{rs}}$, we arrive at

$$\begin{aligned}\frac{\partial H}{\partial x_{rs}} =& \frac{1}{Nh^2}\sum_{u=1}^{N}\frac{1}{\sum_{v=1}^{N}\exp\left[\frac{-\left(b_u^{(r)}-b_v^{(r)}\right)^2}{2h^2}\right]} \\ & \left\{ b_u^{(r)} m_u^{(s)} \sum_{v=1}^{N}\exp\left[\frac{-\left(b_u^{(r)}-b_v^{(r)}\right)^2}{2h^2}\right]\right. \\ & - b_u^{(r)}\sum_{v=1}^{N} m_v^{(s)}\exp\left[\frac{-\left(b_u^{(r)}-b_v^{(r)}\right)^2}{2h^2}\right] \\ & - m_u^{(s)}\sum_{v=1}^{N} b_v^{(r)}\exp\left[\frac{-\left(b_u^{(r)}-b_v^{(r)}\right)^2}{2h^2}\right] \\ & \left. + \sum_{v=1}^{N} b_v^{(r)} m_v^{(s)}\exp\left[\frac{-\left(b_u^{(r)}-b_v^{(r)}\right)^2}{2h^2}\right]\right\}\end{aligned}$$

which is added to $-\dfrac{\partial}{\partial \mathbf{X}}(\log|\det \mathbf{X}|) = -(\mathbf{X}^{\mathrm{T}})^{-1}$ to result in Eq. (15).

Acknowledgment

The authors would like to thank Prof. P.-A. Absil of Université catholique de Louvain, Belgium, for insightful discussion and interaction.

References

[1] Y. Xue, Y. Wang, J. Yang, Independent component analysis based on gradient equation and kernel density estimation, Neurocomputing, 72(7–9)(2009)1597–1604.

[2] R. Boscolo, H. Pan, V. P. Roychowdhury, Independent component analysis based on nonparametric density estimation, IEEE Transactions on Neural Networks, 15(1)(2004)55–65.

[3] K. Sengupta, P. Burman, R. Sharma, A non-parametric approach for independent component analysis using kernel density estimation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, Washington, DC, 27 June–2 July 2004, pp. 667–672.

[4] M. D. Plumbley, Geometrical methods for non-negative ICA: Manifolds, Lie groups and toral subalgebras, Neurocomputing, 67(2005)161–197.

[5] S. C. Douglas, Self-stabilized gradient algorithms for blind source separation with orthogonality constraints, IEEE Transactions on Neural Networks, 11(6)(2000)1490–1497.

[6] M. D. Plumbley, Lie group methods for optimization with orthogonality constraints, Lecture Notes in Computer Science, vol. 3195, Springer, Berlin, 2004, pp. 1245–1252.

[7] S. Fiori, Quasi-geodesic neural learning algorithms over the orthogonal group: A tutorial, The Journal of Machine Learning Research, 6(2005)743–781.

[8] T. E. Abrudan, J. Eriksson, V. Koivunen, Steepest descent algorithms for optimization under unitary matrix constraint, IEEE Transactions on Signal Processing, 56(3)(2008)1134–1147.

[9] T. Abrudan, J. Eriksson, V. Koivunen, Conjugate gradient algorithm for optimization under unitary matrix constraint, Signal Processing, 89(9)(2009)1704–1714.

[10] S. E. Selvan, A. Mustăţea, C. C. Xavier, J. Sequeira, Accurate estimation of ICA weight matrix by implicit constraint imposition using Lie group, IEEE Transactions on Neural Networks, 20(10)(2009)1565–1580.

[11] Y. Nishimori, S. Akaho, M. D. Plumbley, Riemannian optimization method on the flag manifold for independent subspace analysis, Lecture Notes in Computer Science, vol. 3889, Springer, Berlin, 2006, pp. 295–302.

[12] T.-W. Lee, T. Wachtler, T. J. Sejnowski, Color opponency constitutes a sparse representation for the chromatic structure of natural scenes, in: T. K. Leen, T. G. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems, vol. 13, MIT Press, Cambridge, MA, 2001, pp. 866–872.

[13] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, New York, 2001.

[14] M. S. Lewicki, T. J. Sejnowski, Learning overcomplete representations, Neural Computation, 12(2)(2000)337–365.

[15] S. C. Douglas, S.-i. Amari, S.-Y. Kung, On gradient adaptation with unit-norm constraints, IEEE Transactions on Signal Processing, 48(6)(2000)1843–1847.

[16] H. Shen, K. Diepold, K. Hüper, Geometric algorithms for the non-whitened one-unit linear independent component analysis problem, in: Proceedings of the 15th IEEE Workshop on Statistical Signal Processing, Cardiff, UK, 31 August–3 September 2009, pp. 381–384.

[17] P.-A. Absil, K. A. Gallivan, Joint diagonalization on the oblique manifold for independent component analysis, in: Proceedings of the 31st IEEE International Conference on Acoustics, Speech and Signal Processing, vol. V, Toulouse, France, 14–19 May 2006, pp. 945–948.

[18] H. Shen, K. Hüper, Block Jacobi-type methods for non-orthogonal joint diagonalisation, in: Proceedings of the 34th IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009, pp. 3285–3288.

[19] K. E. Hild, D. Erdogmus, J. Principe, Blind source separation using Rényi's mutual information, IEEE Signal Processing Letters, 8(6)(2001)174–176.

[20] D.-T. Pham, F. Vrins, M. Verleysen, On the risk of using Rényi's entropy for blind source separation, IEEE Transactions on Signal Processing, 56(10)(2008)4611–4620.

[21] D.-T. Pham, Fast algorithms for mutual information based independent component analysis, IEEE Transactions on Signal Processing, 52(10)(2004)2690–2700.

[22] A. Chen, Fast kernel density independent component analysis, Lecture Notes in Computer Science, vol. 3889, Springer, Berlin, 2006, pp. 24–31.

[23] S. Shwartz, M. Zibulevsky, Y. Y. Schechner, ICA using kernel entropy estimation with NlogN complexity, Lecture Notes in Computer Science, vol. 3195, Springer, Berlin, 2004, pp. 422–429.

[24] D.-M. Tsai, S.-C. Lai, Independent component analysis-based background subtraction for indoor surveillance, IEEE Transactions on Image Processing, 18(1)(2009)158–167.

[25] D. Gabay, Minimizing a differentiable function over a differential manifold, Journal of Optimization Theory and Applications, 37(2)(1982)177–219.

[26] A. Edelman, T. A. Arias, S. T. Smith, The geometry of algorithms with orthogonality constraints, SIAM Journal on Matrix Analysis and Applications, 20(2)(1999)303–353.

[27] S. Fiori, Learning independent components on the orthogonal group of matrices by retractions, Neural Processing Letters, 25(3)(2007)187–198.

[28] I. Yamada, T. Ezaki, An orthogonal matrix optimization by dual Cayley parametrization technique, in: Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan, 1–4 April 2003, pp. 35–40.

[29] P. C. Yuen, J. H. Lai, Face representation using independent component analysis, Pattern Recognition, 35(6)(2002)1247–1257.

[30] S.-S. Chiang, C.-I. Chang, I. W. Ginsberg, Unsupervised hyperspectral image analysis using independent component analysis, in: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, vol. 7, Honolulu, HI, 24–28 July 2000, pp. 3136–3138.

[31] J.-M. Gaucel, M. Guillaume, S. Bourennane, Non orthogonal component analysis: Application to anomaly detection, Lecture Notes in Computer Science, vol. 4179, Springer, Berlin, 2006, pp. 1198–1209.

[32] B. Afsari, P. S. Krishnaprasad, Some gradient based joint diagonalization methods for ICA, Lecture Notes in Computer Science, vol. 3195, Springer, Berlin, 2004, pp. 437–444.

[33] P.-A. Absil, R. Mahony, R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, Princeton, NJ, 2008.

[34] E. Polak, Optimization: Algorithms and Consistent Approximations, Springer-Verlag, New York, 1997.

[35] C. Qi, K. A. Gallivan, P.-A. Absil, Riemannian BFGS algorithm with applications, in: M. Diehl, F. Glineur, W. Michiels (Eds.), Recent Trends

in Optimization and Its Applications in Engineering, Springer, 2010, pp. 183–192.

[36] N. T. Trendafilov, R. A. Lippert, The multimode Procrustes problem, Linear Algebra and Its Applications, 349(2002)245–264.

[37] C. Qi, Numerical optimization on Riemannian manifolds, Ph.D. Thesis, Department of Mathematics, Florida State University, 2011.

[38] U. Depczynski, J. Stöckler, A differential geometric approach to equidistributed knots on Riemannian manifolds, in: C. K. Chui, L. L. Schumaker (Eds.), Approximation Theory IX, Theoretical Aspects, vol. 1, Vanderbilt University Press, Nashville, TN, 1998, pp. 99–106.

[39] N. D. Buono, C. Elia, Computation of few Lyapunov exponents by geodesic based algorithms, Future Generation Computer Systems, 19(3)(2003)425–430.

[40] B. Silverman, Density Estimation for Statistics and Data Analysis, Chapman & Hall, New York, 1985.

[41] N. Vlassis, Y. Motomura, Efficient source adaptivity in independent component analysis, IEEE Transactions on Neural Networks, 12(3)(2001)559–566.

[42] C. Yang, R. Duraiswami, N. A. Gumerov, L. Davis, Improved fast Gauss transform and efficient kernel density estimation, in: Proceedings of the 9th IEEE International Conference on Computer Vision, vol. 1, Nice, France, 13–16 October 2003, pp. 664–671.

[43] C. Yang, R. Duraiswami, L. Davis, Efficient kernel machines using the improved fast Gauss transform, in: L. K. Saul, Y. Weiss, L. Bottou (Eds.), Advances in Neural Information Processing Systems 17, MIT Press, 2005, pp. 1561–1568.

[44] J. Nocedal, S. Wright, Numerical Optimization, Springer-Verlag, New York, 1999.

[45] J. C. Gilbert, J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, SIAM Journal on Optimization, 2(1)(1992)21–42.

[46] W. W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, Pacific Journal of Optimization, 2(1)(2006)35–58.

[47] W. W. Hager, H. Zhang, CG_descent, a conjugate gradient method with guaranteed descent, ACM Transactions on Mathematical Software, 32(1)(2006)113–137.

[48] Y. H. Dai, Y. Yuan, An efficient hybrid conjugate gradient method for unconstrained optimization, Annals of Operations Research, 103(1–4)(2001)33–47.

[49] Y. H. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM Journal on Optimization, 10(1)(1999)177–182.

[50] M. R. Hestenes, E. L. Stiefel, Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards, 49(6)(1952)409–436.

[51] R. Fletcher, Practical Methods of Optimization, second ed, Wiley, New York, 1987.

[52] W. W. Hager, A derivative-based bracketing scheme for univariate minimization and the conjugate gradient method, Computers & Mathematics with Applications, 18(9)(1989)779–795.

[53] Y. H. Dai, Y. Yuan, Nonlinear Conjugate Gradient Methods, Shanghai Science and Technology Publisher, Shanghai, 2000.

[54] M. J. D. Powell, Restart procedures for the conjugate gradient method, Mathematical Programming, 12(2)(1977)241–254.

[55] S. E. Selvan, Generating linear combination of spectral images with mutually exclusive specific information, Ph.D. Thesis, Université de la Méditerranée, France, 2007.

[56] C. Yang, R. Duraiswami, D. DeMenthon, L. S. Davis, Mean-shift analysis using quasiNewton methods, in: Proceedings of the IEEE International Conference on Image Processing, vol. 2, Barcelona, Spain, 14–18 September 2003, pp. 447–450.

[57] D. Byatt, I. D. Coope, C. J. Price, Effect of limited precision on the BFGS quasi-Newton algorithm, ANZIAM Journal, 45(2004)C283–C295.

[58] B. Savas, L.-H. Lim, Quasi-Newton methods on Grassmannians and multilinear approximations of tensors, SIAM Journal on Scientific Computing, 32(6)(2010)3352–3393.

[59] J. H. Manton, Optimization algorithms exploiting unitary constraints, IEEE Transactions on Signal Processing, 50(3)(2002)635–650.

[60] I. Brace, J. H. Manton, An improved BFGS-on-manifold algorithm for computing weighted low rank approximations, in: Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 24–28 July 2006, pp. 1735–1738.

[61] J. Karhunen, E. Oja, L. Wang, R. Vigário, J. Joutsensalo, A class of neural networks for independent component analysis, IEEE Transactions on Neural Networks, 8(3)(1997)486–504.

[62] A. Hyvärinen, E. Oja, Independent component analysis: Algorithms and applications, Neural Networks, 13(4–5)(2000)411–430.

[63] J.-F. Cardosa, A. Souloumiac, Blind beamforming for non Gaussian signals, IEE Proceedings-F, 140(6)(1993)362–370.

[64] S. Makeig, A. J. Bell, T.-P. Jung, T. J. Sejnowski, Independent component analysis of electroencephalographic data, in: D. S. Touretzky, M. C. Mozer, M. E. Hasselmo (Eds.), Advances in Neural Information Processing Systems, vol. 8, MIT Press, Cambridge, MA, 1996, pp. 145–151.

[65] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Transactions on Neural Networks, 10(3)(1999)626–634.

[66] J.-T. Chien, B.-C. Chen, A new independent component analysis for speech recognition and separation, IEEE Transactions on Audio, Speech, and Language Processing, 14(4)(2006)1245–1254.

[67] J.-T. Chien, H.-L. Hsieh, S. Furui, A new mutual information measure for independent component analysis, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, 30 March–4 April 2008, pp. 1817–1820.

[68] S.-i. Amari, A. Cichocki, H. Yang, A new learning algorithm for blind signal separation, in: D. S. Touretzky, M. C. Mozer, M. E. Hasselmo (Eds.), Advances in Neural Information Processing Systems, vol. 8, MIT Press, Cambridge, MA, 1996, pp. 757–763.

[69] U. Amato, M. Larobina, A. Antoniadis, B. Alfano, Segmentation of magnetic resonance brain images through discriminant analysis, Journal of Neuroscience Methods, 131(1–2)(2003)65–74.