Abstract

Independent Component Analysis (ICA) is a popular model for blind signal separation. The ICA model assumes that a number of independent source signals are linearly mixed to form the observed signal. Traditional ICA algorithms typically aim to recover the mixing matrix, the inverse of which can be applied to data in order to recover the latent independent signals. However, in the presence of noise, this demixing process is non-optimal for signal recovery as measured by signal-to-interference-plus-noise ratio (SINR), even if the mixing matrix is recovered exactly.

This paper has two main contributions. First, we show how any solution to the mixing matrix reconstruction problem can be used to construct an SINR-optimal ICA demixing. The proposed method is optimal for any noise model and applies in the underdetermined setting when there are more source signals than observed signals. Second, we improve the recently proposed Gradient Iteration ICA [21] algorithm to obtain provable and practical SINR optimal signal recovery for Gaussian noise with an arbitrary covariance matrix. We also simplify the original algorithm making the cumbersome quasi-orthogonalization step unnecessary, leading to improved computational performance.

1 Introduction

In the typical ICA setting, we observe \( n \)-dimensional realizations \( \mathbf{x}(1), \ldots, \mathbf{x}(N) \) of a latent variable model \( \mathbf{X} = \sum_{k=1}^{m} S_k A_k = \mathbf{A} \mathbf{S} \) where \( A_k \) denotes the \( k \)th column of the \( n \times m \) mixing matrix \( \mathbf{A} \) and \( \mathbf{S} = (S_1, \ldots, S_m)^T \) is the unseen latent random vector of “signals”. It is assumed that \( S_1, \ldots, S_m \) are independent and non-Gaussian. The source signals and entries of \( \mathbf{A} \) may be either real- or complex-valued. For simplicity, we will assume throughout that \( \mathbf{S} \) has zero mean, as this may be achieved in practice by centering the observed data.

In most applications of ICA (e.g., speech separation [17], MEG/EEG artifact removal [20] and others) one cares about recovering the signals \( s(1), \ldots, s(N) \). This is known as the source recovery problem. This is typically done by first recovering the matrix \( \mathbf{A} \) (up to an appropriate scaling of the column directions[1]). At first, source recovery and recovering the mixing matrix \( \mathbf{A} \) appear to be essentially equivalent. Let \( \mathbf{A}^\dagger \) denote the Moore-Penrose pseudo-inverse of \( \mathbf{A} \). If \( \mathbf{A} \) has full column rank, then \( \mathbf{A}^\dagger \) is a left inverse of \( \mathbf{A} \) and \( s(t) = \mathbf{A}^\dagger \mathbf{x}(t) \) recovers the latent sources signals.

Consider now the model \( \mathbf{X} = \mathbf{A} \mathbf{S} + \mathbf{\eta} \) with additive 0-mean noise \( \mathbf{\eta} \) independent of \( \mathbf{S} \). In this model, the exact recovery of the latent sources \( s(t) \) becomes impossible even if \( \mathbf{A} \) is known exactly. Moreover, in the noisy ICA model \( \mathbf{A} \) and \( \mathbf{s} \) are not defined uniquely as we cannot generally...
distinguish the “true” signal from noise (see the Discussion in section 2). The natural question is then how to recover the signals optimally. It turns out that there is an inherent ambiguity in the setting as part of the “noise” can be incorporated into the “signal” preserving the form of the model. Thus, the natural measure of optimality, the signal to interference-plus-noise ratio (SINR), is not well-defined. We consider recovered signals of the form $\hat{S}(B) := BX$ for a choice of $m \times n$ demixing matrix $B$. Signal recovery is considered optimal if the coordinates of $\hat{S}(B) = (\hat{S}_1(B), \ldots, \hat{S}_m(B))$ are best possible approximations in terms SINR to the source signals within any fixed model $X = AS + \eta$. Remarkably, we are able to show that the SINR optimal demixing (beamforming) matrix is invariant to changes of decomposition of data into signal plus noise: $X = A'S' + \eta'$. Moreover, it turns out that optimal beamforming can be constructed from the matrix $A$ from any fixed model and the data without any assumptions on the noise distribution.

Interestingly, the optimal demixing matrix in the noisy case is generally different from the optimal in the noiseless case. In fact, the optimal beamforming matrix for the noisy case is different from $A^\dagger$ unless the columns of $A$ are orthogonal after whitening the data. This is true in the noiseless case but generally not true in the noisy setting, even when the noise is white Gaussian.

Still, extracting $A$ from data may be difficult without further assumptions. In the case when the noise is Gaussian (with an arbitrary covariance matrix), we develop a practical algorithm for provably optimal signal recovery.

To summarize, we have the following main contributions. In section 2 we demonstrate how any reconstruction of the mixing matrix $A$ may be used to perform SINR-optimal signal recovery in the general underdetermined noisy settings. For this result, we only require that the noise $\eta$ be 0-mean and independent of $S$.

In section 3 we improve upon the gradient iteration ICA (GI-ICA) algorithm [21] for performing optimal signal recovery in the presence of additive Gaussian noise with arbitrary covariance. In particular, we simplify the original GI-ICA algorithm by removing the somewhat complicated quasi-orthogonalization preprocessing step, replacing it with modified gradient iteration using an (not necessarily positive definite) “inner product”.

Finally, in section 4 we demonstrate experimentally that our proposed algorithms for ICA outperform existing practical algorithms at the task of noisy signal recovery, including those specifically designed for beamforming, when given sufficiently many samples. Moreover, all existing practical algorithms for noisy source recovery have a bias and cannot recover the optimal demixing matrix even with infinite samples. We also show that our modified version of GI-ICA requires fewer samples than the original version of GI-ICA to perform ICA accurately.

1.1 The Indeterminacies of ICA

**Notation:** In this paper, we use $M^*$ to denote the entry-wise complex conjugate of a matrix $M$, $M^T$ to denote its transpose, and $M^H$ to denote its conjugate transpose.

Before proceeding with our results, it is important to discuss the somewhat subtle issue of indeterminacies in ICA. These ambiguities arise from the fact that the observed $X$ may have multiple decompositions into ICA models $X = AS + \eta$ and $X = A'S' + \eta'$.

Noise free ICA has two natural indeterminacies. For any nonzero constant $\alpha$, the contribution of the $k$th component $A_kS_k$ to the model can equivalently be obtained by replacing $A_k$ with $\alpha A_k$ and $S_k$ with the rescaled signal $\frac{1}{\alpha} S_k$. To lessen this scaling indeterminacy, we use the convention $\text{cov}(S) = I$ throughout this paper. As such, each source $S_k$ (or equivalently each $A_k$) is defined up to a choice of sign (a unit modulus factor in the complex case). In addition, there is an ambiguity
in the order of the latent signals. For any permutation \( \pi \) of \([m]\) (where \([m]\) := \(\{1, \ldots, m\}\)), the ICA models \( X = \sum_{k=1}^{m} S_k A_k \) and \( X = \sum_{k=1}^{m} S_{\pi(k)} A_{\pi(k)} \) are indistinguishable. In the noise free setting, \( A \) is said to be recovered if we recover each column of \( A \) up to a choice of sign (or up to a unit modulus factor in the complex case) and an unknown permutation. As the sources \( S_1, \ldots, S_m \) are only defined up to the same indeterminacies, inverting the recovered matrix \( \tilde{A} \) to obtain a demixing matrix works for signal recovery.

In the noisy ICA setting, there is an additional indeterminacy in the definition of the source signals. Consider \( \xi \) to be 0-mean axis aligned Gaussian random vector. Then, the noisy ICA model \( X = A(S + \xi) + \eta \) in which \( \xi \) is considered part of the latent source signal \( S' = S + \xi \), and the model \( X = AS + (A\xi + \eta) \) in which \( \xi \) is part of the noise are indistinguishable. In particular, the latent source \( S \) and its covariance are ill-defined. Due to this extra indeterminacy, the lengths of the columns of \( A \) no longer have a fully defined meaning even when we assume \( \text{cov}(S) = I \). In the noisy setting, \( A \) is said to be recovered if we obtain the columns of \( A \) up to non-zero scalar multiplicative factors and an arbitrary permutation.

The last indeterminacy is the most troubling as it suggests that the power of each source signal is itself ill-defined in the noisy setting. Despite this indeterminacy, it is possible to perform an SINR-optimal demixing without additional assumptions about what portion of the signal is source and what portion is noise. In section 2 we will see that SINR-optimal source recovery takes on a simple form: Given any solution \( \tilde{A} \) which recovers \( A \) up to the inherent ambiguities of noisy ICA, then \( \tilde{A}^H \text{cov}(X)^{\dagger} \) is an SINR-optimal demixing matrix.

1.2 Related Work and Contributions

Independent Component Analysis is probably the most used model for Blind Signal Separation. It has seen a large number of applications and has generated a vast literature, including in the noisy and underdetermined settings. We refer the reader to the books [12, 6] for a broad overview of the subject. Early on, it was observed for instance by Cardoso [4] that ICA algorithms based solely on higher order cumulant statistics are invariant to additive Gaussian noise. This observation has allowed the creation of many algorithms for recovering the ICA mixing matrix in the noisy (and often underdetermined) settings.

Research on cumulant-based noisy ICA can largely be split into several lines of work which we only highlight here. Some algorithms such as FOOBI [4] and BIOMe [1] directly use the tensor structure of higher order cumulants. In another line of work, De Lathauwer et al. [7] and Yeredor [23] have suggested algorithms which jointly diagonalize cumulant matrices in a manner somewhat reminiscent of the noise-free JADE algorithm [3]. In addition, Yeredor [22] and Goyal et al. [10] have proposed ICA algorithms based on random directional derivatives of the second characteristic function.

Each line of work has its advantages and disadvantages. For the tensor methods it is difficult (i.e., typically not done) to handle the case where the latent source signals have fourth cumulants with differing signs. The joint diagonalization algorithms and the tensor based algorithms tend to be practical in the sense that they use redundant cumulant information in order to achieve more accurate results. However, they also have a higher memory complexity than popular noise free ICA algorithms such as FastICA [11]. When dealing with high dimensional data, this memory overhead can become prohibitive. Finally, the methods based on random directional derivatives of the second characteristic function rely heavily upon randomness in a way which is not required by the tensor or joint diagonalization methods.

We continue a line of research started by Arora et al. [2] and Voss et al. [21] on fully determined noisy ICA which addresses some of these practical issues by using a deflationary approach
reminiscent of FastICA. In particular, the GI-ICA algorithm [21] has the same memory complexity as FastICA and does not rely on randomness. The GI-ICA algorithm as originally proposed is able to handle latent sources signals which have fourth cumulants with differing signs; however both GI-ICA and Arora et al. [2] require that the data be quasi-orthogonalized, that is the data is preprocessed using higher cumulant techniques to orthogonalize the latent source signals. We demonstrate that such preprocessing is unnecessary. In particular, we modify the GI-ICA algorithm to work within a (not necessarily positive definite) inner product space instead. Experimentally, this leads to improved demixing performance as the GI-ICA preprocessing step requires more samples to accurately estimate. In addition, we generalize GI-ICA to the case of complex latent source signals.

Finally, another line of work attempts to perform SINR-optimal source recovery in the noisy ICA setting. It was noted by Koldovský and Tichavský [14] that for noisy ICA, traditional ICA algorithms such as FastICA and JADE actually outperform algorithms which first recover \( A \) in the noisy setting and then use the resulting approximation of \( A^\dagger \) to perform demixing. It was further observed that \( A^\dagger \) is not the optimal demixing matrix for source recovery. Later, Koldovský and Tichavský [16] proposed an algorithm based on FastICA which performs a low SINR-bias beamforming.

### 2 SINR Optimal Recovery in Noisy ICA

Consider \( B \) an \( m \times n \) demixing matrix, and define \( \hat{\mathbf{S}}(\mathbf{B}) := B\mathbf{X} \) the resulting approximation to \( \mathbf{S} \). It will also be convenient to estimate the source signal \( \mathbf{S} \) one coordinate at a time: Given a row vector \( \mathbf{b} \), we define \( \hat{\mathbf{S}}(\mathbf{b}) := \mathbf{bX} \). If \( \mathbf{b} = B_k \) (the \( k \)th row of \( B \)), then \( \hat{\mathbf{S}}(\mathbf{b}) = [\mathbf{S}(\mathbf{B})]_k = \hat{S}_k(\mathbf{B}) \) is our estimate to the \( k \)th latent signal \( S_k \). Within a specific ICA model \( \mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{\eta} \), signal to interference-plus-noise ratio (SINR) is defined by the following equation:

\[
\text{SINR}_k(\mathbf{b}) := \frac{\text{var}(\mathbf{b}_k \mathbf{A}_k \mathbf{S}_k)}{\text{var}(\mathbf{b}_k \mathbf{A}_k \mathbf{S}_k) + \text{var}(\mathbf{b}\mathbf{\eta})} = \frac{\text{var}(\mathbf{b}_k \mathbf{A}_k \mathbf{S}_k)}{\text{var}(\mathbf{b}_k \mathbf{A}_k \mathbf{X}) - \text{var}(\mathbf{b}_k \mathbf{A}_k \mathbf{S}_k)}.
\]

\( \text{SINR}_k \) is the variance of the contribution of \( k \)th source divided by the variance of the noise and interference contributions within the signal.

Given access to the mixing matrix \( \mathbf{A} \), we define \( B_{\text{opt}} = \mathbf{A}^H(\mathbf{A}\mathbf{A}^H + \text{cov}(\mathbf{\eta}))^{\dagger} \). Since \( \text{cov}(\mathbf{X}) = \mathbf{A}\mathbf{A}^H + \text{cov}(\mathbf{\eta}) \), this may be rewritten as \( B_{\text{opt}} = \mathbf{A}^H \text{cov}(\mathbf{X})^{\dagger} \). Here, \( \text{cov}(\mathbf{X})^{\dagger} \) may be estimated from data, but due to the ambiguities of the noisy ICA model, \( \mathbf{A} \) (and specifically its column norms) cannot be estimated from data.

Koldovský and Tichavský [14] observed in the setting where \( \eta \) is a white, additive Gaussian noise that \( B_{\text{opt}} \) jointly maximizes \( \text{SINR}_k \) for each \( k \in [m] \), i.e., \( \text{SINR}_k \) takes on its maximal value at \( (B_{\text{opt}})_k \). Below in Proposition 1, we generalize this result to include arbitrary non-spherical, potentially non-Gaussian noise.

Before proceeding, it is interesting to note that even after the data is whitened, i.e. made to have \( \text{cov}(\mathbf{X}) = \mathbf{I} \), the optimal SINR solution is different from the optimal solution in the noiseless case unless \( \mathbf{A} \) is an orthogonal (unitary in the complex case) matrix, i.e. \( \mathbf{A}^\dagger = \mathbf{A}^H \). This is generally not the case, even if \( \eta \) is white Gaussian noise.

**Proposition 1.** For each \( k \in [m] \), \((B_{\text{opt}})_k\) is a maximizer of \( \text{SINR}_k \).
The proof of Proposition 1 is deferred to the end of this section.

Since SINR is scale invariant, Proposition 1 implies that any matrix of the form $DB_{opt} = DA^H \text{cov}(X)^\dagger$ where $D$ is a diagonal scaling matrix (with non-zero diagonal entries) is an SINR-optimal demixing matrix. More formally, we have the following result.

**Theorem 2.** Suppose that $\hat{A}$ is an $n \times m$ matrix containing the columns of $A$ up to scale and an arbitrary permutation. That is, there exists a permutation $\pi$ of $[m]$ and non-zero constants $\alpha_1, \ldots, \alpha_m$ such that $\alpha_k \hat{A}_{\pi(k)} = A_k$ for each $k \in [m]$. Then, $(\hat{A}^H \text{cov}(X)^\dagger)_{\pi(k)}$ is a maximizer of SINR.

Note that $\hat{A}$ in Theorem 2 is any matrix recovering $A$ up to the ambiguities of the noisy ICA model. In particular $(\hat{A}^H \text{cov}(X)^\dagger)$ constructed in one ICA model decomposition for $X$ will be optimal in any other ICA model for $X$. As such, mixing matrix recovery in noisy ICA gives rise to an SINR-optimal source recovery. In the noise-free case, the SINR-optimal source recovery simplifies to be $A^\dagger$.

**Corollary 3.** Suppose that $X = AS$ is a noise free (possibly underdetermined) ICA model. Suppose that $A \in \mathbb{R}^{n \times m}$ contains the columns of $A$ up to scale and permutation, i.e., there exists diagonal matrix $D$ with non-zero entries and a permutation matrix $\Pi$ such that $\hat{A} = AD\Pi$. Then $\hat{A}^\dagger$ is an SINR-optimal demixing matrix.

**Proof.** By Theorem 2 $(AD^{-1}\Pi)^H \text{cov}(X)^\dagger$ is an SINR-optimal demixing matrix. Expanding, we obtain:

$$(AD^{-1}\Pi)^H \text{cov}(X)^\dagger = \Pi^H D^{-1}A^H (AA^H)^\dagger = \Pi^H D^{-1}A^\dagger = (AD\Pi)^\dagger = \hat{A}^\dagger.$$  

Corollary 3 is consistent with known beamforming results. In particular, it is known that $A^\dagger$ is an optimal (in terms of minimum mean squared error) beamforming matrix for underdetermined ICA 18 section 3B.

**Proof of Proposition 1.** This proof shows the connection between two notions of optimality, minimum mean squared error and SINR. The mean squared error of the recovered signal $\hat{S}(b)$ from $k^{th}$ latent signal is defined as $\text{MSE}_k(b) := \mathbb{E}[[S_k - \hat{S}(b)]^2]$. It has been shown 13 equation 39 that $B_{opt}$ jointly minimizes the mean squared errors of the recovered signals. In particular, if $b = (B_{opt})_k$, then $b$ is a minimizer of $\text{MSE}_k(b)$.

We will first show that finding a matrix $B$ which minimizes the mean squared error has the side effect of maximizing the magnitude of the Pearson correlations $\rho_{S_k, \hat{S}_k(B)}$ for each $k \in [m]$, where $\rho_{S_k, \hat{S}_k(B)} := \frac{\mathbb{E}[S_k \hat{S}_k^*(B)]}{\sigma_{S_k} \sigma_{\hat{S}_k(B)}}$. We will then demonstrate that if $B$ is a maximizer of $|\rho_{S_k, \hat{S}_k(B)}|$, then $B_k$ is a maximizer of SINR$_k$. These two facts imply the desired result. We will use the convention that $\rho_{S_k, \hat{S}_k(B)}$ is 0 if $\sigma_{\hat{S}_k(B)} = 0$.

We fix a $k \in [m]$. We have:

$$\text{MSE}_k(b) = \mathbb{E}[S_k S_k^* - 2 \text{Re}(S_k \hat{S}_k(b)) + \hat{S}(b)\hat{S}^*(b)] = 1 - 2\sigma_{\hat{S}(b)} \text{Re}(\rho_{S_k, \hat{S}(b)}) + \sigma_{\hat{S}(b)}^2.$$  

5
Letting $\omega = \text{sgn}(\rho_{S_k, \hat{S}(b)})$, we obtain
\[
\rho_{S_k, \hat{S}(\omega b)} = \frac{\mathbb{E}[S_k \hat{S}^* (\omega b)]}{\sigma_S \sigma_{\hat{S}(\omega b)}} = \omega^* \frac{\mathbb{E}[S_k \hat{S}^* (b)]}{\sigma_S \sigma_{\hat{S}(b)}} = |\rho_{S_k, \hat{S}(b)}| .
\] (2)

Further, $\text{MSE}_k(\omega b) = 1 - 2\sigma_{\hat{S}(b)}|\rho_{S_k, \hat{S}(b)}| + \sigma_{\hat{S}(b)}^2 \leq \text{MSE}_k(b)$ with equality if and only if $\rho_{S_k, \hat{S}(b)}$ is real and non-negative. As such, all global minima of $\text{MSE}_k$ are contained in the set $A = \{b \mid \rho_{S_k, \hat{S}(b)} \in [0, 1]\}$, and we may restrict our investigation to this set.

We define a function $g(x, y) := 1 - 2xy + y^2$ such that under the change of variable $x(b) = \sigma_{\hat{S}(b)}$ and $y(b) = \rho_{S_k, \hat{S}(b)}$, we obtain $\text{MSE}_k(b) = g(x, y)$. Let $M = \max_{b \in A} \rho_{S_k, \hat{S}(b)}$ and let $y_0 \in [0, M]$ be fixed. Then, $\arg \min_{x \in \mathbb{R}} g(x, y_0) = y_0$ with the resulting value $g(y_0, y_0) = 1 - y_0^2$. As such, the minimum of $g(x, y)$ over the domain $\mathbb{R} \times [0, M]$ occurs when $x = y = M$. If $M = 0$, then the choice of $\xi = 0$ satisfies that $x(\xi) = y(\xi) = 0$, making $\text{MSE}_k(\xi) = g(x, y)$ the global minimum of $\text{MSE}_k$. If $M \neq 0$, then we may choose $\xi$ such that $y(\xi) = \rho_{S_k, \hat{S}(\xi)} = M$. As $\sigma_{\hat{S}(\xi)} > 0$ must hold, it follows that there exists $\alpha \in (0, \infty)$ such that setting $\zeta = \alpha\xi$, we obtain $\{\sigma_{\hat{S}(\zeta)} = x(\xi) = y(\xi)$. Since $y(\xi) = \rho_{S_k, \hat{S}(b)}$ is scale invariant, we obtain that $x(\zeta) = y(\zeta) = y(\xi) = M$, making $\zeta$ a global minimum of $\text{MSE}_k$. In both cases, it follows that if $b$ minimizes $\text{MSE}_k(b)$, then $b$ maximizes $\rho_{S_k, \hat{S}(b)}$ over $A$.

From equation (2), we see that $\max_{b \in \mathbb{C}^n} |\rho_{S_k, \hat{S}(b)}| = \max_{b \in A} \rho_{S_k, \hat{S}(b)}$. Thus if $b$ is a minimizer of $\text{MSE}_k(\omega b)$, then $b$ is also a maximizer of $|\rho_{S_k, \hat{S}(b)}|$ as claimed.

We now demonstrate that $b$ is a maximizer of $|\rho_{S_k, \hat{S}(b)}|$ if and only if it is also a maximizer of $\text{SINR}_k(b)$. Under the conventions that $\frac{\xi}{\eta} = +\infty$ when $x > 0$ and that $\frac{\eta}{\eta} = \infty$ where $s = -1$ for maximization problems and $s = +1$ for minimization problems, the following problems have equivalent solution sets over choices of $b$:
\[
\max_{b} \text{SINR}_k(b) \equiv \max_{b} \frac{\mathbb{E}[|bA_k S_k|^2]}{\mathbb{E}|bA_k S_k|^2 - \mathbb{E}|bA_k S_k|^2} \equiv \max_{b} \frac{|\mathbb{E}[S_k \hat{S}^*(b)]|^2}{\mathbb{E}|\hat{S}(b)|^2 - \mathbb{E}|S_k \hat{S}^*(b)|^2} \equiv \min_{b} \frac{\mathbb{E}|\hat{S}(b)|^2}{\mathbb{E}|S_k \hat{S}^*(b)|^2} \equiv \max_{b} \frac{|\mathbb{E}[S_k \hat{S}^*(b)]|^2}{\mathbb{E}|\hat{S}(b)|^2} .
\]

In the above, the first equivalence is a rewriting of equation (1). To see the second equivalence, we note that $|\mathbb{E}[S_k \hat{S}^*(b)]|^2 = |\mathbb{E}[S_k (bA S + b\eta)]^*|^2 = |bA_k|^2$ using the independence of $S_k$ from all other terms. Then, noting that $|bA_k|^2 = \mathbb{E}|bA_k S_k|^2$ gives the equivalence. The fourth equivalence is only changing the problem by the additive constant $-1$.

3 Gradient Iteration ICA Revisited

In Section 2 we demonstrated how to perform optimal beamforming in ICA given that we recover $A$ up to the inherent ambiguities. In this section, we provide an algorithm for recovering $A$ in
the “fully determined” setting where \( m \leq n \). In particular, we extend the Gradient Iteration ICA (GI-ICA) algorithm of Voss et al. [21] to include the case where the source signals are not orthogonalized. We sketch the original GI-ICA algorithm in Section 3.1, and then provide our new variant of GI-ICA in Section 3.2. For simplicity, we limit this discussion to the case of real-valued signals. We show how to construct GI-ICA for complex-valued signals in Appendix A.

In this section we assume that \( m \leq n \), that \( m \) is known, and that the columns of \( A \) are linearly independent.

### 3.1 GI-ICA with Orthogonality

The gradient iteration relies on the properties of cumulants. We will focus on the fourth cumulant, though similar constructions may be given using other even order cumulants of higher order. For a zero-mean random variable \( X \), the fourth order cumulant may be defined as \( \kappa_4(X) := E[X^4] - 3E[X^2]^2 \) [see 6, Chapter 5, Section 1.2]. Higher order cumulants have nice algebraic properties which make them useful for ICA. In particular, \( \kappa_4 \) has the following properties:

1. (Independence) If \( X \) and \( Y \) are independent random variables, then \( \kappa_4(X + Y) = \kappa_4(X) + \kappa_4(Y) \).
2. (Homogeneity) If \( \alpha \) is a scalar, then \( \kappa_4(\alpha X) = \alpha^4 \kappa_4(X) \).
3. (Vanishing Gaussians) If \( X \) is normally distributed then \( \kappa_4(X) = 0 \).

In this section, we consider a noisy ICA model \( X = AS + \eta \) where \( \eta \) is a 0-mean Gaussian and independent of \( S \). We consider the following function defined on the unit sphere: \( f(u) := \kappa_4((X, u)) \). Expanding \( f(u) \) using the above properties we obtain:

\[
f(u) = \kappa_4 \left( \sum_{k=1}^{m} \langle A_k, u \rangle S_k + \langle u, \eta \rangle \right)
= \sum_{k=1}^{m} \langle A_k, u \rangle^4 \kappa_4(S_k)
\]

Taking derivatives we obtain:

\[
\nabla f(u) = 4 \sum_{k=1}^{m} \langle A_k, u \rangle^3 \kappa_4(S_k) A_k \tag{3}
\]

\[
\mathcal{H} f(u) = 12 \sum_{k=1}^{m} \langle A_k, u \rangle^2 \kappa_4(S_k) A_k^T A_k^T
= AD(u) A^T \tag{4}
\]

where \( D(u) \) is a diagonal matrix with entries \( D(u)_{kk} = 12 \langle A_k, u \rangle^2 \kappa_4(S_k) \).

Voss et al. [21] introduced GI-ICA as a fixed point algorithm under the assumption that the columns of \( A \) are orthogonal but not necessarily unit vectors. The main idea is that the update \( u \leftarrow \frac{\nabla f(u)}{\|\nabla f(u)\|} \) is a form of a generalized power iteration. From equation \([3]\), each \( A_k \) may be considered as a direction in a hidden orthogonal basis of the space. During each iteration, the \( A_k \) coordinate of \( u \) is raised to the 3\(^{rd}\) power and multiplied by a constant. Treating this iteration as a fixed point update, it was shown that given a random starting point, this iterative procedure converges rapidly to one of the columns of \( A \) (up to a choice of sign). The rate of convergence is cubic.
For the original GI-ICA algorithm, a somewhat complicated preprocessing step called quasi-orthogonalization was used to linearly transform the data to make columns of $A$ orthogonal. Quasi-orthogonalization makes use of evaluations of Hessians of the fourth cumulant function to construct a matrix of the form $C = ADA^T$ where $D$ has all positive diagonal entries—a task which is complicated by the possibility that the latent signals $S_i$ may have fourth order cumulants of differing signs—and requires taking the matrix square root of a positive definite matrix of this form. However, the algorithm used for constructing $C$ under sampling error is not always positive definite in practice, which can make the preprocessing step fail. We will demonstrate that a variant on GI-ICA can be performed without quasi-orthogonalization, getting rid of this issue. To distinguish the original GI-ICA algorithm from the one proposed in section 3.2, we will call the original GI-ICA algorithm qorth+GI-ICA, designating it by its preprocessing step.

### 3.2 GI-ICA in a Quasi-Inner Product Space

In this section, we demonstrate that the gradient iteration can be performed using a generalized notion of an inner product space in which the columns of $A$ are orthogonal. The natural candidate for the “inner product space” would be to use $(\langle \cdot, \cdot \rangle)_C$ defined as $(\langle u, v \rangle)_C := u^T(ADA^T)^T v$. Clearly, $(A_i, A_j)_C = \delta_{ij}$ gives the desired orthogonality property. However, there are two issues with this “inner product space”: First, it is only an inner product space when $A$ is non-singular (invertible). This turns out not to be a major issue, and we will move forward largely ignoring this point. The second issue is more fundamental: We only have access to the matrix $AA^T$ in the noise free setting where $\text{cov}(X)^T = (AA^T)^T = AA^T$. In the noisy setting, we have access to matrices of the form $Hf(u) = AD(u)A^T$ from equation (4) instead.

We consider a pseudo-inner product defined as follows: Let $C = ADA^T$ where $D$ is a diagonal matrix with non-zero diagonal entries, and define $(\langle \cdot, \cdot \rangle)_C$ by $(\langle u, v \rangle)_C = u^T C^T v$. When $D$ contains negative entries, this is not a proper inner product since $C$ is not positive definite. In particular, $(A_k, A_k)_C = A_k^T (ADA^T)^T A_k = d_{kk}^{-1}$ may be negative. Nevertheless, when $k \neq j$, $(A_k, A_j)_C = A_k^T (ADA^T)^T A_j = 0$ gives that the columns of $A$ are orthogonal in this pseudo-inner product space.

We define functions $\alpha_k : \mathbb{R}^n \rightarrow \mathbb{R}$ by $\alpha_k(u) = (A^T u)_k$ such that for any $u \in \text{span}(A_1, \ldots, A_m)$, then $u = \sum_{i=1}^{m} \alpha_i(u) A_i$ is the expansion of $u$ in its $A_i$ basis. Continuing from equation (3), for any $u \in S^{n-1}$ we see

$$
\nabla f(C^T u) = 4 \sum_{k=1}^{n} (A_k, C^T u)^3 \kappa_4(S_k) A_k \\
= 4 \sum_{k=1}^{n} (A_k, u^T C^T u_3) \kappa_4(S_k) A_k
$$

is the gradient iteration recast in the $(\langle \cdot, \cdot \rangle)_C$ space. Using the expansion of $u$ in its $A_k$ basis, we obtain

$$
\nabla f(C^T u) = 4 \sum_{k=1}^{m} (\alpha_k(u) (A_k, A_k)_C)^3 \kappa_4(S_k) A_k \\
= 4 \sum_{k=1}^{m} \alpha_k(u)^3 (d_{kk}^{-3} \kappa_4(S_k)) A_k ,
$$

which is a power iteration in the unseen $A_k$ coordinate system. As no assumptions are made upon the $\kappa_4(S_k)$ values, the $d_{kk}^{-3}$ scalings which were not present in equation (3) cause no issues. Using
We are able to recover a single column of $A$.

### 3.3 Full Non-Orthogonal GI-ICA Recovery

We are able to recover a single column of $A$ up to an unknown scaling factor when $u_0$ is generically chosen.

**Algorithm 1** Recovers a column of $A$ up to an unknown scaling factor when $u_0$ is generically chosen.

| Inputs: | $u_0$ (a unit vector), $C$, $\nabla f$ |
|---------|---------------------------------|
| $k \leftarrow 1$ |                                 |
| **repeat** |                                 |
| $u_k \leftarrow \nabla f(C^T u_{k-1})/\|\nabla f(C^T u_{k-1})\|$ |                                 |
| $k \leftarrow k + 1$ |                                 |
| **until** Convergence (up to sign) |                                 |
| **return** $u_k$ |                                 |

This update, we obtain Algorithm 1 a fixed point method for recovering a single column of $A$ up to an unknown scaling.

It is worth making a couple of remarks about Algorithm 1. First, we should clarify the notion of fixed point convergence. We say that the sequence $\{u_k\}_{k=0}^\infty$ converges to $v$ up to sign if there exists a sequence $\{c_k\}_{k=0}^\infty$ such that each $c_k \in \{\pm 1\}$ and $c_k u_k \rightarrow v$ as $k \rightarrow \infty$. We have the following convergence guarantee.

**Theorem 4.** If $u_0$ is chosen uniformly at random from $S^{n-1}$, then with probability 1, there exists $\ell \in [m]$ such that the sequence $\{u_k\}_{k=0}^\infty$ defined as in Algorithm 1 converges to a $A_\ell/\|A_\ell\|$ up to sign. Further, the rate of convergence is cubic.

Due to space limitations, we omit the proof of Theorem 5, but its proof is very similar to the proof of an analogous result for qorth+GI-ICA algorithm [24, Theorem 4].

In practice, we test near convergence by testing if we are still making significant progress. In particular, for some predefined $\epsilon > 0$, if there exists a sign value $c_k \in \{\pm 1\}$ such that $\|u_k - c_k u_{k-1}\| < \epsilon$, then we declare convergence achieved and return the result. As there are only two choices for $c_k$, this is easily checked, and we exit the loop if this condition is met.

### 3.3 Full Non-Orthogonal GI-ICA Recovery

We are able to recover a single column of $A$ up to its unknown scale. However, for full recovery of $A$, we would like (given recovered columns $A_{\ell_1}, \ldots, A_{\ell_j}$) to be able to recover a column $A_k$ such that $k \not\in \{\ell_1, \ldots, \ell_j\}$ on demand.

The main idea behind the simultaneous recovery of all columns of $A$ is two-fold. First, instead of just finding columns of $A$ using Algorithm 1, we simultaneously find rows of $A^\dagger$. Then, using the recovered columns of $A$ and rows of $A^\dagger$, we may project $u$ onto the orthogonal complement of the recovered columns of $A$ within the $(\cdot, \cdot)_C$ pseudo-inner product space.

**Recovering rows of $A^\dagger$.** Suppose we have access to a column $A_k$ (which may be achieved using Algorithm 1). Let $A^\dagger_k$ denote the $k$th row of $A^\dagger$. Then, we note that $C^T A_k = (ADA^T)^T A_k = d_{kk}^{-1}(A^T)^T_k = d_{kk}^{-1}(A_{k,})^T$ recovers $A^\dagger_k$ up to an arbitrary, unknown constant $d_{kk}^{-1}$. However, the constant $d_{kk}^{-1}$ may be recovered by noting that $\langle A_k, A_k \rangle_C = (A^\dagger_k)^T A_k = d_{kk}^{-1}$. As such, we may estimate $A^\dagger_k$ as $[C^T A_k/((C^T A_k)^T A_k)]^T$.

**Enforcing Pseudo-Orthogonality in the GI Update.** Given access to a vector $u = \sum_{k=1}^m \alpha_k(u) A_k + P_{A^\perp} u$ (where $P_{A^\perp}$ is the projection onto the orthogonal complements of the range of $A$), some recovered columns $A_{\ell_1}, \ldots, A_{\ell_j}$, and corresponding rows of $A^\dagger$, we may zero out the components of $u$ corresponding to the recovered columns of $A$. Letting $u' = u - \sum_{j=1}^r A_{\ell_j} A_{\ell_j}^\dagger u$, then
Algorithm 2 Full ICA matrix recovery algorithm. Estimates and returns two matrices: (1) $\tilde{A}$ is the recovered mixing matrix for the noisy ICA model $X = AS + \eta$, and (2) $\tilde{B}$ is a running estimate of $\tilde{A}^\dagger$.

1: Inputs: $C$, $\nabla f$
2: $\tilde{A} \leftarrow 0$, $\tilde{B} \leftarrow 0$
3: for $j \leftarrow 1$ to $m$ do
4: Draw $u$ uniformly at random from $S^{n-1}$.
5: repeat
6: $u \leftarrow u - \tilde{A} \tilde{B} u$
7: $u \leftarrow \nabla f(C^\dagger u) / \| \nabla f(C^\dagger u) \|$
8: until Convergence (up to sign)
9: $\tilde{A}_j \leftarrow u$
10: $\tilde{B}_j \leftarrow [C^\dagger A_j / ((C^\dagger A_j)^T A_j)]^T$
11: end for
12: return $\tilde{A}$, $\tilde{B}$

$u' = \sum_{k \in [m] \setminus \{\ell_1, \ldots, \ell_r\}} \alpha_k(u) A_k + P_{A\perp} u$. In particular, $u'$ is orthogonal (in the $\langle \cdot, \cdot \rangle_C$ space) to the previously recovered columns of $A$. This allows us to modify the non-orthogonal gradient iteration algorithm to recover a new column of $A$.

Using these ideas, we obtain the Algorithm 2 for recovery of the mixing matrix $A$ in noisy ICA up to the inherent ambiguities of the problem. Within this Algorithm, step 6 enforces orthogonality with previously found columns of $A$, guaranteeing that convergence is to a new column of $A$.

Practical Construction of $C$ In our implementation, we set $C = \frac{1}{12} \sum_{k=1}^n \mathcal{H} f(e_k)$, as it can be shown from equation (4) that $\sum_{k=1}^n \mathcal{H} f(e_k) = ADA^T$ with $d_{kk} = \| A_k \|^2 \kappa_4(S_k)$. This deterministically guarantees that each latent signal has a significant contribution to $C$.

4 Experimental Results

We now compare our proposed quasi-inner product GI-ICA (qip+GI-ICA) algorithm with several more established ICA algorithms. In addition to qorth+GI-ICA, we use the following algorithms as baselines:
- JADE [3] is a popular fourth cumulant based ICA algorithm designed for the noise free setting. We use the implementation of Cardoso and Souloumiac [5].
- FastICA [11] is a popular ICA algorithm designed for the noise free setting based on a deflationary approach of recovering each component one at a time. We use the implementation of Gävert et al. [8].
- 1FICA [15, 16] is a variation of FastICA with the tanh contrast function designed to have low bias for performing SINR-optimal beamforming in the presence of Gaussian noise.
- Ainv is the oracle demixing algorithm which uses $A^\dagger$ as the demixing matrix.
- SINR-opt is the oracle demixing algorithm which demixes using $A^H \text{cov}(X)^{\dagger}$ to achieve an SINR-optimal demixing.

We compare these algorithms on real simulated data with $n = m$. In particular, we constructed mixing matrices $A$ with condition number 3 via a reverse singular value decomposition ($A = U \Lambda V^T$). The matrices $U$ and $V$ were random orthogonal matrices, and $\Lambda$ was chosen to have 1 as its minimum singular and 3 as its maximum singular value, with the intermediate singular values...
chosen uniformly at random. We drew data from a noisy ICA model $X = AS + \eta$ where $\text{cov}(\eta) = \Sigma$ was chosen to be malaligned with $\text{cov}(AS) = AA^T$. In particular, we set $\Sigma = p(10I - AA^T)$ where $p$ is a constant defining the noise power. It can be shown that $p = \frac{\max_v \text{var}(v^T \eta)}{\max_v \text{var}(v^T AS)}$ is the ratio of the maximum directional noise variance to the maximum directional signal variance.

We generated 100 matrices $A$ for our experiments with 100 corresponding ICA data sets for each sample size and noise power. In particular, when reporting results, we apply each algorithm to each of the 100 data sets for the corresponding sample size and noise power, we report the mean performance of that algorithm with error bars corresponding to a 95% confidence interval on the mean. The source distributions used in our ICA experiments were the Laplace distribution, the Bernoulli distribution with parameters 0.05 and 0.5 respectively, the t distribution with 3 and 5 degrees of freedom respectively, the exponential distribution, and the uniform distribution. Each distribution was normalized to have unit variance, and the distributions were each used twice to create 14-dimensional data.

We compare the algorithms using either SINR or the SINR loss from the optimal demixing matrix (defined by SINR Loss = [Optimal SINR – Achieved SINR]).

In Figure 1 we compare our proposed ICA algorithm with various ICA algorithms for signal recovery. In the qip+GI-ICA-$\kappa_4$+SINR algorithm, we use qip+GI-ICA-$\kappa_4$ to estimate $A$, and then perform demixing using the resulting estimate of $A^H \text{cov}(X)^{-1}$, the formula for SINR-optimal demixing. It is apparent that when given sufficient samples, qip+GI-ICA-$\kappa_4$+SINR provides the best SINR demixing. JADE, FastICA-tanh, and 1FICA each have a bias in the presence of additive Gaussian noise which keeps them from being SINR-optimal even when given many samples.

In Figure 2 we see how the various algorithms compare at various sample sizes. The qip+GI-ICA-$\kappa_4$+SINR algorithm relies more heavily on accurate estimates of fourth order statistics than JADE, and the FastICA-tanh and 1FICA algorithms do not require the estimation of fourth order
statistics. For this reason, qip+GI-ICA-κ₄+SINR requires more samples than the other algorithms in order to be run accurately. However, once sufficient samples are taken, qip+GI-ICA-κ₄+SINR outperforms the other algorithms including 1FICA which is designed to have low SINR bias.

In order to avoid clutter, we did not include qorth+GI-ICA-κ₄+SINR (the SINR optimal demixing estimate constructed using qorth+GI-ICA-κ₄ to estimate A) in the figures 1 and 2. It is also asymptotically unbiased in estimating the directions of the columns of A, and similar conclusions could be drawn using qorth+GI-ICA-κ₄ in place of qip+GI-ICA-κ₄. However, in Figure 3 we see that qip+GI-ICA-κ₄+SINR requires fewer samples than qorth+GI-ICA-κ₄+SINR to achieve good performance. This is particularly highlighted in the medium sample regime.

**On the Decent Performance of Traditional ICA Algorithms for Noisy ICA.** An interesting observation [previously made by 14] is that the popular noise free ICA algorithms JADE and FastICA actually perform reasonably well in the noisy setting. In particular in Figures 1 and 2 they significantly outperform demixing using A⁻¹ for source recovery, and they appear to be doing something in between a demixing using A⁻¹ and the SINR-optimal solution. It turns out that this may be explained by a shared preprocessing step.

Both JADE and FastICA rely on a whitening preprocessing step in which the data are linearly transformed to have identity covariance. It can be shown in the noise free setting that after whitening, the mixing matrix A is a rotation matrix. These algorithms proceed by recovering an orthogonal matrix ˜A to approximate the true mixing matrix A. Demixing is performed using ˜A⁻¹ = ˜Aᴴ. Since the data is white (has identity covariance), then the demixing matrix ˜Aᴴ = ˜Aᴴ cov(X)⁻¹ is an estimate of the SINR-optimal demixing matrix.

Nevertheless, the traditional ICA algorithms give a biased estimate of A under additive Gaussian noise. This is unsurprising seeing as the algorithmic assumptions are violated. The additive Gaussian noise corrupts the covariance information, making it so that the columns of A are not
actually orthogonal after the covariance-based whitening operation.

References

[1] L. Albera, A. Ferréol, P. Comon, and P. Chevalier. Blind identification of overcomplete mixtures of sources (BIOME). *Linear algebra and its applications*, 391:3–30, 2004.

[2] S. Arora, R. Ge, A. Moitra, and S. Sachdeva. Provable ICA with unknown Gaussian noise, with implications for Gaussian mixtures and autoencoders. In *NIPS*, pages 2384–2392, 2012.

[3] J. Cardoso and A. Souloumiac. Blind beamforming for non-Gaussian signals. In *Radar and Signal Processing, IEE Proceedings F*, volume 140(6), pages 362–370. IET, 1993.

[4] J.-F. Cardoso. Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 3109–3112. IEEE, 1991.

[5] J.-F. Cardoso and A. Souloumiac. Matlab JADE for real-valued data v 1.8. [http://perso.telecom-paristech.fr/~cardoso/Algo/Jade/jadeR.m](http://perso.telecom-paristech.fr/~cardoso/Algo/Jade/jadeR.m) 2005. [Online; accessed 8-May-2013].

[6] P. Comon and C. Jutten, editors. *Handbook of Blind Source Separation*. Academic Press, 2010.

[7] L. De Lathauwer, B. De Moor, and J. Vandewalle. Independent component analysis based on higher-order statistics only. In *Statistical Signal and Array Processing, 1996. Proceedings., 8th IEEE Signal Processing Workshop on*, pages 356–359. IEEE, 1996.
[8] H. Gävert, J. Hurri, J. Särelä, and A. Hyvärinen. Matlab FastICA v 2.5. [http://research.ics.aalto.fi/ica/fastica/code/dlcode.shtml](http://research.ics.aalto.fi/ica/fastica/code/dlcode.shtml) 2005. [Online; accessed 1-May-2013].

[9] E. Gonen and J. M. Mendel. Applications of cumulants to array processing, part vi. polarization and direction of arrival estimation with minimally constrained arrays. *Signal Processing, IEEE Transactions on*, 47(9):2589–2592, 1999.

[10] N. Goyal, S. Vempala, and Y. Xiao. Fourier PCA. *CoRR*, abs/1306.5825, 2013.

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

[12] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*. John Wiley & Sons, 2001.

[13] M. Joho, H. Mathis, and R. H. Lambert. Overdetermined blind source separation: Using more sensors than source signals in a noisy mixture. In *Proc. International Conference on Independent Component Analysis and Blind Signal Separation. Helsinki, Finland*, pages 81–86, 2000.

[14] Z. Koldovský and P. Tichavský. Methods of fair comparison of performance of linear ica techniques in presence of additive noise. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, pages V–V. IEEE, 2006.

[15] Z. Koldovský and P. Tichavský. Asymptotic analysis of bias of fastica-based algorithms in presence of additive noise. Technical report, Technical report, 2007.

[16] Z. Koldovský and P. Tichavský. Blind instantaneous noisy mixture separation with best interference-plus-noise rejection. In *Independent Component Analysis and Signal Separation*, pages 730–737. Springer, 2007.

[17] S. Makino, T.-W. Lee, and H. Sawada. *Blind speech separation*. Springer, 2007.

[18] B. D. Van Veen and K. M. Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine*, 5(2):4–24, 1988.

[19] M. Viberg and B. Ottersten. Sensor array processing based on subspace fitting. *Signal Processing, IEEE Transactions on*, 39(5):1110–1121, 1991.

[20] R. Vigário, J. Sarela, V. Jousmiki, M. Hamalainen, and E. Oja. Independent component approach to the analysis of EEG and MEG recordings. *Biomedical Engineering, IEEE Transactions on*, 47(5):589–593, 2000.

[21] J. R. Voss, L. Rademacher, and M. Belkin. Fast algorithms for Gaussian noise invariant independent component analysis. In *Advances in Neural Information Processing Systems 26*, pages 2544–2552. 2013.

[22] A. Yeredor. Blind source separation via the second characteristic function. *Signal Processing*, 80(5):897–902, 2000.

[23] A. Yeredor. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *Signal Processing, IEEE Transactions on*, 50(7):1545–1553, 2002.
A Qip+GI-ICA for Complex Signals

In Section 3 we showed how to perform gradient iteration ICA within an (possibly not positive definite) inner product space. In this appendix, we show how this qip+GI-ICA algorithm can be extended to include complex valued signals. For clarity, we repeat the entire qip+GI-ICA algorithmic construction from Section 3 with the necessary modifications to handle the complex setting.

Throughout this appendix, we assume that $m \leq n$, and that the columns of $A$ are linearly dependent.

A.1 Fourth Cumulants of Complex Variables

The gradient iteration relies on the properties of cumulants. We will focus on the fourth cumulant, though similar constructions may be given using other even order cumulants of higher order. We will use two versions of the fourth cumulant which capture slightly different fourth order information. For a zero-mean random variable $X$, they may be defined as $\kappa_4(X) := \mathbb{E}[X^4] - 3\mathbb{E}[X^2]^2$ and $\kappa_4^*(X) := \mathbb{E}[X^2X^*]^2 - 2\mathbb{E}[XX^*]^2 - \mathbb{E}[X^2]\mathbb{E}[X^*]^2$. For real random variables, these two definitions are equivalent, and they come from two different conjugation schemes when constructing the fourth order cumulant [see [6], Chapter 5, Section 1.2]. However, in general, only $\kappa_4^*$ is guaranteed to be real valued. The higher order cumulants have nice algebraic properties which make them useful for ICA:

1. (Independence) If $X$ and $Y$ are independent random variables, then $\kappa_4(X + Y) = \kappa_4(X) + \kappa_4(Y)$ and $\kappa_4^*(X + Y) = \kappa_4(X + Y)$.

2. (Homogeneity) If $\alpha$ is a scalar, then $\kappa_4(\alpha X) = \alpha^4 \kappa_4(X)$ and $\kappa_4^*(\alpha X) = |\alpha|^4 \kappa_4^*(X)$.

3. (Vanishing Gaussians) If $X$ is normally distributed then $\kappa_4(X) = 0$ and $\kappa_4^*(X) = 0$.

In this appendix, we consider a noisy ICA model $\mathbf{X} = \mathbf{AS} + \mathbf{\eta}$ where $\mathbf{\eta}$ is a 0-mean (possibly complex) Gaussian and independent of $\mathbf{S}$. We consider the following functions defined on the unit sphere: $f(\mathbf{u}) := \kappa_4(\langle \mathbf{X}, \mathbf{u} \rangle)$ and $f_*(\mathbf{u}) := \kappa_4^*(\langle \mathbf{X}, \mathbf{u} \rangle)$. Then, expanding using the above properties we obtain:

$$f(\mathbf{u}) = \kappa_4 \left( \sum_{k=1}^{m} \langle A_k, \mathbf{u} \rangle S_k + \langle \mathbf{u}, \mathbf{\eta} \rangle \right)$$

$$= \sum_{k=1}^{m} \langle A_k, \mathbf{u} \rangle^4 \kappa_4(S_k)$$

Using similar reasoning, it can be seen that $f_*(\mathbf{u}) = \sum_{k=1}^{m} |\langle A_k, \mathbf{u} \rangle|^4 \kappa_4^*(S_k)$.

It turns out that some slightly non-standard notions of derivatives are most useful in constructing the gradient iteration in the complex setting. We use real derivatives for the gradient and we use the complex Hessian. In particular, expanding $u_k = x_k + iy_k$, we use the gradient operator $\nabla := \sum_{k=1}^{n} e_k \frac{\partial}{\partial x_k}$. We make use of the operators $\partial u_k := \frac{1}{2}(\partial \partial_{x_k} - i \partial \partial_{y_k})$ and $\partial u_k^* := \frac{1}{2}(\partial \partial_{x_k} + i \partial \partial_{y_k})$ to define $\mathcal{H} := \sum_{j=1}^{n} \sum_{k=1}^{n} e^j_k \partial u_k \partial u_k^*$. Applying this version of the Hessian is different than using real derivatives as in the gradient operation.

Taking derivatives, we obtain:

$$\nabla f(\mathbf{u}) = 4 \sum_{k=1}^{m} \langle A_k, \mathbf{u} \rangle^3 \kappa_4(S_k) A_k$$

(6)
where $D(u)$ is a diagonal matrix with entries $D(u)_{kk} = 4|\langle A_k, u \rangle|^2 \kappa_4^*(S_k)$.

## A.2 GI-ICA in a Quasi-Inner Product Space

In this section, we demonstrate that the gradient iteration can be performed using a generalized notion of an inner product space in which the columns of $A$ are orthogonal. The natural candidate for the “inner product space” would be to use $\langle \cdot, \cdot \rangle_A$ defined as $\langle u, v \rangle_A := u^T(A^*A^T)\dagger v^*$. Clearly, $\langle A_i, A_j \rangle_A = \delta_{ij}$ gives the desired orthogonality property. However, there are two issues with this “inner product space”: First, it is only an inner product space when $A$ is non-singular (invertible). This turns out not to be a major issue, and we will move forward largely ignoring this point. The second issue is more fundamental: We only have access to the matrix $A$ with non-zero diagonal entries, and define $\langle \cdot, \cdot \rangle_C := u^T C^\dagger v^*$. When $C$ contains negative entries, this is not a proper inner product since $\langle u, u \rangle_C < 0$ when $C$ is non-positive definite. In particular, $\langle A_k, A_k \rangle_C = A_k^T (A^*A^T)^\dagger A_k^* = d_{kk}^{-1}$ may be negative. Nevertheless, when $k \neq j$, $\langle A_k, A_j \rangle_C = A_k^T (A^*A^T)^\dagger A_j^* = 0$ gives that the columns of $A$ are orthogonal in this pseudo-inner product space.

We define functions $\alpha_k : \mathbb{C}^n \to \mathbb{C}$ by $\alpha_k(u) := (A^\dagger u)_k$ such that for any $u \in \text{span}(A_1, \ldots, A_m)$, then $u = \sum_{i=1}^m \alpha_i(u)A_i$ is the expansion of $u$ in its $A_i$ basis. Continuing from equation (6), for any $u \in S^{n-1}$ we see

$$
\nabla f(C^\dagger u) = 4 \sum_{k=1}^n \langle A_k, C^\dagger u \rangle^3 \kappa_4(S_k) A_k 
= 4 \sum_{k=1}^n \langle A_k, u \rangle^3 \kappa_4(S_k) A_k 
$$

is the gradient iteration recast in the $\langle \cdot, \cdot \rangle_C$ space. Using the expansion of $u$ in its $A_k$ basis, we obtain

$$
\nabla f(C^\dagger u) = \sum_{k=1}^m (\alpha_k(u) \langle A_k, A_k \rangle_C^3 \kappa_4(S_k)) A_k 
= \sum_{k=1}^m \alpha_k(u)^3 (d_{kk}^{-3} \kappa_4(S_k)) A_k ,
$$

which is a power iteration in the unseen $A_k$ coordinate system. As no assumptions are made upon the $\kappa_4(S_k)$ values, the $d_{kk}^{-3}$ scalings which were not present in equation (6) cause no issues. Using this update, we obtain Algorithm 3, a fixed point method for recovering a single column of $A$ up to an unknown scaling.

It is worth making a couple of remarks about about Algorithm 3. First, we should clarify the notion of fixed point convergence. We say that the sequence $\{u_k\}_{k=0}^\infty$ converges to $v$ up to a unit modulus factor if there exists a sequence of constants $\{c_k\}_{k=0}^\infty$ such that each $|c_k| = 1$ and $c_k u_k \to v$ as $k \to \infty$. We have the following convergence guarantee.
Algorithm 3 Recovers a column of $A$ up to an unknown scaling factor when $u_0$ is generically chosen.

**Inputs:** $u_0$ (a unit vector), $C$, $\nabla f$

$k \leftarrow 1$

repeat

$u_k \leftarrow \nabla f(C^\dagger u_{k-1})/\|\nabla f(C^\dagger u_{k-1})\|$

$k \leftarrow k + 1$

until Convergence (up to a unit modulus factor)

return $u_k$

**Theorem 5.** If $u_0$ is chosen uniformly at random from $S^{n-1}$. Then with probability 1, there exists $\ell \in [m]$ such that the sequence ${u_k}_{k=0}^\infty$ defined as in Algorithm 3 converges to a $A_\ell/\|A_\ell\|$ up to a unit modulus factor. Further, the rate of convergence is cubic.

Due to space limitations, we omit the proof of Theorem 5. However, its proof is very similar to that of an analogous result for qorth+GI-ICA algorithm [21, Theorem 4].

**Fact 6.** Suppose that $\mathbf{u}$ and $\mathbf{v}$ are non-orthogonal unit modulus vectors. The expression $\|\mathbf{u} - e^{i\theta}\mathbf{v}\|$ is minimized by the choice of $\theta = \text{atan2}(\text{Im}(\langle \mathbf{u}, \mathbf{v}\rangle), \text{Re}(\langle \mathbf{u}, \mathbf{v}\rangle))$.

Letting $\theta = \text{atan2}(\text{Im}(\langle \mathbf{u}_k, \mathbf{u}_{k-1}\rangle), \text{Re}(\langle \mathbf{u}_k, \mathbf{u}_{k-1}\rangle))$, we exit the loop if $\|\mathbf{u}_k - e^{i\theta}\mathbf{u}_{k-1}\| < \epsilon$.

### A.3 Full Non-Orthogonal GI-ICA Recovery

We are able to recover a single column of $A$ in noisy ICA. However, for full matrix recovery, we would like (given recovered columns $A_{\ell_1}, \ldots, A_{\ell_\ell}$) to be able to recover a column $A_k$ such that $k \not\in \{\ell_1, \ldots, \ell_\ell\}$ on demand.

The main idea behind the simultaneous recovery of all columns of $A$ is two-fold. First, instead of just finding columns of $A$ using Algorithm 3, we simultaneously find rows of $A^\dagger$. Then, using the recovered columns of $A$ and rows of $A^\dagger$, we may project $\mathbf{u}$ onto the orthogonal complement of the recovered columns of $A$ within the $(\cdot, \cdot)_C$ pseudo-inner product space.

**Recovering rows of $A^\dagger$.** Suppose we have access to a column $A_k$ (which may be achieved using Algorithm 3). Let $A^\dagger_{k,\ell}$ denote the $k^{th}$ row of $A^\dagger$. Then, we note that $C^\dagger A^\star_k = (A^\star D A^T)^\dagger A^\star_k = d^{-1}_k(A^T)_{k,\ell} = d^{-1}_k(A^\dagger_{k,\ell})^T$ recovers $A^\dagger_{k,\ell}$ up to an arbitrary, unknown constant $d^{-1}_k$. However, the constant $d^{-1}_k$ may be recovered by noting that $(A_k, A_k)_C = (C^\dagger A_k)^T A_k = d^{-1}_k$. As such, we may estimate $A^\dagger_{k,\ell}$ as $[C^\dagger A_k / ((C^\dagger A_k)^T A_k)]^T$.

**Enforcing Pseudo-Orthogonality in the GI Update.** Given access to $\mathbf{u} = \sum_{k=1}^m \alpha_k(\mathbf{u})A_k + P_{A^\perp}\mathbf{u}$, some recovered columns $A_{\ell_1}, \ldots, A_{\ell_\ell}$, and corresponding rows of $A^\dagger$, we may zero out the components of $\mathbf{u}$ corresponding to the recovered columns of $A$. Letting $\mathbf{u}' = \mathbf{u} - \sum_{j=1}^r A_{\ell_j}A_{\ell_j}^\dagger \mathbf{u}$, then $\mathbf{u}' = \sum_{k \in [m] \setminus \{\ell_1, \ldots, \ell_\ell\}} \alpha_k(\mathbf{u})A_k + P_{A^\perp}\mathbf{u}$. In particular, $\mathbf{u}'$ is orthogonal (in the $(\cdot, \cdot)_C$ space) to the previously recovered columns of $A$. This allows us to modify the non-orthogonal gradient iteration algorithm to recover a new column of $A$.  

17
Algorithm 4 Full ICA matrix recovery algorithm. Estimates and returns two matrices: (1) \( \hat{A} \) is the recovered mixing matrix for the noisy ICA model \( \mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{\eta} \), and (2) \( \hat{B} \) is a running estimate of \( \hat{A}^\dagger \).

```
1: Inputs: \( C, \nabla f \)
2: \( \hat{A} \leftarrow 0, \hat{B} \leftarrow 0 \)
3: for \( j \leftarrow 1 \) to \( m \) do
4: \hspace{1em} Draw \( \mathbf{u} \) uniformly at random from \( S^{n-1} \).
5: \hspace{2em} repeat
6: \hspace{3em} \( \mathbf{u} \leftarrow \mathbf{u} - \hat{A}\hat{B}\mathbf{u} \)
7: \hspace{3em} \( \mathbf{u} \leftarrow \nabla f(C^\dagger\mathbf{u})/\|\nabla f(C^\dagger\mathbf{u})\| \).
8: \hspace{2em} until Convergence (up to a unit modulus factor)
9: \( \hat{A}_j \leftarrow \mathbf{u} \)
10: \( \hat{B}_j \leftarrow [C^\dagger A_j/((C^\dagger A_j)^T A_j)]^T \)
11: end for
12: return \( \hat{A}, \hat{B} \)
```

Using these ideas, we obtain the Algorithm 4 for recovery of the ICA mixing matrix. Within this Algorithm, step 6 enforces orthogonality with previously found columns of \( A \), guaranteeing that convergence is to a new column of \( A \).

**Practical Construction of \( C \)** We suggest the choice of \( \mathbf{C} = \frac{1}{4} \sum_{k=1}^{n} \mathcal{H}_f(e_k) \), as it can be shown from equation (7) that \( \sum_{k=1}^{n} \mathcal{H}_f(e_k) = \mathbf{A}^\ast \mathbf{D} \mathbf{A}^T \) with \( d_{kk} = \|A_k\|^2 \kappa_k^2(S_k) \). This deterministically guarantees that each latent signal has a significant contribution to \( C \).