1 Appendix

1.1 Pseudo-code for Computation of Differential Form for GT-PPCA

We detail the algorithm for computing the differential form correction necessary to sample on angles that correspond to orthonormal matrices on the Stiefel Manifold below. We omit the finer details for the functions *GenerateCumulativeForwardRotations*, *GenerateCumulativeReverseRotations*, and *GenerateGivensJacobians*. *GenerateCumulativeForwardRotations* and *GenerateCumulativeReverseRotations* computes the quantity

$$(R_{pn}^{-1} \cdots R_{p,p+2}^{-1} R_{p,p+1}^{-1}) \cdots (R_{2n}^{-1} \cdots R_{24}^{-1} R_{23}^{-1})(R_{1n}^{-1} \cdots R_{13}^{-1} R_{12}^{-1}) Y \tag{1}$$

from the left and from the right, saving the cumulative products at each step. *GenerateGivensJacobians* computes the Jacobian using these saved partial products noting that the Jacobian for the i, j-th entry is simply replacing the i, j-th rotation matrix in (1) by the derivative of the i, j-th matrix.

For a complete implementation in Stan, we also include the complete code for the computation as Supplementary Material.

Algorithm 1 Given's Differential Form

Input:

θ, *vector of angles*
n, *first dimension of matrix*
p, *second dimension of matrix*

Output:

LD, *value of differential form to add to the log probability*

function GIVENSDIFFERENTIALFORM(θ, n, p)
  $d = np - \frac{p(p+1)}{2}$
  $AreaMatrix \leftarrow Identity(d)$
  $GF \leftarrow GenerateCumulativeForwardRotations(\theta)$
  $GR \leftarrow GenerateCumulativeReverseRotations(\theta)$
  $GJ \leftarrow GenerateGivensJacobians(GF, GR, \theta)$
  $Givens \leftarrow GF[d]$
  $idx \leftarrow 0$
  for i = 0 to p **do**
    $OneForms \leftarrow (Givens[i+1:n,:]^T * GJ[i])^T$
    for j = 0 to n **do**
      $AreaMatrix[:, idx] = OneForms[:, j]$
      $idx = idx + 1$
    end for
  end for
  $LD \leftarrow log(det(AreaMatrix))$
  return LD
end function