Supplementary Materials

1. Proof of Proposition 3.1

Without loss of generality, we may assume $\|D_i\| = 1$ for all $i$. From the definition of $\Phi$, we know
$$\|\Phi(D_i)\|_2^2 = \langle \Phi(D_i), \Phi(D_i) \rangle = \psi(0), \forall i,$$
and
$$\langle \Phi(D_i), \Phi(D_j) \rangle = \psi(2-2\mu_0), \forall i \neq j.$$  
We complete the proof by noting $c_0 = \sqrt{\psi(0)}$ and $\eta = \psi(2-2\mu_0)$.

2. Proof of Proposition 3.5

Since $H(C, D) = \frac{1}{2} \text{Tr}(C^\top QC - 2K(D, Y)^\top C)$ and $k(x, y) = \exp(-\|x - y\|^2/2\sigma^2)$, we have
$$\nabla_C H(C, D) = QC - K(D, Y),$$
$$\nabla_{D_i} H(C, D) = \sum_{i=1}^n a_{\ell_i} (D_\ell - Y_i), \forall \ell,$$
where $a_{\ell_i} = -\frac{1}{\sigma^2} C_{\ell_i} \exp\left(-\frac{\|D_\ell - Y_i\|^2}{2\sigma^2}\right)$.

As $\nabla_C^2 H(C, D) = Q$ implies that $\nabla_C H(C, D)$ is Lipschitz with modulus $\lambda_{\text{max}}(Q)$, where $\lambda_{\text{max}}(Q)$ is the maximal eigenvalue of $Q$. Moreover, the Hessian matrix $\nabla_{D_i}^2 H(C, D)$ is given by
$$\sum_{i=1}^n a_{\ell_i} \left(I - \frac{1}{\sigma^2} (D_\ell - Y_i)(D_\ell - Y_i)^\top\right).$$
By the fact $(1 - \|y\|^2/\sigma^2)^2 \leq \|d - y\|^2 \leq (1 + \|y\|^2/\sigma^2)^2$ for any $\|d\|_2 = 1$, we have $|a_{\ell_i}| \leq \frac{1}{\sigma^2} |C_{\ell_i}| \exp\left(-\frac{(1 + \|Y_i\|^2/\sigma^2)}{2\sigma^2}\right)$ and the maximal eigenvalue is bounded by $1 + \frac{1}{\sigma^2} \|D_\ell - Y_i\|^2 \leq 1 + \frac{1}{\sigma^2} (1 + \|Y_i\|^2/\sigma^2)^2$. Thus, the maximal eigenvalue of $\nabla_{D_i}^2 H(C, D)$ is bounded by $L(C_\ell)$ which is defined as
$$\sum_{i=1}^n \frac{1}{\sigma^2} |C_{\ell_i}| \exp\left(-\frac{1 + \|Y_i\|^2/\sigma^2}{2\sigma^2}\right)(1 + \frac{1}{\sigma^2}(1 + \|Y_i\|^2/\sigma^2)^2).$$

3. Numerical Algorithm for The Supervised Equiangular Kernel Sparse Coding Problem (16)

Recall that the supervised extension of our equiangular kernel dictionary learning method is formulated as the following minimization model:
$$\min_{D \in D, C \in C, W} \frac{1}{2} \text{Tr}(C^\top QC - 2K(D, Y)^\top C) + \frac{\mu}{2} \|L - WC\|_F^2 + \frac{\nu}{2} \|W\|_F^2,$$
where $C = \{C : \|C\|_\infty \leq M, \|C_i\|_0 \leq T, \forall i\}$ and $D = \{D : D^\top D = DD^\top = I\}$. We give the detailed algorithm for solving (3) as follows. Define
$$H(C, D, W) = \frac{1}{2} \text{Tr}(C^\top QC - 2K(D, Y)^\top C) + \frac{\mu}{2} \|L - WC\|_F^2,$$
$$F(C) = \delta_C(C), G(D) = \delta_D(C), E(W) = \frac{\nu}{2} \|W\|_F^2.$$  
Then the sparse code $C$, dictionary $D$ and classifier $W$ are updated by the following proximal alternating scheme.

1. **Kernel sparse coding.** When the dictionary $D$ and the classifier $W$ are fixed, we update the sparse code $C$ via solving:
$$C^{j+1} \in \text{argmin}_C F(C) + \frac{\mu}{2} \|C - U^j\|_F^2,$$
where $U^j = C^j - \nabla_C H(C^j, D^j, W^j)/s^j$ and $s^j$ is some positive step size. This subproblem has a closed-form solution given by
$$C^{j+1} = \text{sign}(U^j) \odot \text{argmin}_C \|H_T(\|U^j\|)\|_1,$$
where $H_T(X)$ keeps the largest $T$ entries in each column of $X$ and sets others to zero.

2. **Dictionary update.** When the sparse code $C$ and the classifier $W$ are fixed, the update of dictionary $D$ is the same as that in the unsupervised version, i.e. we update the dictionary $D$ by solving
$$D^{j+1} \in \text{argmin}_D G(D) + \frac{\nu}{2} \|D - V^j\|_F^2,$$
where $V^j = D^j - \nabla_D H(C^{j+1}, D^j, W^j)/t^j$ and $t^j$ is some positive step size. This problem (6) has a closed-form solution given by the Proposition 3.4 in our paper.

3. **Classifier update.** When the dictionary $D$ and sparse code $C$ are fixed, we update $W$ via solving the following minimization:
$$\text{argmin}_W \frac{\mu}{2} \|L - WC\|_F^2 + \frac{\nu}{2} \|W\|_F^2 + \frac{\nu'}{2} \|W - W^j\|_F^2,$$
where \( p^j > 0 \). The solution of (7) is given by

\[
W^{j+1} = (\beta L C^j + p^j W^j) (\beta C^j C^j + (\alpha + p^j I))^{-1}
\]  

(8)

**Setting of step size.** The three step sizes \( p^j, s^j, t^j \) are set as follows. Since \( \| W^j \|_F^2 \) has coercive property, we know \( W^j \) is a bounded sequence and the maximal eigenvalue of \( Q + W^j W^j \) is defined by \( \lambda_{\text{max}}^j \) and \( \lambda_{\text{max}} = \max_j \lambda_{\text{max}}^j \).

Given \( \gamma_j > 0, 0 < a < b \) and \( 0 < c < d \) such that \( b > \lambda_{\text{max}}, d > L_{\text{max}}, \) where \( L_{\text{max}} = \max \{ L(C_\ell) : \ell = 1, 2, \ldots, m, C \in C \} \) and \( L(C_\ell) \) is defined in (2).

\[
s^j = \max(\min(\gamma_j \lambda_{\text{max}}, b), a), \quad (9a)
\]

\[
t^j = \max(\min(\gamma_j L(C^{j+1}), d), c), \quad (9b)
\]

\[
p^j \in [p_{\min}, p_{\max}], \quad (9c)
\]

where \( L(C^{j+1}) = \max \{ L(C^{j+1}_\ell), \ell = 1, 2, \ldots, m \} \) and \( p_{\min}, p_{\max} \) are two positive numbers.

**Convergence analysis.** We can easily extend the convergence result of Alg. 1 to the supervised version by checking the conditions in the proof of Theorem 3.7. The proof is omitted here.

4. **Algorithm for Solving Problem (17)**

The minimization problem (17) is equivalent to

\[
\min_X \text{Tr}(X^\top A X - B^\top X),
\]

subject to \( \| X \|_0 \leq T \), where \( A = K(D, D) \) and \( B = K(D, Y) \). We use proximal gradient descent method to solve (10). More specifically, we update \( X \) via

\[
X^{j+1} = \text{sign}(\hat{X}^j) \odot H_T(|\hat{X}^j|),
\]

where \( \hat{X}^j = X^j - (AX^j - B)/v \) and \( H_T \) is defined in (5). The step size \( v \) is set as \( v = \lambda(A) \) where \( \lambda(A) \) is the maximal eigenvalue of \( A \).

5. **Details of The Global Feature Extraction**

Given the sparse code \( C \in \mathbb{R}^{m \times n \times l \times k} \) of a DT sequence \( g \in \mathbb{R}^{m \times n \times l} \), we use \( C(i) = C(:, :, :, i) \in \mathbb{R}^{m \times n \times l} \) to denote the sparse code that corresponds to the \( i \)th dictionary atom \( D_i \). As the sparse code is extracted by a sliding window, \( C(i) \) can be viewed as a sequence whose size is the same as the original DT sequence. Then, we extract a histogram \( h^A_{(i)} \in \mathbb{R}^{l_0 \times 1} \) on \( C(i) \) w.r.t. code value. Moreover, we extract three mean histograms along \( X, Y, \) and \( T \) axes, which are denoted by \( h^X_{(i)}, h^Y_{(i)}, h^T_{(i)} \in \mathbb{R}^{l_0 \times 1} \) respectively. Take the X-axis case for example. We cut \( C(i) \) into slices along the X axis, and compute a histogram w.r.t. code value on each slice. These histograms are averaged to be the mean histogram for the X axis. See Fig. 1 for an illustration of such a process. Define \( h_{(i)} = [h^A_{(i)}, h^X_{(i)}, h^Y_{(i)}, h^T_{(i)}] \). The final feature vector for \( g \) is the concatenation of \( h_{(i)} \) over \( i \).