Assume **w** is a DFT matrix with transform length N = 3, and its Gram $\mathbf{w}^*\mathbf{w}$:

$$\mathbf{W} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} \\ \mathbf{W}^* \mathbf{W} = \begin{pmatrix} a_{1,1}^2 + a_{2,1}^2 + a_{3,1}^2 & a_{1,1}a_{1,2} + a_{2,1}a_{2,2} + a_{3,1}a_{3,2} & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} \\ a_{1,1}a_{1,2} + a_{2,1}a_{2,2} + a_{3,1}a_{3,2} & a_{1,2}^2 + a_{2,2}^2 + a_{3,2}^2 & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} \\ a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} & a_{1,3}^2 + a_{2,2}^2 + a_{3,2}^2 \end{pmatrix}$$

Assume hop L=2, construct frame operator Φ :

Main body of Gram $\Phi^*\Phi$:

$$\boldsymbol{\Phi}^*\boldsymbol{\Phi} = \begin{pmatrix} a_{1,2}^{\ 2} + a_{2,2}^{\ 2} + a_{3,2}^{\ 2} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} & 0 & 0 & 0 & 0 \\ a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} & a_{1,1}^{\ 2} + a_{1,3}^{\ 2} + a_{2,1}^{\ 2} + a_{2,3}^{\ 2} + a_{3,1}^{\ 2} + a_{2,2}^{\ 2} + a_{2,1}a_{2,2} + a_{3,1}a_{3,2} & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & 0 & 0 \\ 0 & a_{1,1}a_{1,2} + a_{2,1}a_{2,2} + a_{3,1}a_{3,2} & a_{1,2}^{\ 2} + a_{2,2}^{\ 2} + a_{3,2}^{\ 2} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} & 0 & 0 \\ 0 & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & a_{1,2}^{\ 2} + a_{2,2}^{\ 2} + a_{3,2}^{\ 2} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,1}^{\ 2} + a_{3,1}^{\ 2}$$

An alternate method to construct Gram $\Phi^*\Phi$ without performing costly matrix multiplication:

Step 1 of overlap-add $\mathbf{W}^*\mathbf{W}$ along diagonal, main body of Gram $\mathbf{\Phi}^*\mathbf{\Phi}$:

Step 2 of overlap-add $\mathbf{W}^*\mathbf{W}$ along diagonal, main body of Gram $\mathbf{\Phi}^*\mathbf{\Phi}$:

Step 3 of overlap-add $\mathbf{W}^*\mathbf{W}$ along diagonal, main body of Gram $\mathbf{\Phi}^*\mathbf{\Phi}$:

$$\begin{pmatrix} a_{1,2}^2 + a_{2,2}^2 + a_{3,2}^2 & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} & 0 & 0 & 0 & 0 \\ a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} & a_{1,1}^2 + a_{1,3}^2 + a_{2,1}^2 + a_{2,3}^2 + a_{3,1}^2 + a_{3,3}^2 & a_{1,1}a_{1,2} + a_{2,1}a_{2,2} + a_{3,1}a_{3,2} & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & 0 & 0 \\ 0 & a_{1,1}a_{1,2} + a_{2,1}a_{2,2} + a_{3,1}a_{3,2} & a_{1,2}^2 + a_{2,2}^2 + a_{3,2}^2 & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} & 0 & 0 \\ 0 & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & a_{1,2}^2 + a_{2,2}^2 + a_{3,2}^2 & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}^2 + a_{3,1}^2 + a_{3,1}^2 + a_{3,1}^2 + a_{3,1}^2 + a_{3,1}^2 + a_{3,1}^2 + a_{2,1}^2 + a_{2,1}^2 + a_{2,1}^2 + a_{2,1}^2 + a_{2,1}^2 + a_{2,1}^2 + a_{2,2}^2 + a_{3,2}^2 & a_{1,2}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} \\ 0 & 0 & a_{1,1}a_{1,2} + a_{2,1}a_{2,2} + a_{3,1}a_{3,2} & a_{1,2}^2 + a_{2,2}^2 + a_{3,2}^2 & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} \\ 0 & 0 & a_{1,1}a_{1,2} + a_{2,1}a_{2,2} + a_{3,1}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} \\ 0 & 0 & a_{1,1}a_{1,2} + a_{2,1}a_{2,2} + a_{3,1}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} \\ 0 & 0 & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} \\ 0 & 0 & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} & a_{1,2}a_{2,2} + a_{3,2}a_{3,3} \\ 0 & 0 & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} \\ 0 & 0 & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} \\ 0 & 0 & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} \\ 0 & 0 & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3} + a_{3,2}a_{3,3} \\ 0 & 0 & a_{1,1}a_{1,3} + a_{2,1}a_{2,3} + a_{3,1}a_{3,3} & a_{1,2}a_{1,3} + a_{2,2}a_{2,3$$

Method that performs overlap-add w^*w along diagonal is in fact, equivalent to compute Gramian $\Phi^*\Phi$ directly