S.1 Winograd Matrices

Comparison among Winograd transform matrices (A^T, B^T, G) for four algorithms ($F(4,3)$, complex-$F(4,3)$, $F(6,3)$ and $F(2,3)$). Although $F(6,3)$ S1.C algorithm provides a bigger MAC reduction ($5\times$) compared to other algorithms, its matrices present non-integer values that require floating-point units to perform transformations, increasing the computational complexity. $F(2,3)$ S1.D transform matrices contain integer hardware-friendly coefficients that minimize the numerical error discussed in this paper, however, MAC reduction is limited compared to other algorithms ($2.25\times$). We identified the $F(4,3)$ S1.A algorithm as the best compromise in terms of MAC reduction ($4\times$), memory overhead for offline weights transformation and numerical error. In this paper we provide a novel technique to recover the accuracy degradation introduced by 8-bit quantized Winograd algorithms. Our approach performs even better on the complex-$F(4,3)$ S1.B algorithm, where, at the cost of a *slightly* lower MAC reduction ($3.13\times$), we fully recover the accuracy degradation for the full-8 bit Winograd algorithm.

F(4,3)

$$B^T = \begin{bmatrix} 4 & 0 & -5 & 0 & 1 & 0 \\ 0 & -4 & -4 & 1 & 1 & 0 \\ 0 & 4 & -4 & -1 & 1 & 0 \\ 0 & -2 & -1 & 2 & 1 & 0 \\ 0 & 2 & -1 & -2 & 1 & 0 \\ 0 & 4 & 0 & -5 & 0 & 1 \end{bmatrix} G = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ -\frac{1}{6} & -\frac{1}{6} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{1}{6} & -\frac{1}{6} \\ \frac{1}{24} & \frac{1}{12} & \frac{1}{6} \\ \frac{1}{24} & -\frac{1}{12} & \frac{1}{6} \\ 0 & 0 & 1 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & 0 \\ 0 & 1 & 1 & 4 & 4 & 0 \\ 0 & 1 & -1 & 8 & -8 & 1 \end{bmatrix} \tag{S1.A}$$

F(4,3) complex

$$B^T = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & -1 & 1 & -1 & 1 & 0 \\ 0 & -j & -1 & j & 1 & 0 \\ 0 & j & -1 & -j & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix} G = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{j}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{j}{4} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & j & -j & 0 \\ 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & -1 & -j & j & 1 \end{bmatrix} \tag{S1.B}$$

F(6,3)

$$B^T = \begin{bmatrix} 1 & 0 & -21/4 & 0 & 21/4 & 0 & -1 & 0 \\ 0 & 1 & 1 & -17/4 & -17/4 & 1 & 1 & 0 \\ 0 & -1 & 1 & 17/4 & -17/4 & -1 & 1 & 0 \\ 0 & 1/2 & 1/4 & -5/2 & -5/4 & 2 & 1 & 0 \\ 0 & -1/2 & 1/4 & 5/2 & -5/4 & -2 & 1 & 0 \\ 0 & 2 & 4 & -5/2 & -5 & 1/2 & 1 & 0 \\ 0 & -2 & 4 & 5/2 & -5 & -1/2 & 1 & 0 \\ 0 & -1 & 0 & 21/4 & 0 & -21/4 & 0 & 1 \end{bmatrix} G = \begin{bmatrix} 1 & 0 & 0 \\ -2/9 & -2/9 & -2/9 \\ -2/9 & 2/9 & -2/9 \\ 1/90 & 1/45 & 2/45 \\ 1/90 & -1/45 & 2/45 \\ 32/45 & 16/45 & 8/45 \\ 32/45 & 16/45 & 8/45 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & 1 & 4 & 4 & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 1 & -1 & 8 & -8 & \frac{1}{8} & -\frac{1}{8} & 0 \\ 0 & 1 & 1 & 16 & 16 & \frac{1}{16} & \frac{1}{16} & 0 \\ 0 & 1 & -1 & 32 & -32 & \frac{1}{32} & -\frac{1}{32} & 1 \end{bmatrix} \tag{S1.C}$$

F(2,3)

$$B^T = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} G = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 \end{bmatrix} \tag{S1.D}$$

S.2 Complex tile organization

In Fig. 4, the complex-Winograd $F(4,3)$ algorithm is shown. Green and yellow cells represent real and complex values, respectively. Each matrix in the complex-Winograd domain contains 10 pairs of complex conjugate values and 16 real values. Each complex conjugate pair (x_{c1} and x_{c2}) is defined as reported in equation S2.A.

$$x_{c1} = x_r + jx_j \qquad\qquad x_{c2} = x_r - jx_j \tag{S2.A}$$

x_r and x_j represent the real and imaginary parts, respectively. Exploiting the complex conjugates property, we can store only the real part and the imaginary part, and build (if necessary), the complex conjugate by adding and subtracting the two values. This is shown in in Fig. 5, where the blue and yellow parts can be combined to reproduce the needed complex conjugates accordingly. Moreover, the number of *real* multiplications required to perform the complex element-wise matrix multiplication (EWMM) should be $16 + 4 \times 20 = 96$. However, the 20 complex multiplications can be grouped into 10 pairs of complex conjugate multiplications and by using the Karatzuba algorithm, each complex multiplication takes only three real multiplications. Therefore, the total number of *real* multiplications needed to perform complex EWMM can be rewritten as: $16 + 3 \times 10 = 46$.

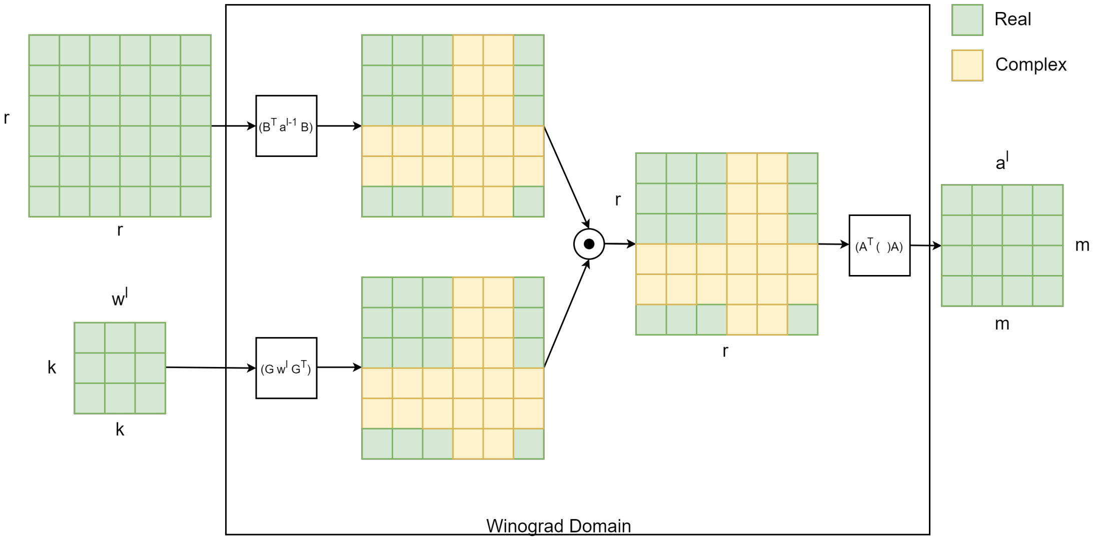


Figure 4. Complex Winograd $F(4,3)$ algorithm. Green and yellow elements represent real and complex values, respectively.

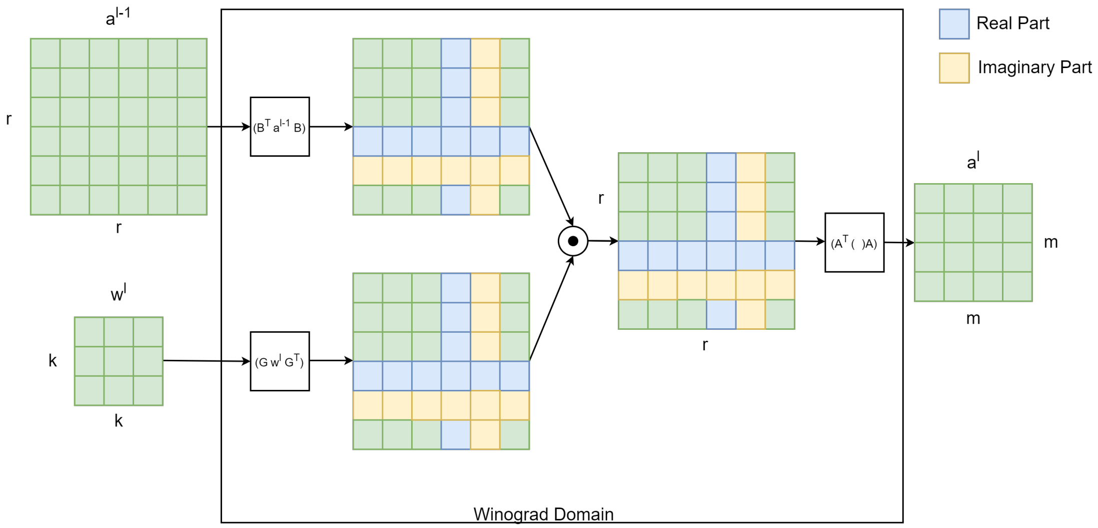
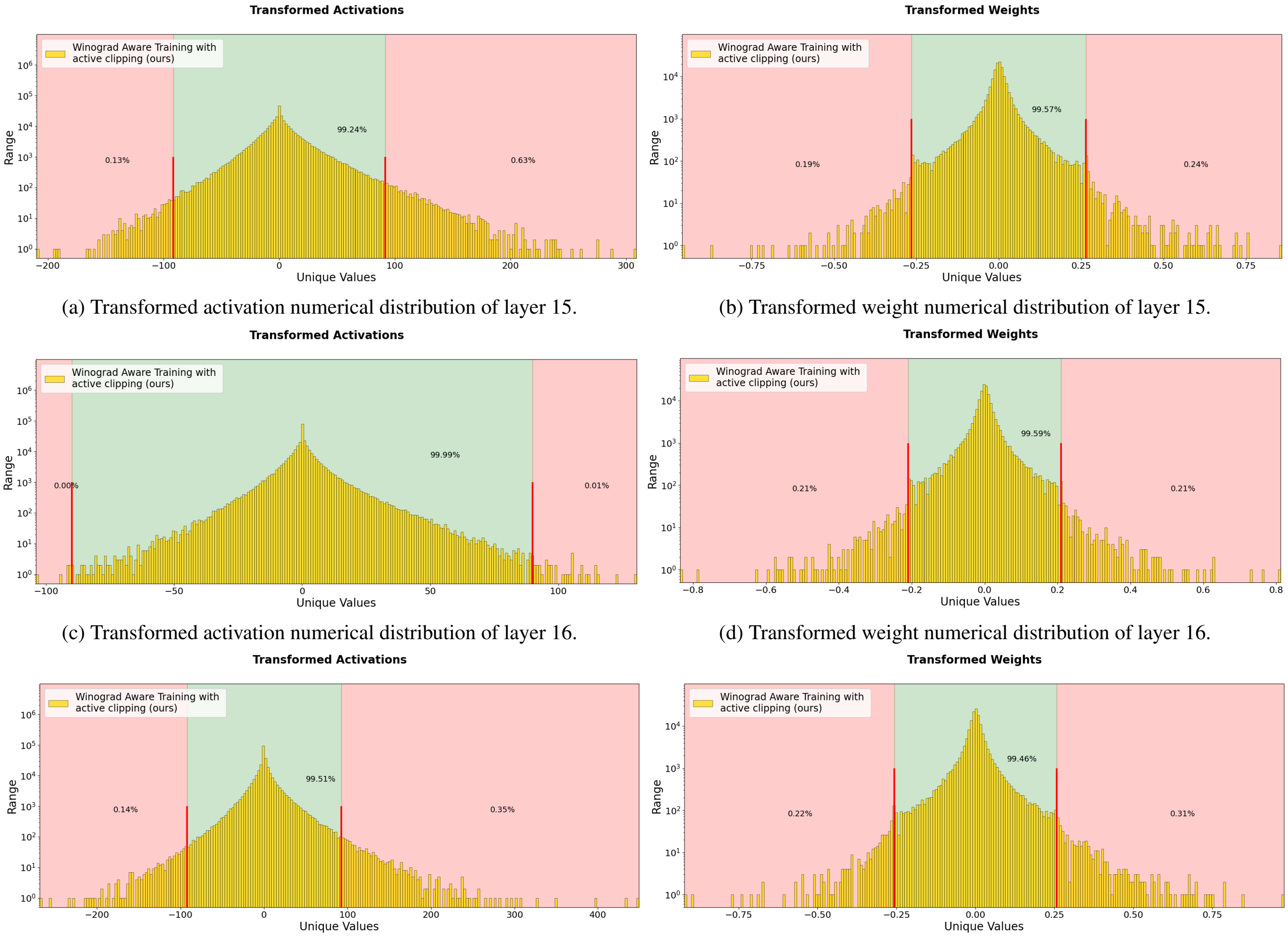


Figure 5. Efficient Complex Winograd $F(4, 3)$ algorithm representation. Complex conjugates values can be stored as real (blue) and imaginary parts (yellow).

S.3 Effectiveness of Clipping Factors

In Fig. 6, we show further examples on the distributions of transformed weights and transformed activations when using the proposed method with Winograd $F(4, 3)$ algorithm for three layers of the ResNet-20 model (layers 15, 16, and 17, respectively) trained on CIFAR-10. For activations, the overflow factor increases the numerical range in the Winograd domain, causing a huge quantization error that leads to severe accuracy degradation. Our approach allows to dynamically limit the distribution, guaranteeing a better exploitation of the quantized range. Our method also effectively clips the transformed weights, maintaining over 99% of the numbers appearing in the transform. More generally, the clipping range highlighted in green sufficiently covers all the necessary data to achieve a transformation with no accuracy degradation when accelerating full 8-bit Winograd convolution on edge hardware.



(a) Transformed activation numerical distribution of layer 15.

(b) Transformed weight numerical distribution of layer 15.

(c) Transformed activation numerical distribution of layer 16.

(d) Transformed weight numerical distribution of layer 16.

(e) Transformed activation numerical distribution of layer 17.

(f) Transformed weight numerical distribution of layer 17

Figure 6. Numerical distributions of example layers for transformed weights and activations of ResNet-20 on CIFAR-10. The values in the clipped range (green) sufficiently contain the information needed to maintain high-accuracy full 8-bit Winograd.