Method 2
Compressing the Convolution

$$\mathcal{Q}(f,h,w,r_4) = \sum_{s=1}^{S} \boldsymbol{U}^{(4)}(s,r_4)\ \mathcal{X}(f,h,w,s)$$

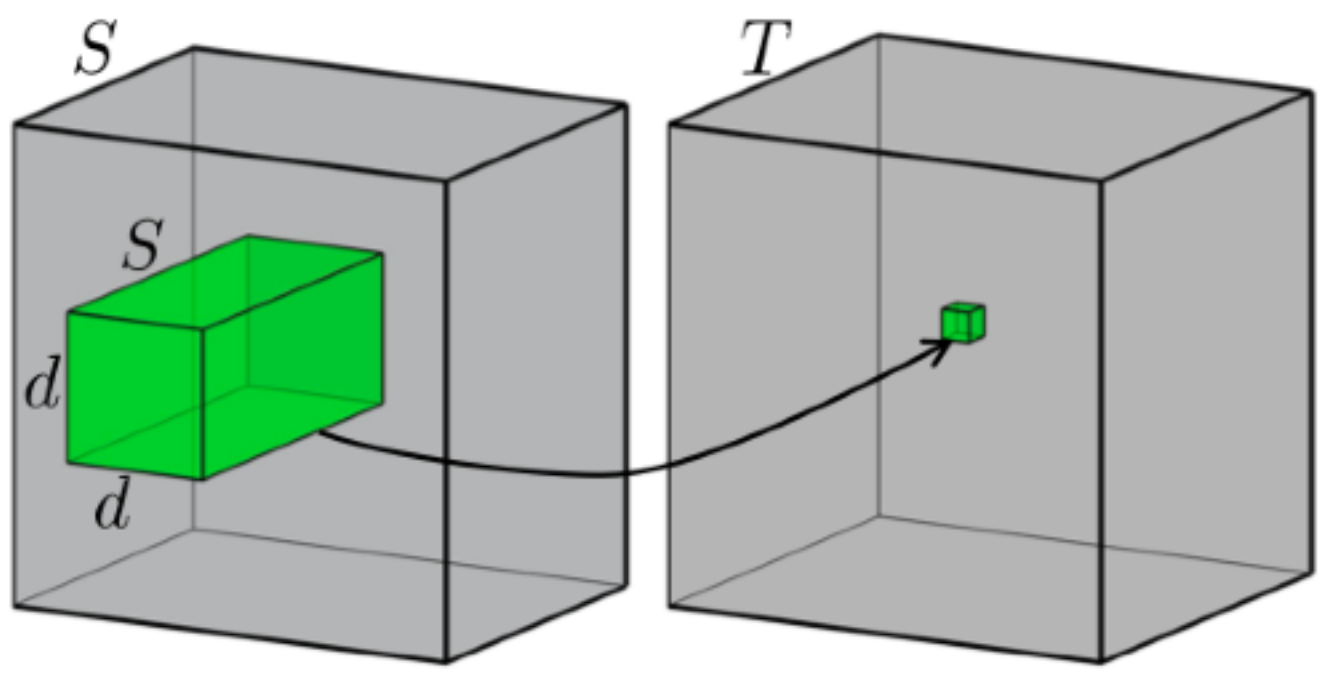
$1 \times 1 \times 1$ convolution with S input channels and R_4 output channels

$$\mathcal{Q}'(f',h',w',r_5) = \sum_{i=1}^{D_F}\sum_{j=1}^{D_H}\sum_{l=1}^{D_W}\sum_{r_4=1}^{R_4} \mathcal{C}(i,j,l,r_4,r_5)\mathcal{Q}(f_i,h_j,w_l,r_4)$$

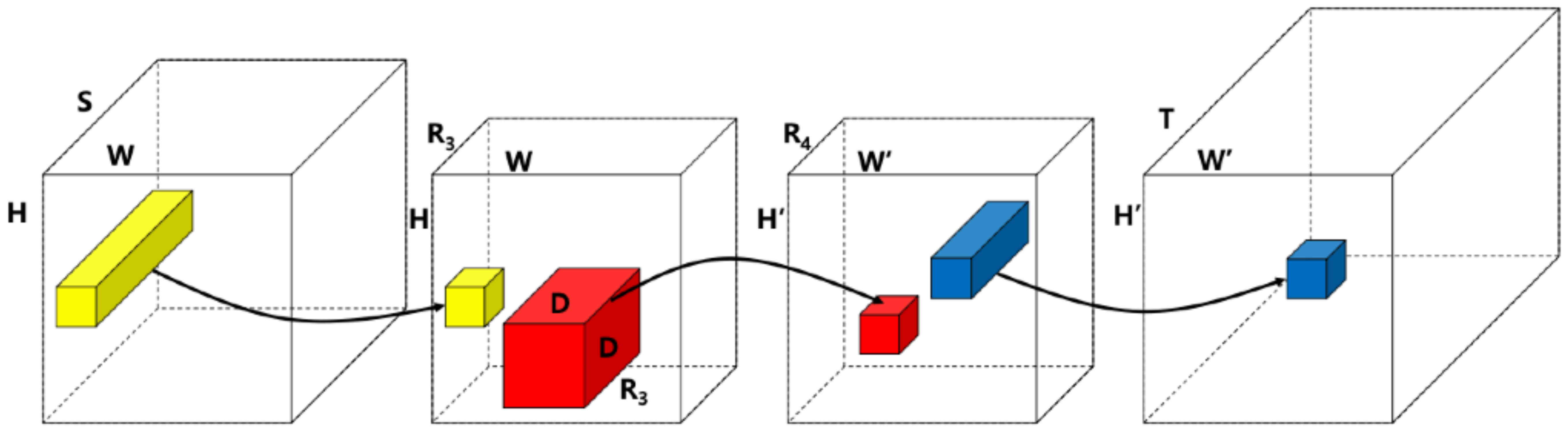
$D_F \times D_H \times D_W$ convolution with R_4 input channels and R_5 output channels

$$\mathcal{Y}(f',h',w',t) = \sum_{r_5=1}^{R_5} \boldsymbol{U}^{(5)}(t,r_5)\ \mathcal{Q}'(f',h',w',r_5)$$

$1 \times 1 \times 1$ convolution with R_5 input channels and T output channels



Original convolution



Sequence of convolutions of the compressed version

Illustration of the 2D case with
different Rs due to lower dimensions

Method 2

Compressing the Linear Layer

The original layer

$$\boldsymbol{y}^{N_{out}} = \boldsymbol{W}^{N_{out} \times N_{in}} \boldsymbol{x}^{N_{in}} + \boldsymbol{b}^{N_{out}}$$

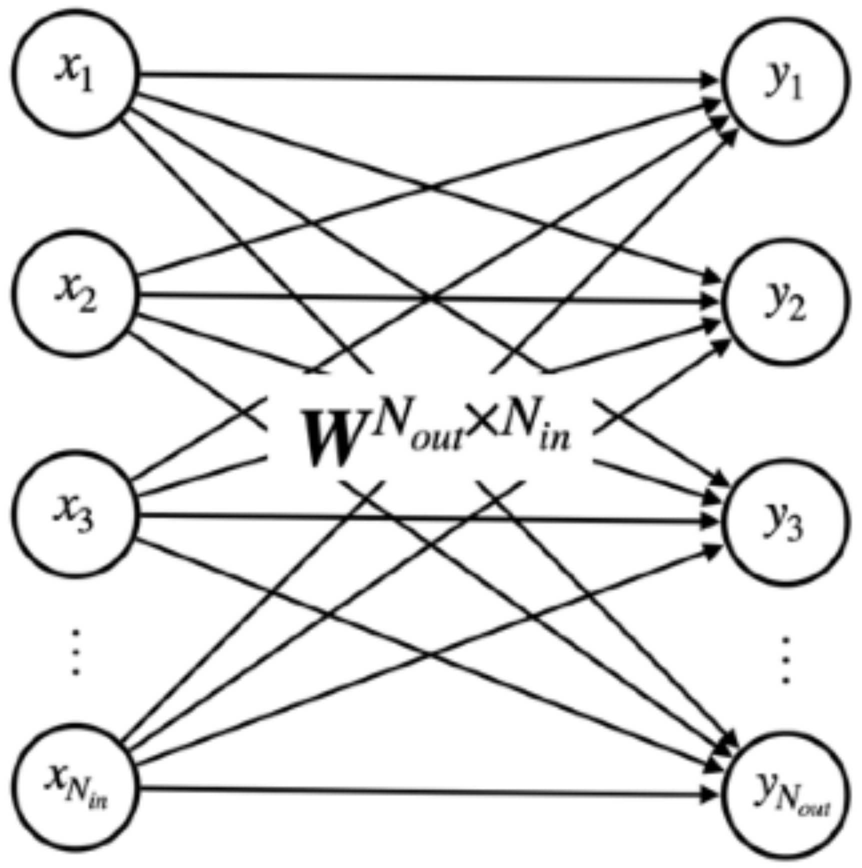
Tucker-2 approximation of weight matrix

$$\boldsymbol{W} \approx \boldsymbol{G} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} = \boldsymbol{A}\boldsymbol{G}\boldsymbol{B}^\top$$

The compressed layer

$$\boldsymbol{y}^{N_{\text{out}}} \approx \underbrace{\boldsymbol{A}^{N_{\text{out}} \times R_A} \boldsymbol{G}^{R_A \times N_{\text{in}}} \underbrace{\boldsymbol{B}^{\top R_B \times N_{\text{in}}} \boldsymbol{x}^{N_{\text{in}}}}_{\boldsymbol{q}_1}}_{\boldsymbol{q}_2} + \boldsymbol{b}^{N_{\text{out}}}$$

The original layer



The compressed layer