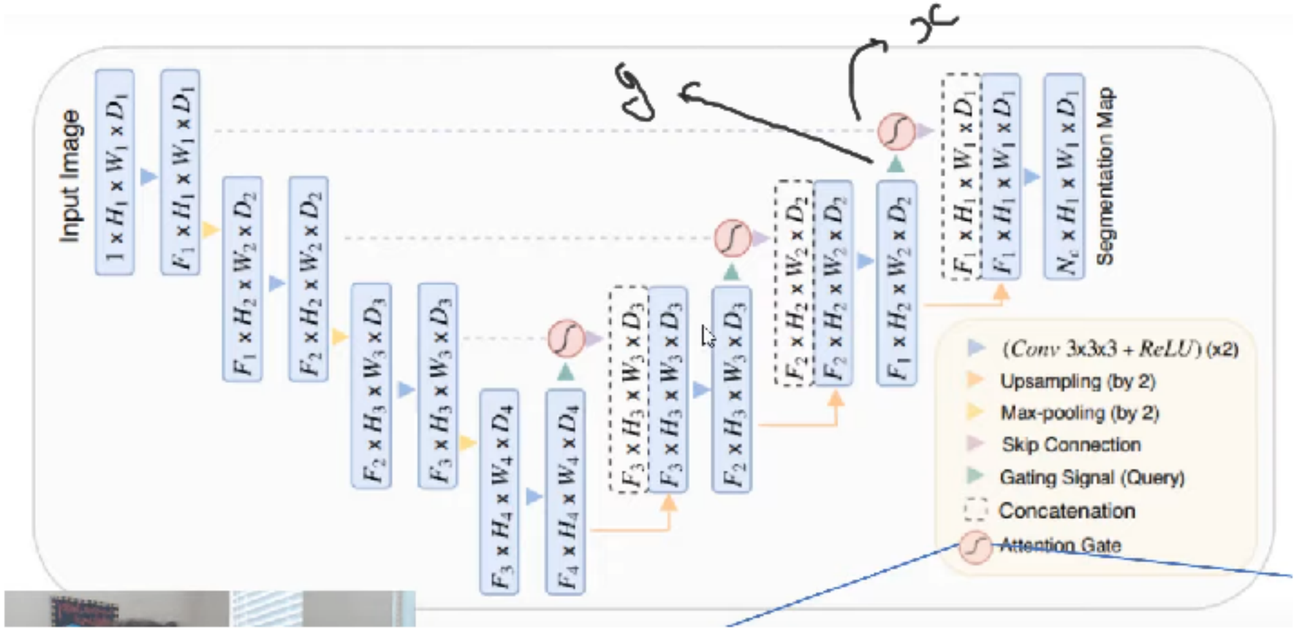


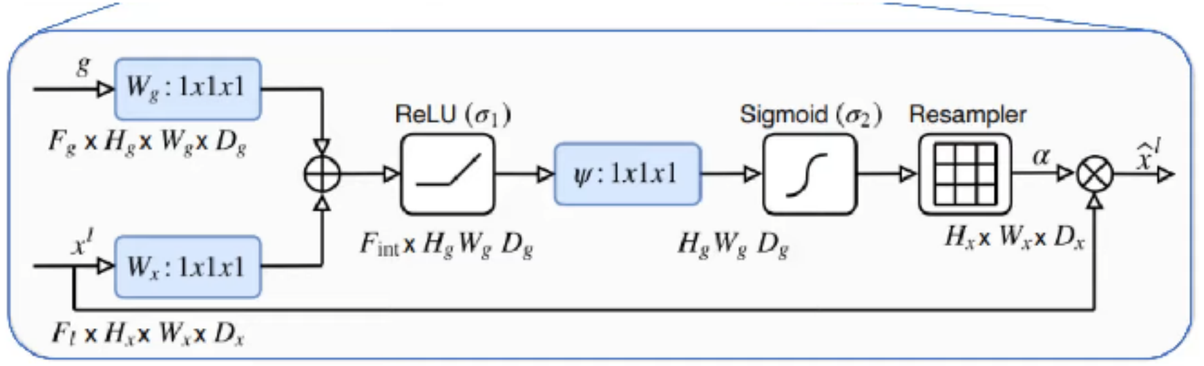
attention gate takes x & g

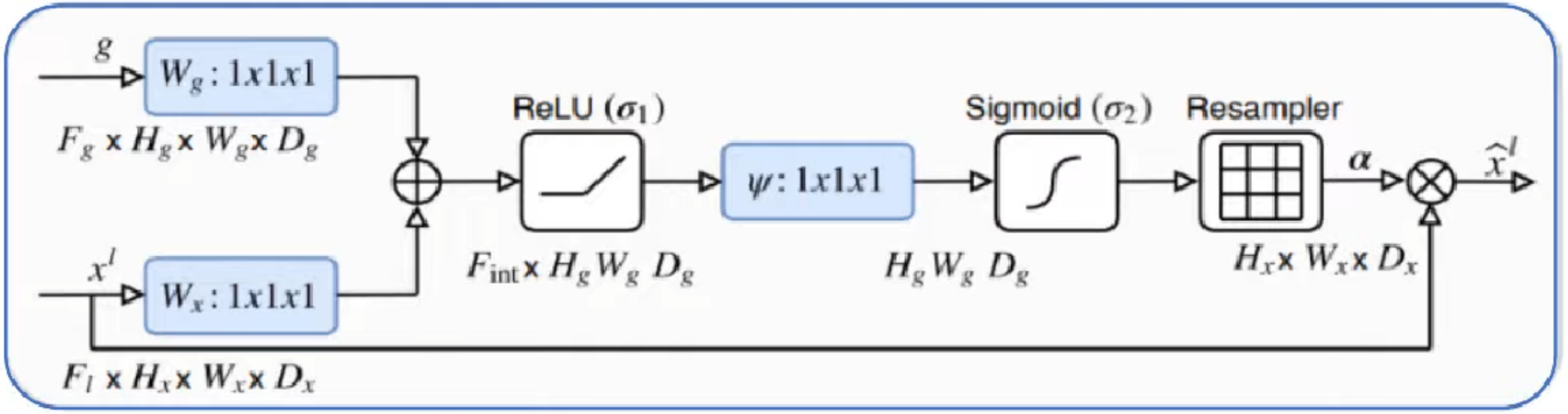
x ⇒ comes from skip connection

↳ it provide better information

g ⇒ gating signal ⇒ comes from lowest layer of network

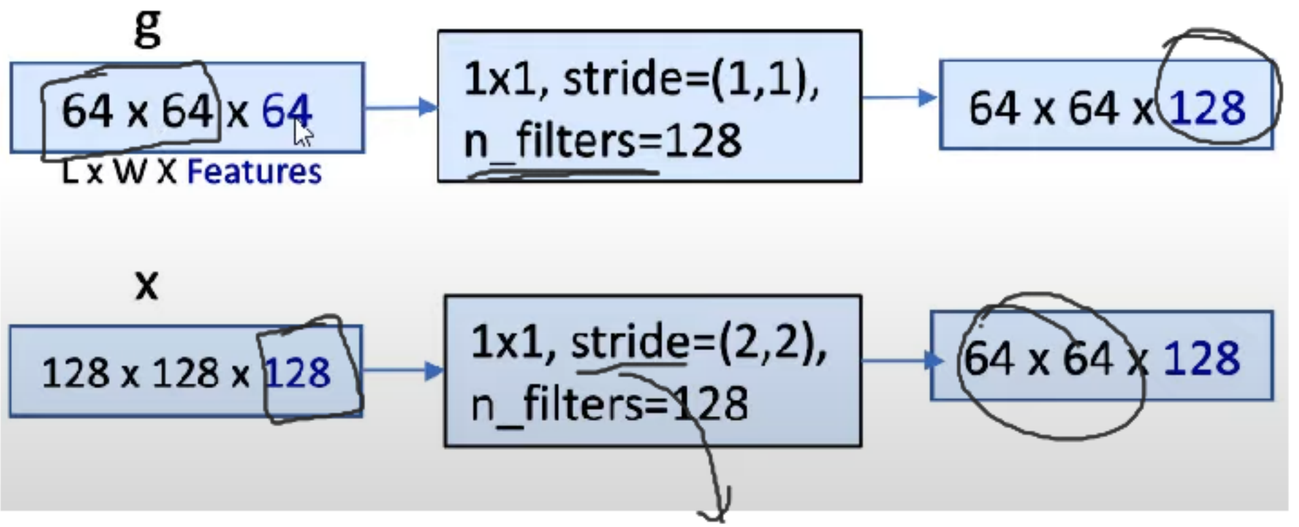
↳ it provide better representation



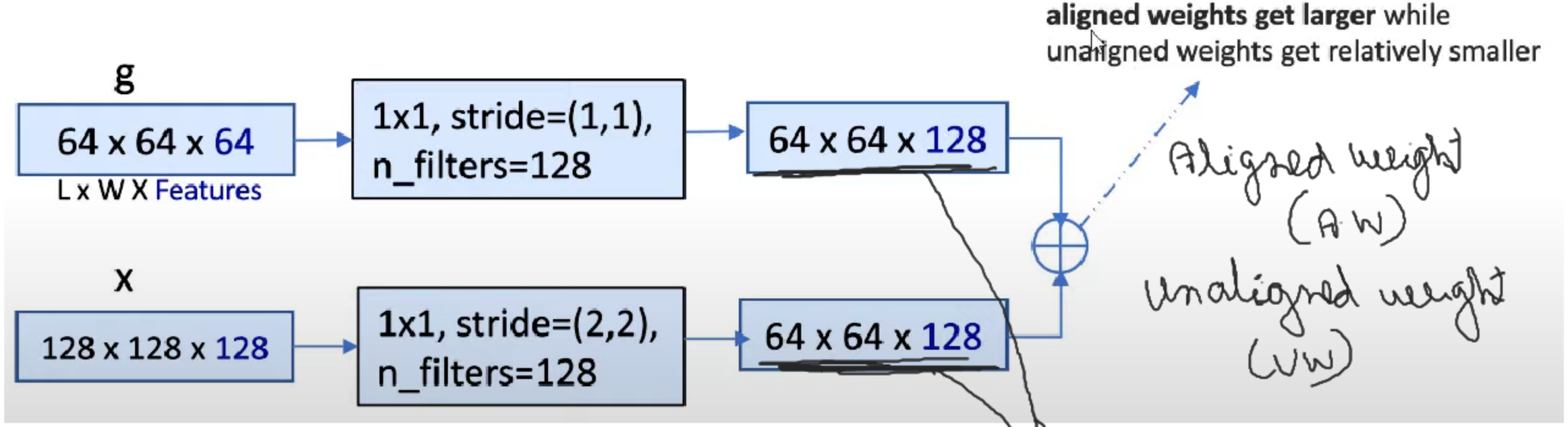


Taking Size of g
[64 x 64]

Taking Channel of x [128]

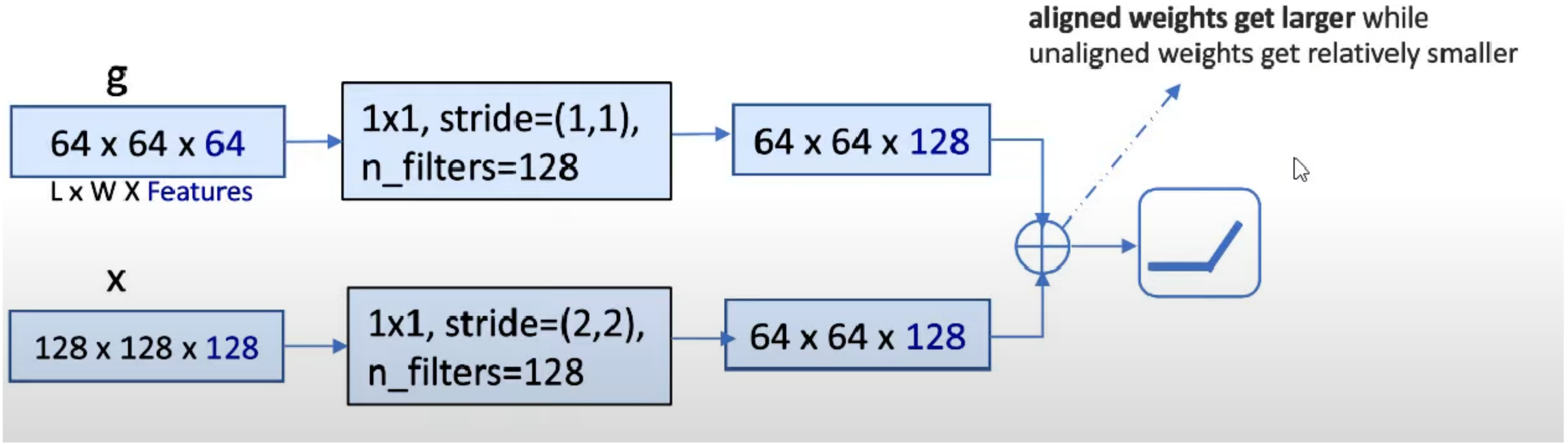


Size become half



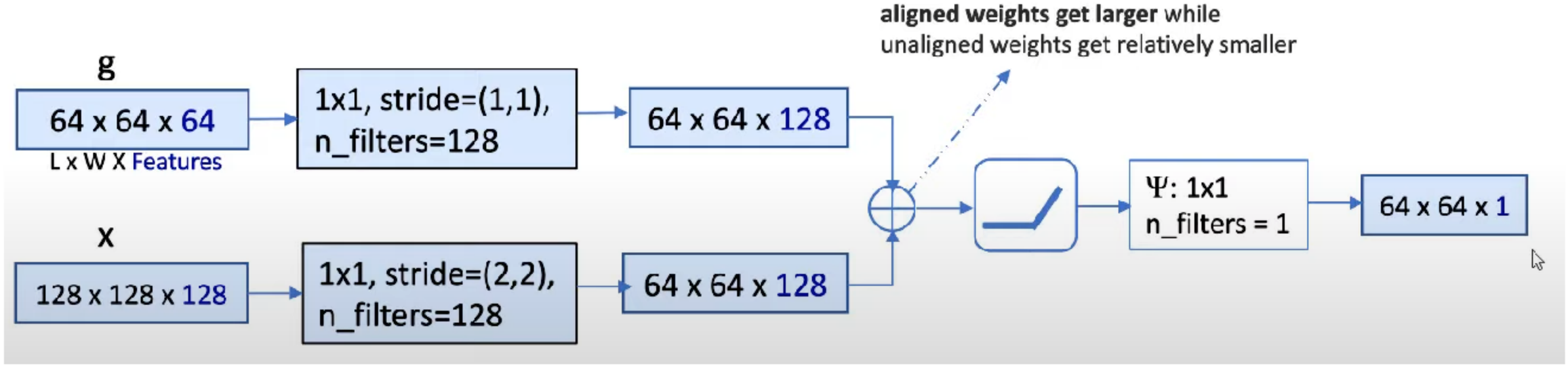
The reason to add both is that to aligned weight which makes weight of unaligned weight much smaller,

eg → A.W = 0.9 → 0.9 + 0.9 ⇒ 1.8 ↩ relatively
U.W = 0.1 → 0.1 + 0.1 ⇒ 0.2 ↩ smaller



After adding, passing it through ReLU activation function which represent

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}$$



passing through Ψ function with filters = 1

which gives output as 64 x 64 x 1

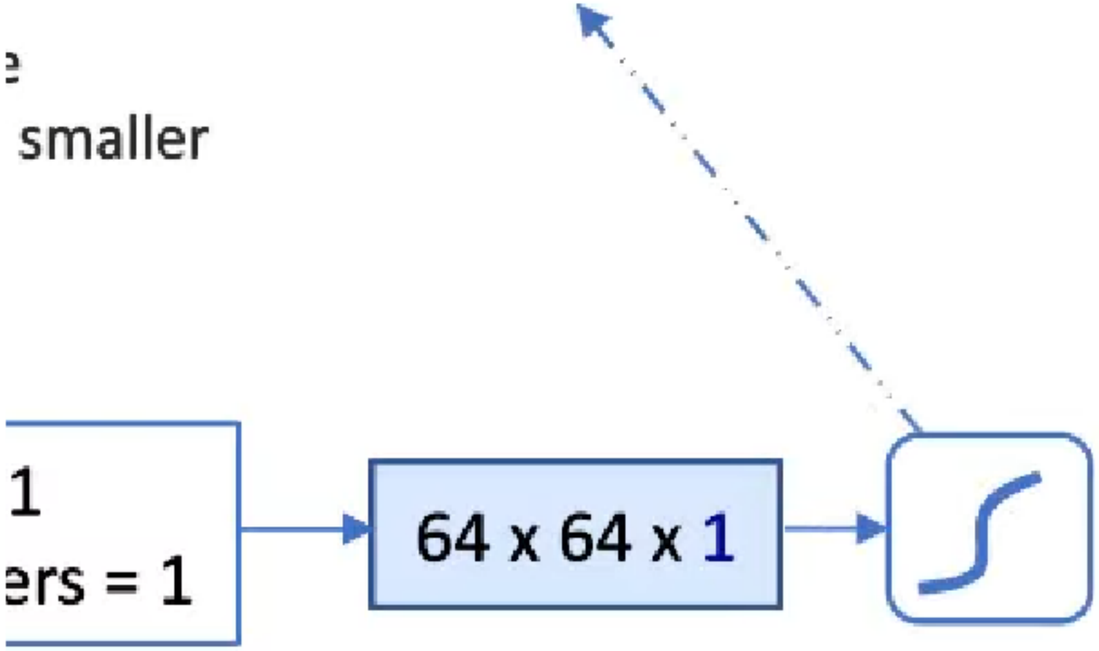
These are nothing but weights but its range could be

zero → ∞ as we are using ReLU

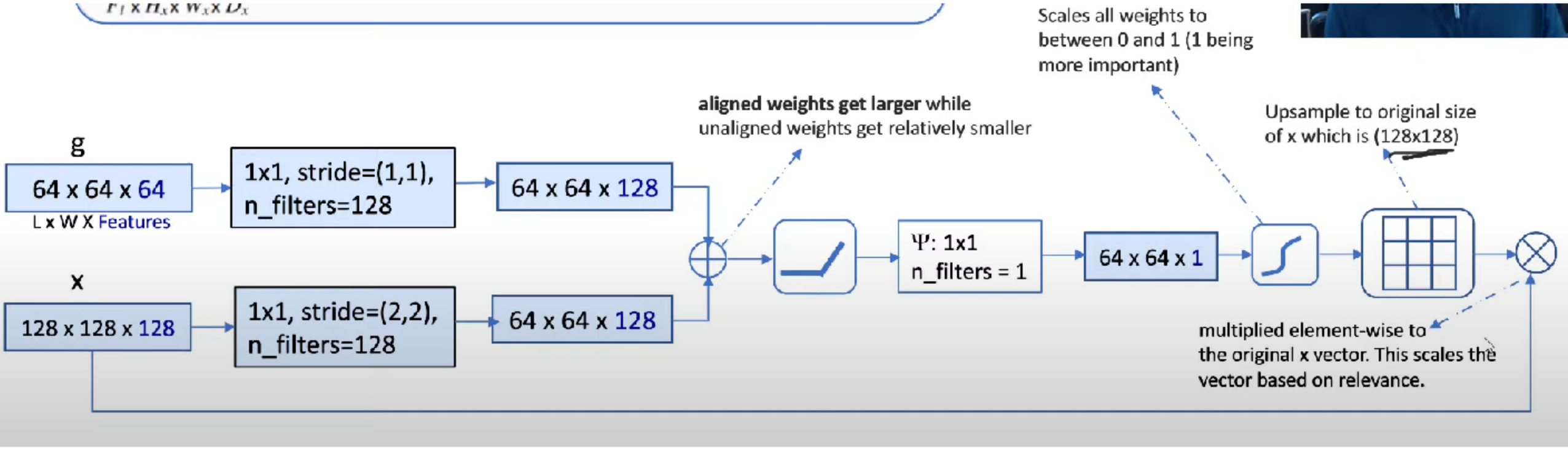
Scales all weights to between 0 and 1 (1 being more important)

e

smaller



So with the help of Sigmoid activation function all value comes in range from 0 to 1

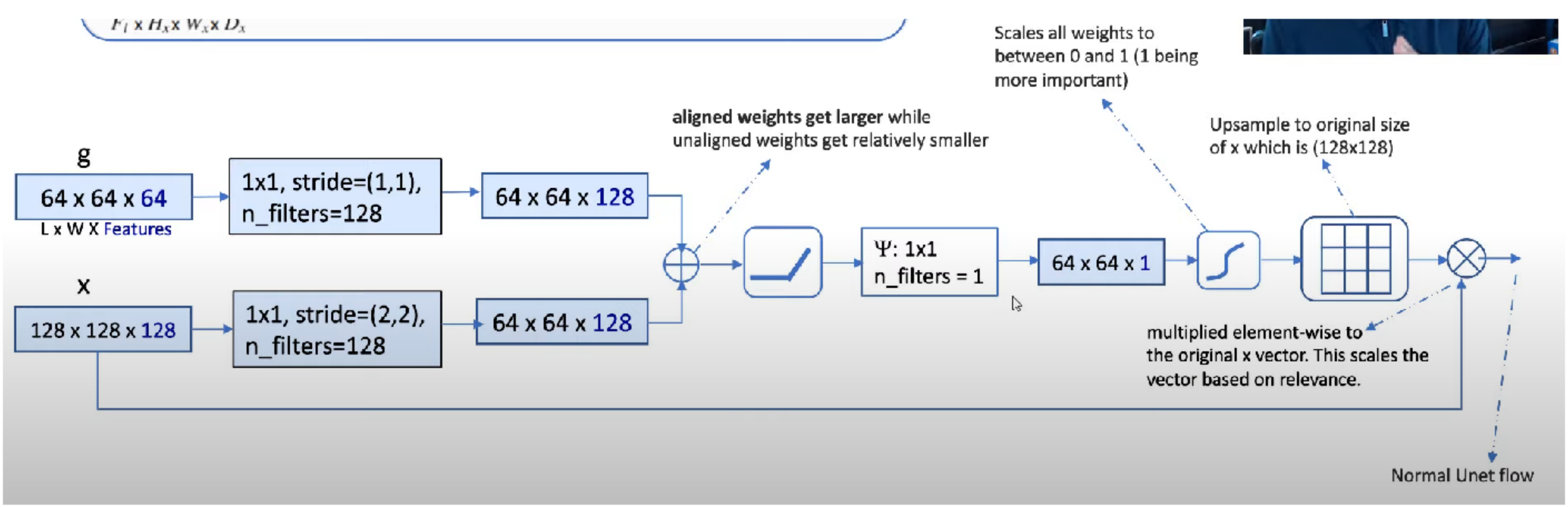


as we get all value in range 0 → 1 we have to up sample to the original size of x

And multiple with x, which scales the relevent part of image

We can say that, at each pixel that coming from x we multiplying the pixel value with the weight value (that we calculated using g)

And output goes to next layer at normal U-net flow


$F_l \times H_x \times W_x \times D_x$
Scales all weights to between 0 and 1 (1 being more important)
aligned weights get larger while unaligned weights get relatively smaller
Upsample to original size of x which is (128x128)
g
64 x 64 x 64
L x W X Features
1x1, stride=(1,1), n_filters=128
64 x 64 x 128
x
128 x 128 x 128
1x1, stride=(2,2), n_filters=128
64 x 64 x 128
Ψ: 1x1 n_filters = 1
64 x 64 x 1
multiplied element-wise to the original x vector. This scales the vector based on relevance.
Normal Unet flow