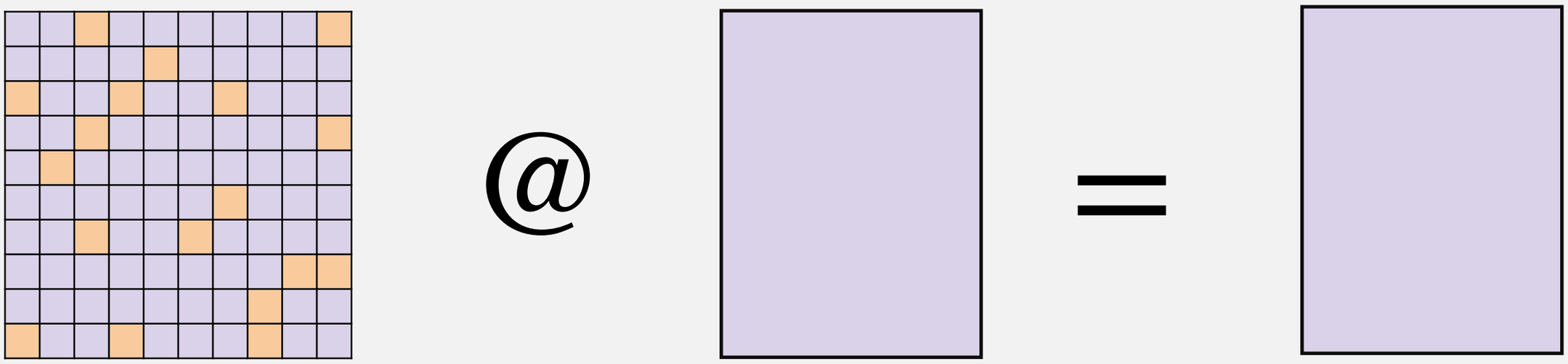
Graph Attention

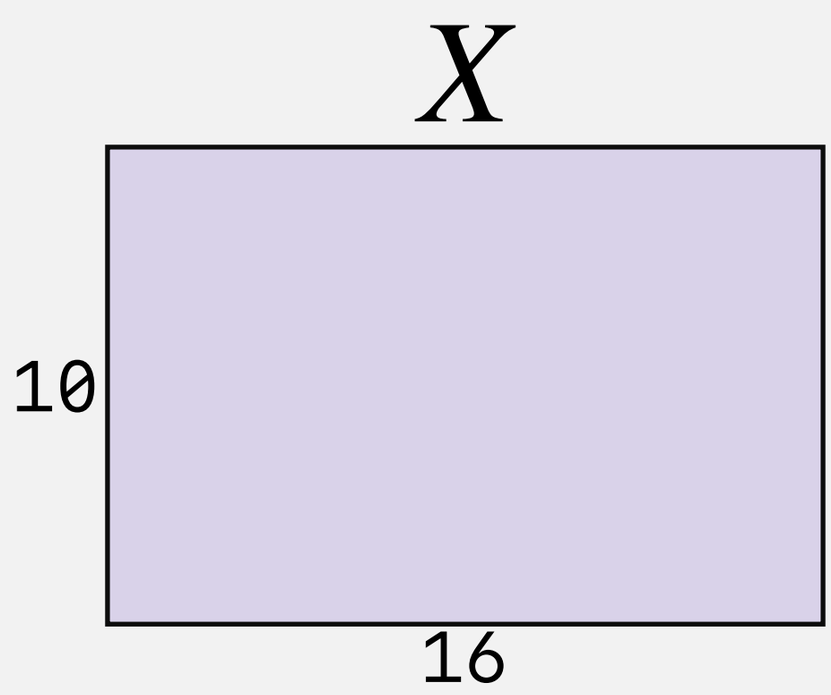
A Walkthrough

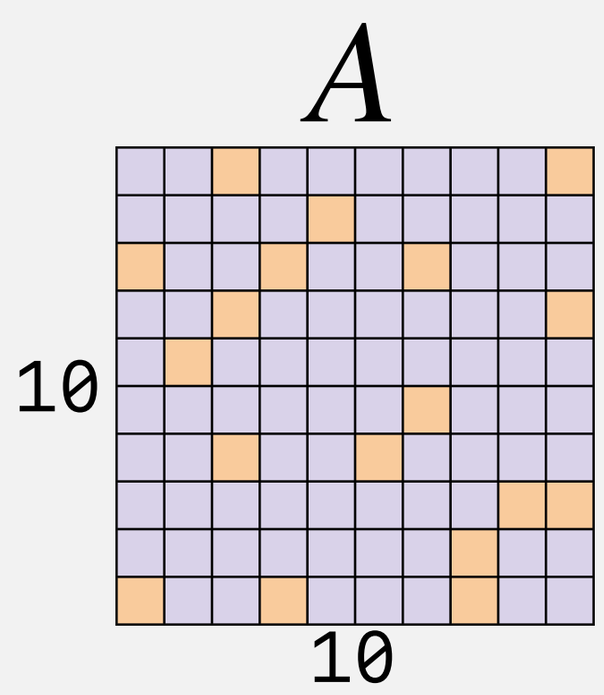


youtube.com/@mashaan14

Inputs

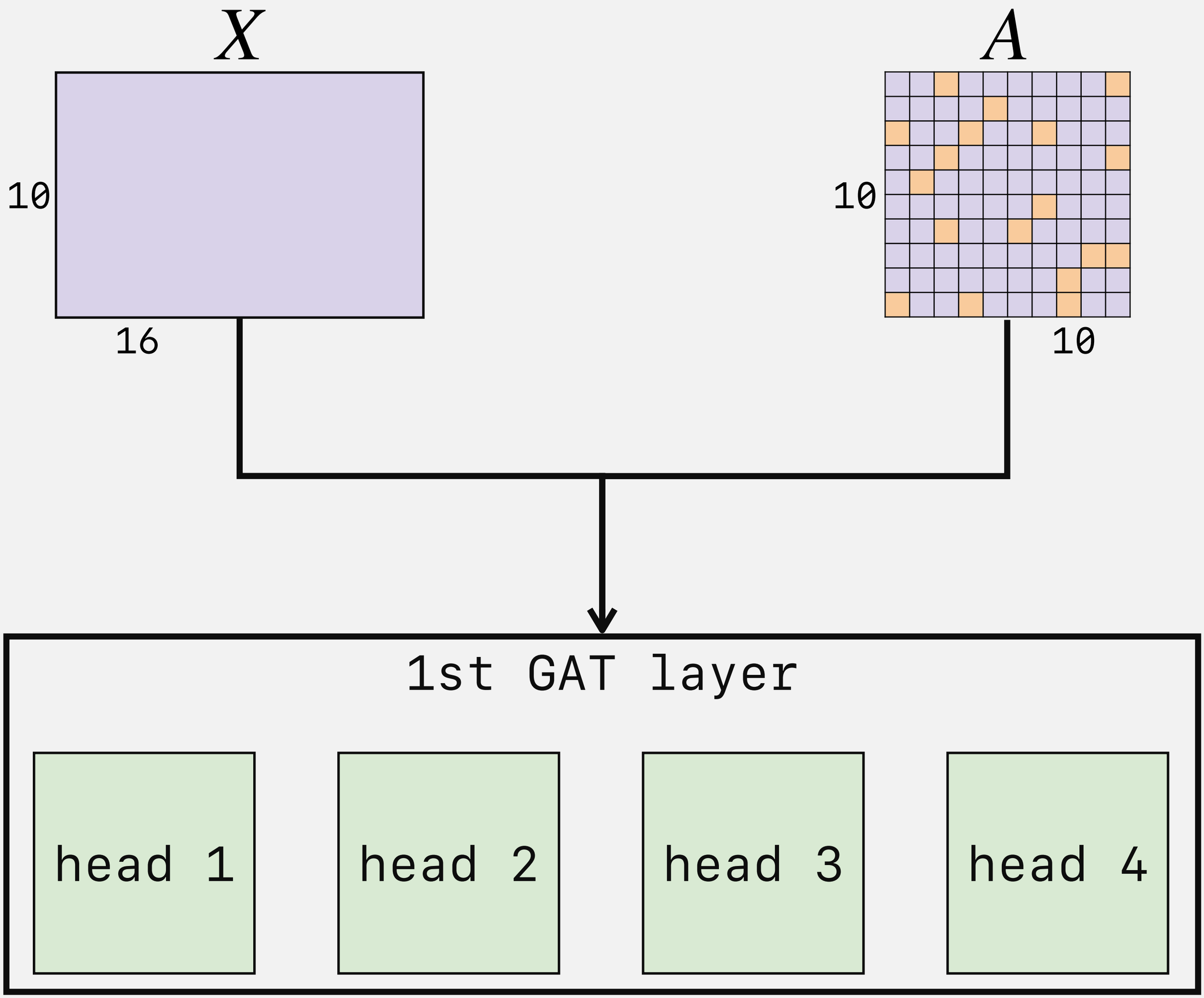
```
# hyperparameters
n_nodes          = 10
in_features      = 16
hidden_features = 8
n_classes        = 7
n_heads          = 4
```





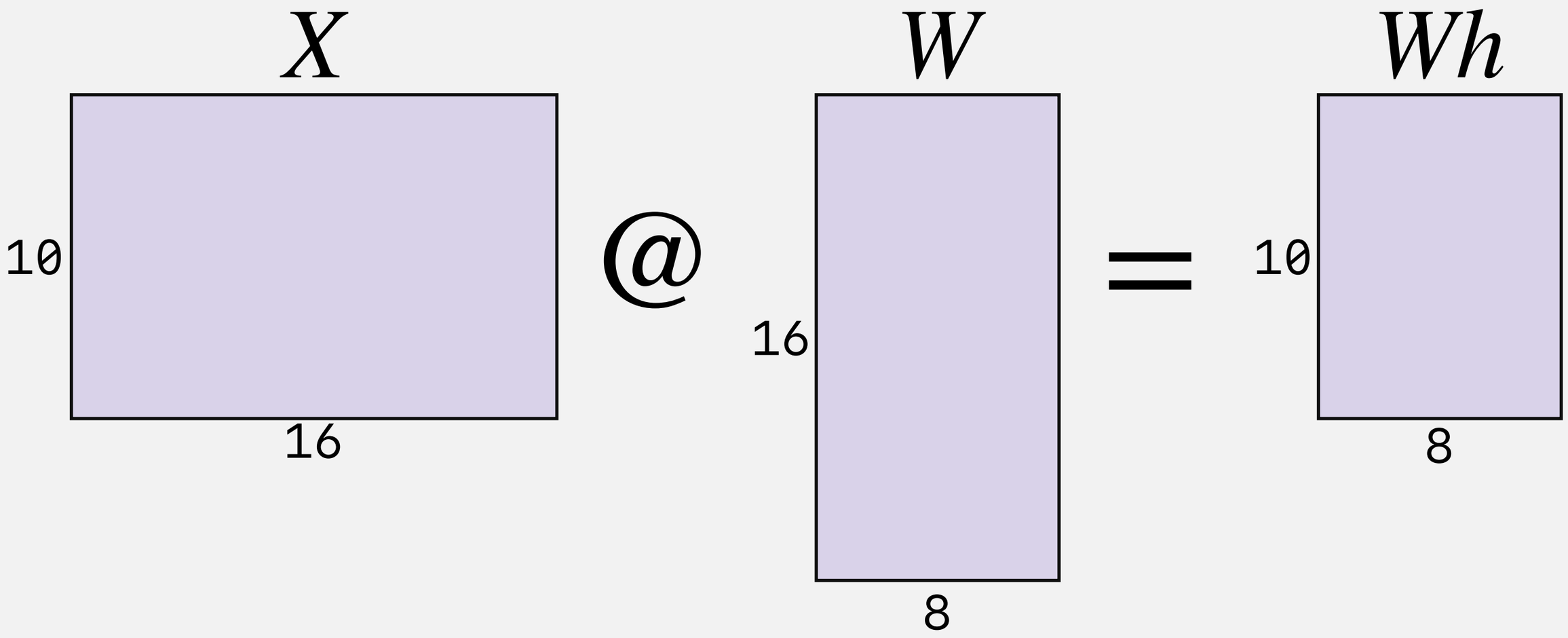
youtube.com/@mashaan14

1st GAT layer, input

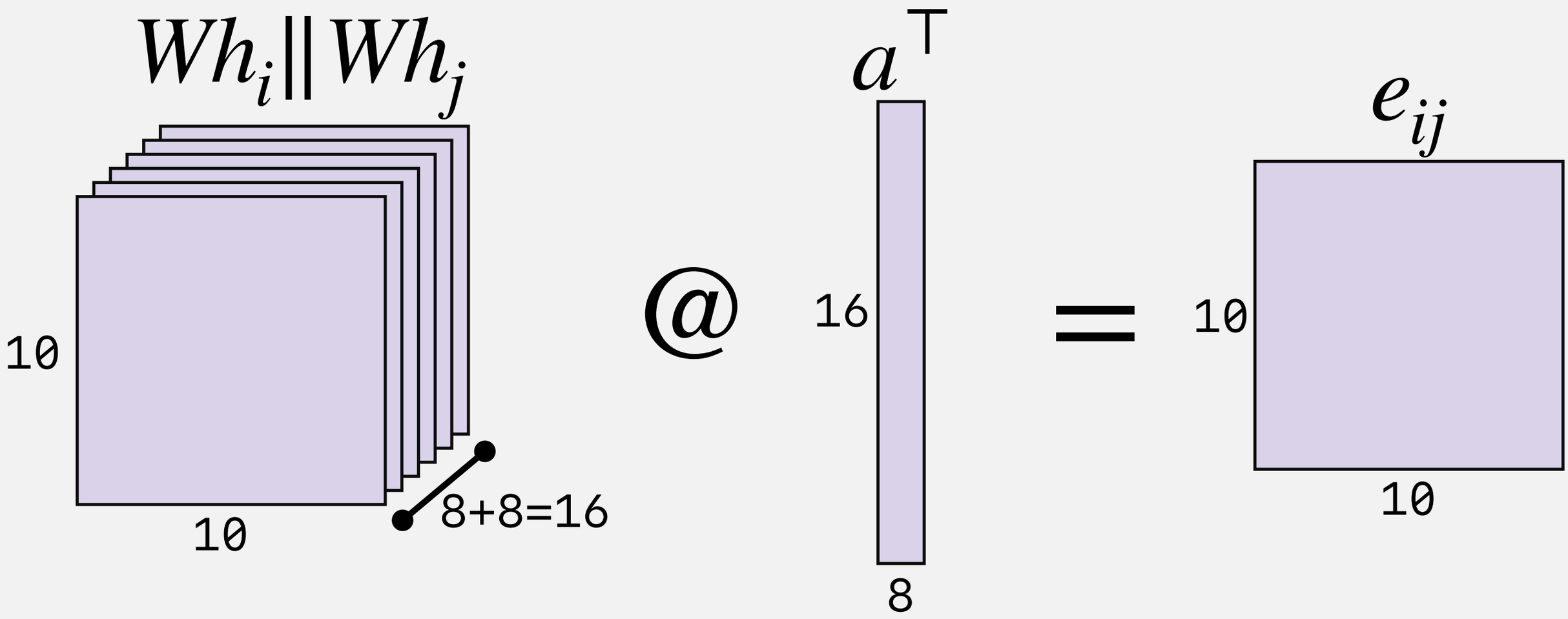


youtube.com/@mashaan14

1st GAT layer, head 1



After applying the weight matrix W, the resulting node representations are denoted as Wh. For any two nodes i and j, the concatenation $Wh_i || Wh_j$ combines their feature vectors along the feature dimension.



youtube.com/@mashaan14

1st GAT layer, head 1

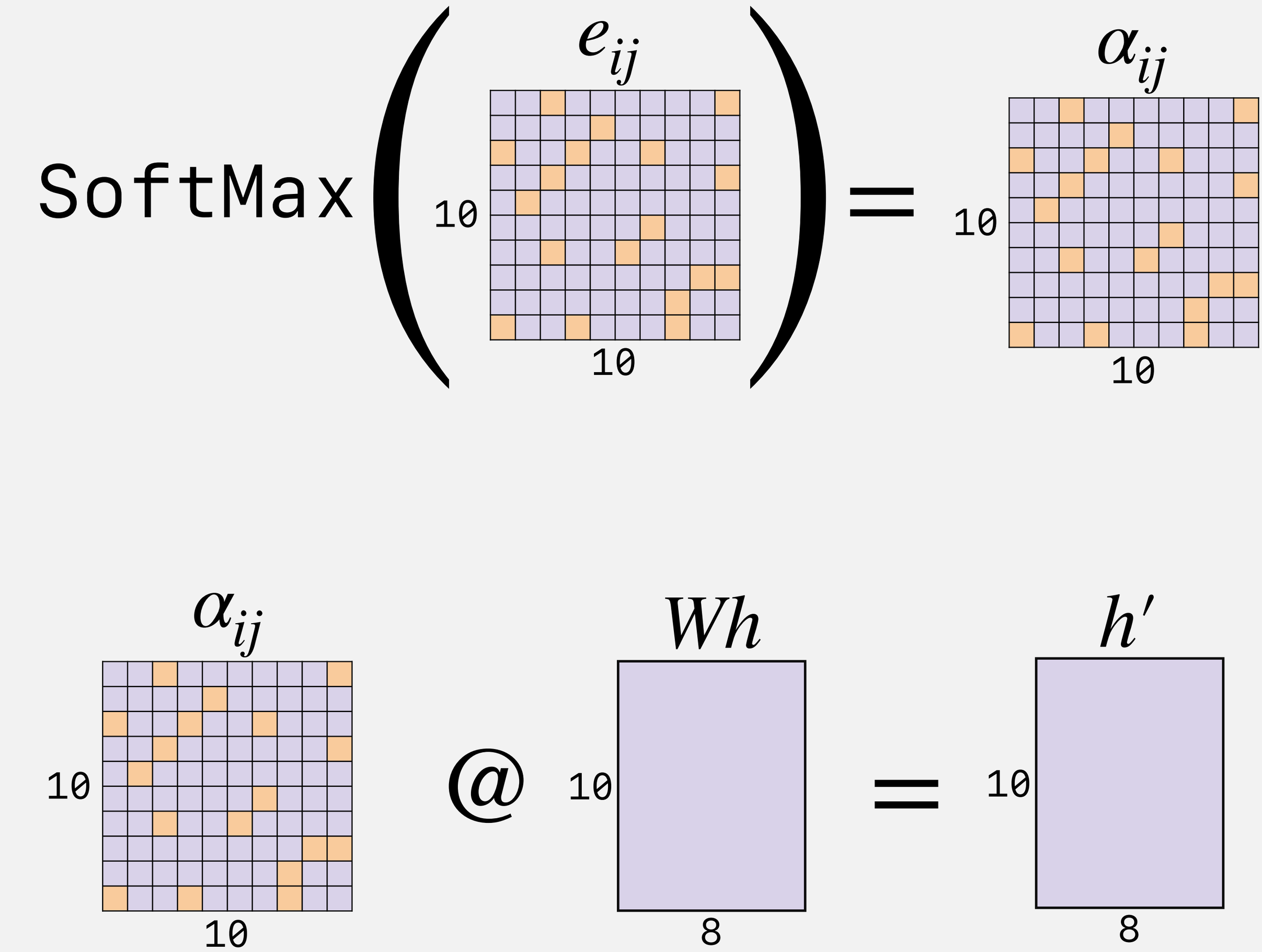
$$\texttt{LeakyReLU}\left(\begin{array}{c} e_{ij} \\ \texttt{10} \\ \\ \texttt{10} \end{array}\right)$$

```
# masked attention
torch.where(adj > 0, e, zero_vec)
```

Masked attention must be applied before the `softmax` function. Reversing this order leads to `softmax` being calculated over incorrect values.

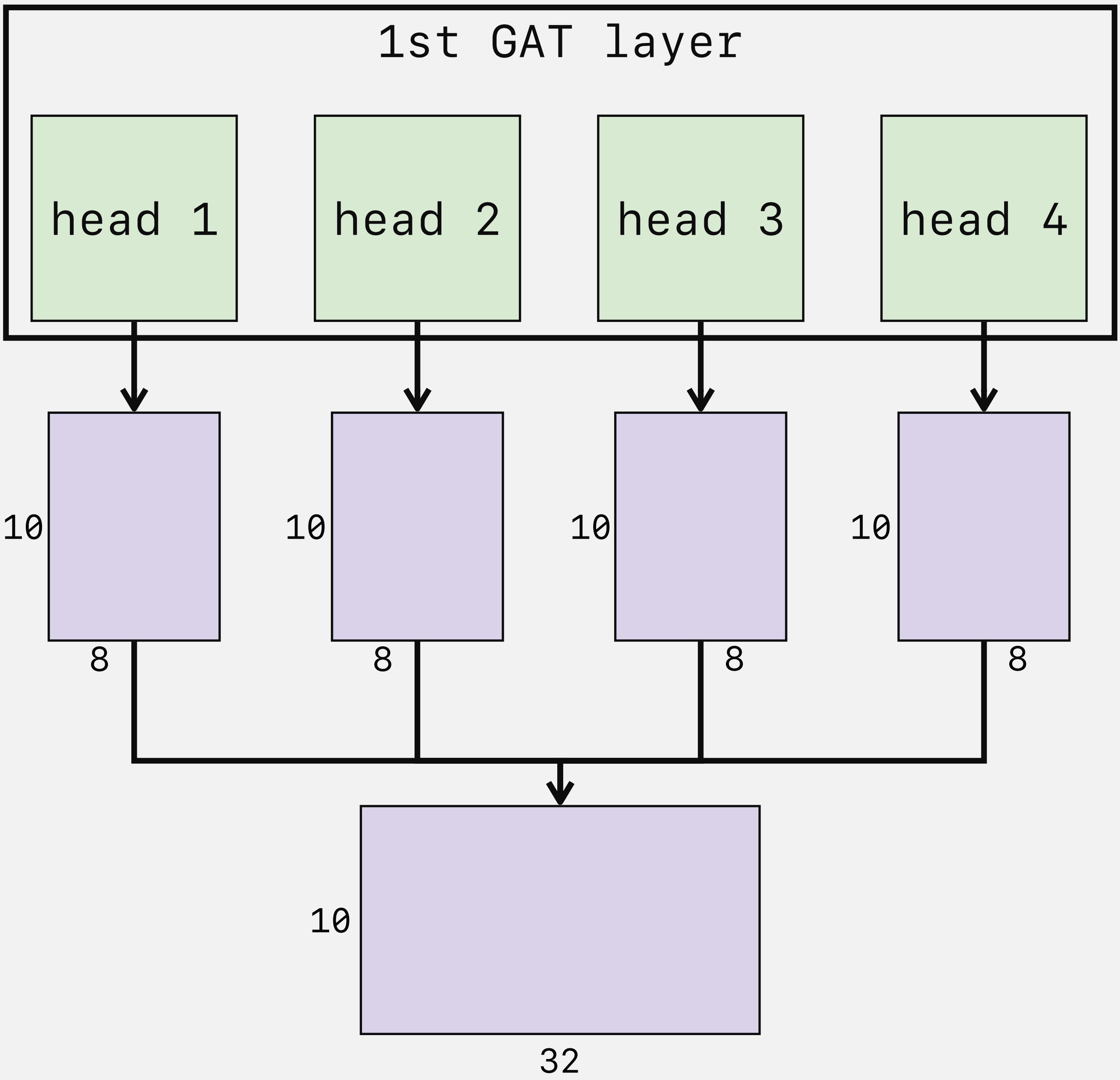
youtube.com/@mashaan14

1st GAT layer, head 1



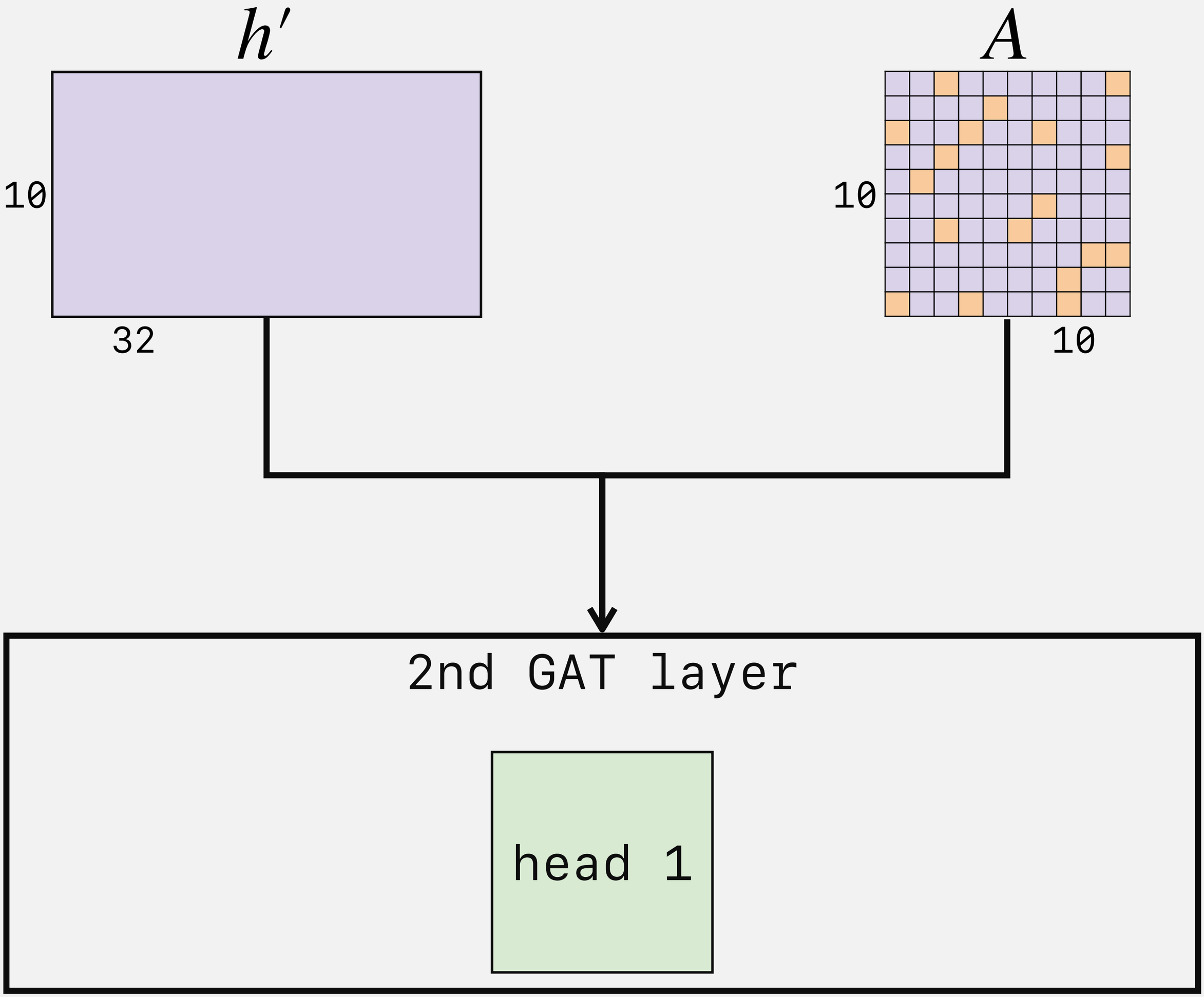
youtube.com/@mashaan14

1st GAT layer, output



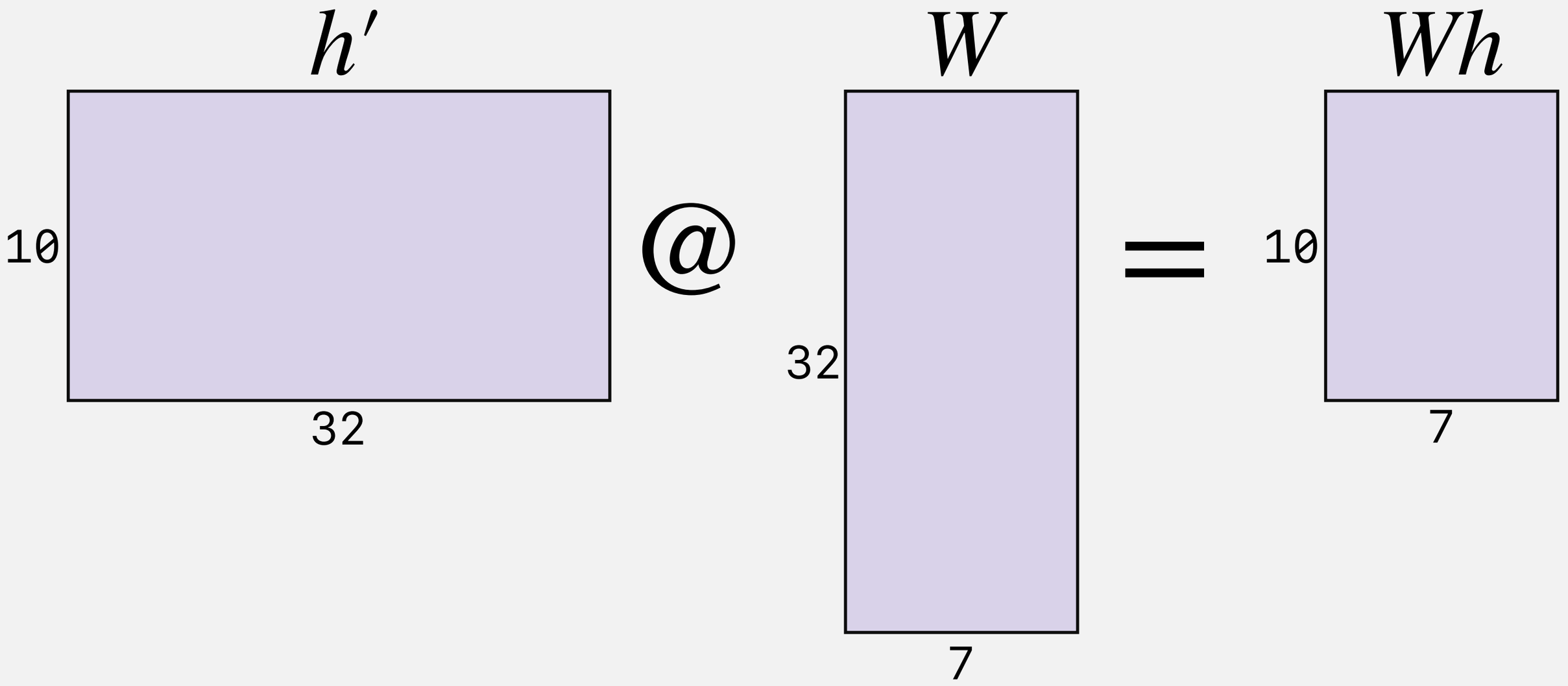
youtube.com/@mashaan14

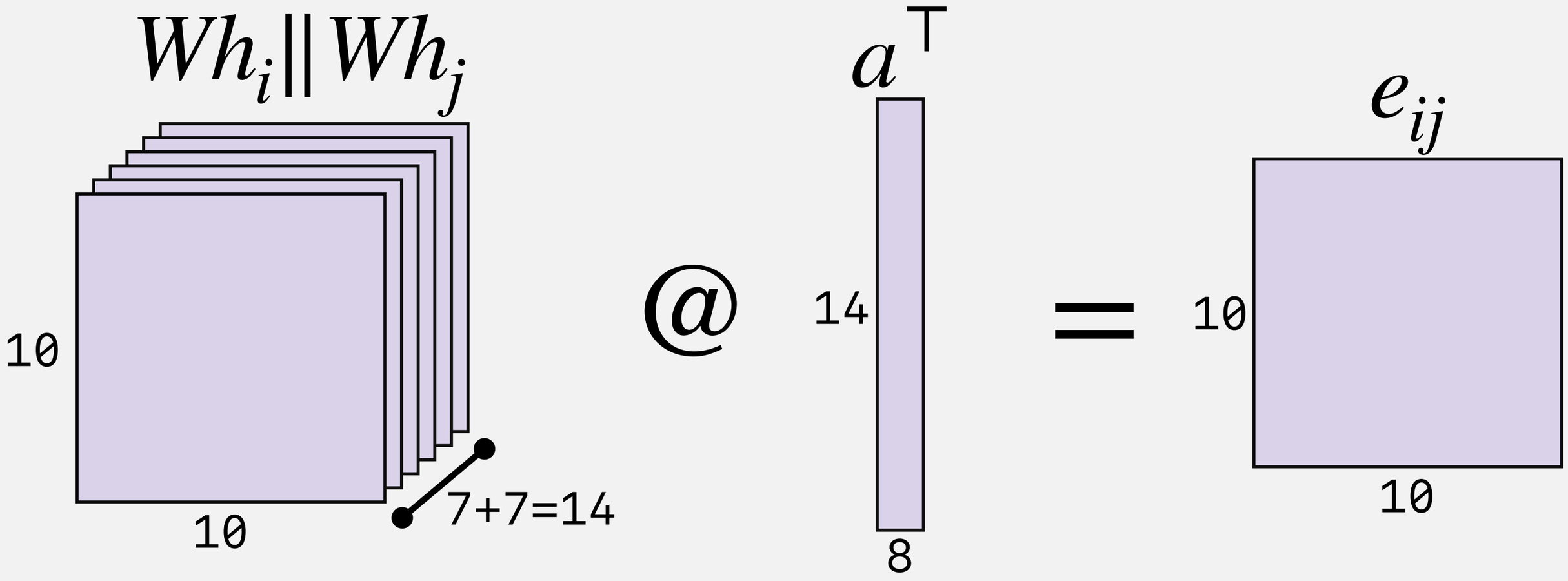
2nd GAT layer, input



youtube.com/@mashaan14

2nd GAT layer, head 1





youtube.com/@mashaan14

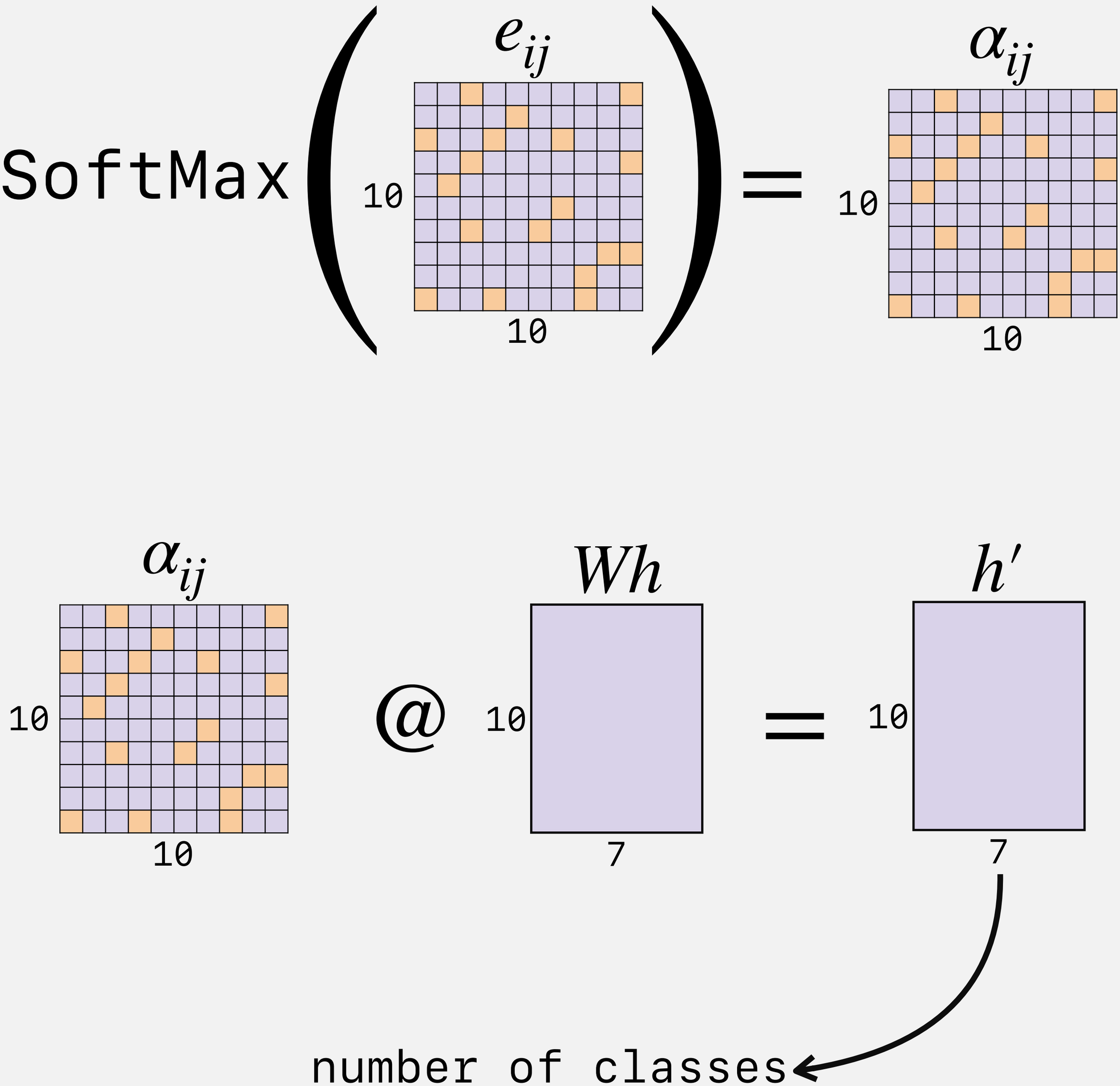
2nd GAT layer, head 1

$$\texttt{LeakyReLU}\left(\begin{array}{c} e_{ij} \\ \boxed{\begin{array}{c}10\end{array}} \\ 10 \end{array}\right)$$

```
# masked attention
torch.where(adj > 0, e, zero_vec)
```

youtube.com/@mashaan14

2nd GAT layer, head 1



youtube.com/@mashaan14