-->

🧠 Transformer Attention — Summary of Formulas, Matrices, and Dimensions

Basic Components

Symbol	Meaning	Typical Shape
T	Target sequence length (decoder)	—
T'	Source sequence length (encoder)	—
d_{model}	Model (embedding) dimension	e.g. 512
d_k	Key / Query dimension per head	e.g. 64
d_v	Value dimension per head	e.g. 64
P	Number of attention heads	e.g. 8

Linear Projections

For each head $p = 1, \ldots, P$:

$$\mathbf{Q}^p = \mathbf{H}\mathbf{W}_Q^p \tag{1}$$
$$\mathbf{K}^p = \mathbf{H}\mathbf{W}_K^p \tag{2}$$
$$\mathbf{V}^p = \mathbf{H}\mathbf{W}_V^p \tag{3}$$

Matrix	Description	Shape
$\mathbf{H}$	Input sequence embeddings	(T',d_{model})
$\mathbf{W}_Q^p$	Query projection matrix	(d_{model},d_k)
$\mathbf{W}_K^p$	Key projection matrix	(d_{model},d_k)
$\mathbf{W}_V^p$	Value projection matrix	(d_{model},d_v)
$\mathbf{Q}^p$	Queries	(T,d_k)
$\mathbf{K}^p$	Keys	(T',d_k)

$\mathbf{V}^p$	Values	(T',d_v)

Scaled Dot-Product Attention

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{M}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V}$$

Term	Meaning	Shape
$\mathbf{Q}\mathbf{K}^\top$	Similarity scores	(T, T')
$\frac{1}{\sqrt{d_k}}$	Scaling factor	scalar
$\mathbf{M}$	Mask matrix (0 or –∞)	(T, T')
Softmax output	Attention weights	(T, T')
Final output	Weighted sum of values	(T,d_v)

Self-Attention vs Cross-Attention

Type	Queries from	Keys/Values from	Formula	Shapes
Self-Attention (Encoder or Decoder)	Same sequence	Same sequence	$\text{Attention}(Q(H_{1:T}), K(H_{1:T}), V(H_{1:T}))$	Q, K, V : $(T,\cdot)$
Cross-Attention (Decoder → Encoder)	Decoder hidden states	Encoder hidden states	$\text{Attention}(Q(h_{1:T}), K(g_{1:T'}), V(g_{1:T'}))$	Q : $(T,\cdot)$, K, V : $(T',\cdot)$

Mask Types

Mask Type	Shape	Description
Causal (left-to-right)	(T, T)	Upper-triangular (prevent attending to future tokens)
Padding mask	(T, T')	Masks out padded tokens in the encoder input
Bidirectional	None / zeros	Allows all tokens to attend to all others

The mask is added **before the softmax**:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)$$

Multi-Head Attention

Each head computes attention independently, then concatenates and projects:

$$\text{MultiHeadAttention}(Q, K, V, M) = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_P\right)\mathbf{W}_O$$

where

$$\text{head}_p = \text{Attention}(\mathbf{Q}^p, \mathbf{K}^p, \mathbf{V}^p, \mathbf{M})$$

Symbol	Meaning	Shape
head_p	Output from head$p$	(T,d_v)
Concatenated heads	All heads combined	(T,$P \cdot d_v$)
$\mathbf{W}_O$	Output projection matrix	($P \cdot d_v$,d_{model})
Final output	Combined attention	(T,d_{model})

Encoder–Decoder Stack Summary

Layer	Input	Attention Type	Mask
Encoder layer	Source embeddings	Self-attention	None (bidirectional)
Decoder layer 1	Target embeddings	Self-attention	Causal mask
Decoder layer 2	Encoder output + decoder states	Cross-attention	Padding mask (for source)

Common Dimension Conventions

Component	Notation	Shape
Embedding matrix	E	(Vocab,d_{model})
Positional encoding	P	(T,d_{model})
Input to encoder	$X + P$	(T',d_{model})
Input to decoder	$Y + P$	(T,d_{model})

Encoder output	$g_{1:T'}$	(T', d_{model})
Decoder output	$h_{1:T}$	(T, d_{model})

Attention Weight Matrix Intuition

Axes	Meaning
Rows = target positions t'	"Which source tokens does this output token look at?"
Columns = source positions t	"How much attention is given to each input token?"

Shape: (T, T')

Overall Data Flow Summary

Encoder (self-attention):

$$\mathbf{g}_{1:T'} = \text{EncoderSelfAttention}(\mathbf{x}_{1:T'})$$

Decoder:

$$\tilde{\mathbf{h}}_{1:T} = \text{DecoderSelfAttention}(\mathbf{y}_{1:T})$$
$$\mathbf{h}_{1:T} = \text{CrossAttention}(\tilde{\mathbf{h}}_{1:T}, \mathbf{g}_{1:T'})$$

Then feed-forward + layer normalization layers complete the block.
