



Before(Original slide)

Core Mechanism 2: Multi-Head Attention

MULTI-HEAD ATTENTION FORMULA
 $\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \cdot W^O$
 where $\text{head}_i = \text{Attention}(Q \cdot W_i^Q, K \cdot W_i^K, V \cdot W_i^V)$

Core Idea

- Project Q, K, and V into **h subspaces**, then compute attention independently
- Each head captures **different semantic relations** across positions
- Concatenate all heads and aggregate them through the W^O projection
- Cost stays comparable: each head uses $1/h$ dimensions while h heads run in parallel

Three Attention Uses

- Encoder self-attention**: models the full input sequence at each encoder layer
- Decoder self-attention**: uses masking to attend only to current and previous outputs
- Encoder-decoder cross-attention**: Q comes from the decoder; K/V come from encoder outputs

h = 8
attention heads

d_model = 512
model dimension

d_k = d_v = 64
per-head dimension (512/8)

4 groups
projection matrices (W^Q, W^K, W^V, W^O)

Target text

5 / 10



B. DeepPresenter (Whole-slide rewrite)

Core Mechanism 2: Multi-Head Attention

Multi-Head Attention Mechanism

- Project Q, K, and V into h subspaces and compute attention independently
- Each head captures different semantic relations across positions
- Concatenate all heads and aggregate them through the W^O projection
- Cost remains comparable: each head uses $1/h$ dimensions while h heads run in parallel

Three Uses of Attention in Transformers

Encoder Self-Attention
 Q, K, and V come from previous-layer outputs and capture source-sequence dependencies

Decoder Self-Attention
 Masked attention attends only to current and previous positions for autoregressive generation

Encoder-Decoder Cross-Attention
 Q comes from the decoder; K and V come from encoder outputs for source-target alignment

h = 8
attention heads

d_k = d_v = 64
per-head dimension (512/8)

8 heads
projection matrices (W^Q, W^K, W^V, W^O)



C. MemSlides (Localized patch revision)

Core Mechanism 2: Multi-Head Attention

MULTI-HEAD ATTENTION FORMULA
 $\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \cdot W^O$
 where $\text{head}_i = \text{Attention}(Q \cdot W_i^Q, K \cdot W_i^K, V \cdot W_i^V)$

Core Idea

- Project Q, K, and V into **h subspaces**, then compute attention independently
- Each head captures **different semantic relations** across positions
- Concatenate all heads and aggregate them through the W^O projection
- Cost stays comparable: each head uses $1/h$ dimensions while h heads run in parallel

Three Attention Uses

- Encoder self-attention**: models the full input sequence at each encoder layer
- Decoder self-attention**: uses masking to attend only to current and previous outputs
- Encoder-decoder cross-attention**: Q comes from the decoder; K/V come from encoder outputs

h = 8
attention heads

d_model = 512
model dimension

d_k = d_v = 64
per-head dimension (512/8)

8 heads
projection matrices (W^Q, W^K, W^V, W^O)

Update target

5 / 10



User feedback : change "4 groups" at the bottom of page 5 to "8 heads."