Attention Mechanism and Transformers

Agenda

- Transformers Quiz
- Introduction to Transformers
- Positional Embedding
- Multi-head Attention
- Masking
- Encoder-Decoder Attention
- Transformer Architectures

Let’s begin the discussion by answering a few questions on Attention mechanism and Transformers

Transformers Quiz

Considering the transformer architecture proposed In the original paper (Attention Is All You Need, 2017), which of the following statements are true?

A A transformer is a neural network

B The encoder stage outputs a latent representation of the input

C The decoder stage outputs a sequence of tokens based on the latent representation from the encoder stage

D The encoder and decoder stages always consist of one encoder and decoder block only

Transformers Quiz

Considering the transformer architecture proposed In the original paper (Attention Is All You Need, 2017), which of the following statements are true?

A — A transformer is a neural network

B — The encoder stage outputs a latent representation of the input

C — The decoder stage outputs a sequence of tokens based on the latent representation from the encoder stage

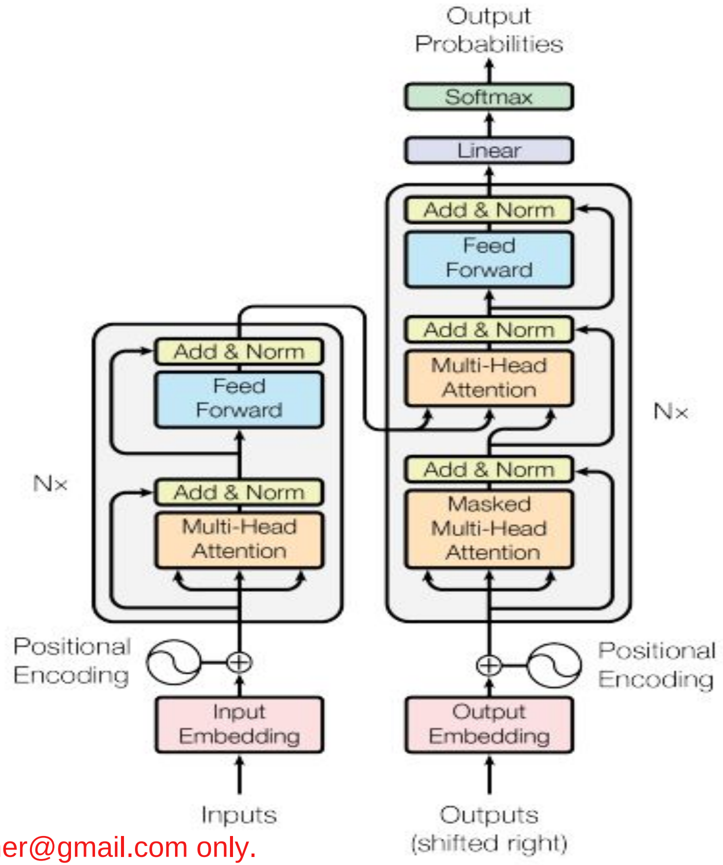
D — The encoder and decoder stages always consist of one encoder and decoder block only

Transformers

Transformers are a **type of neural network architecture**

Transformers were **introduced** in a paper by **Vaswani et al. in 2017**

Transformers are based on the idea of **self-attention**

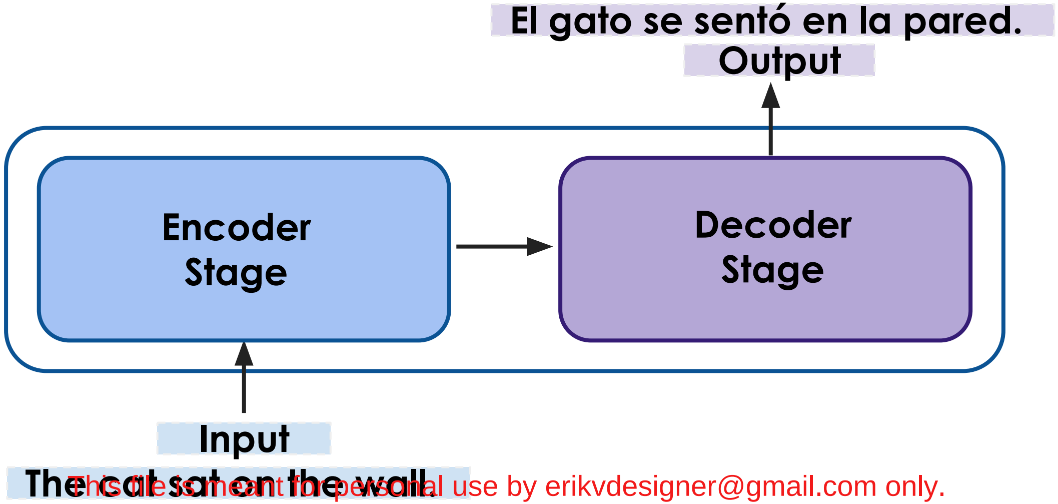


Source: Image from the original research paper Attention Is All You Need

Transformers - Working

The **encoder** takes in a sequence of tokens (e.g. words or characters) and outputs a **latent representation**

The **decoder** then takes this latent representation as input and outputs a **sequence of tokens**



El gato se sentó en la pared.

Output

Encoder Stage

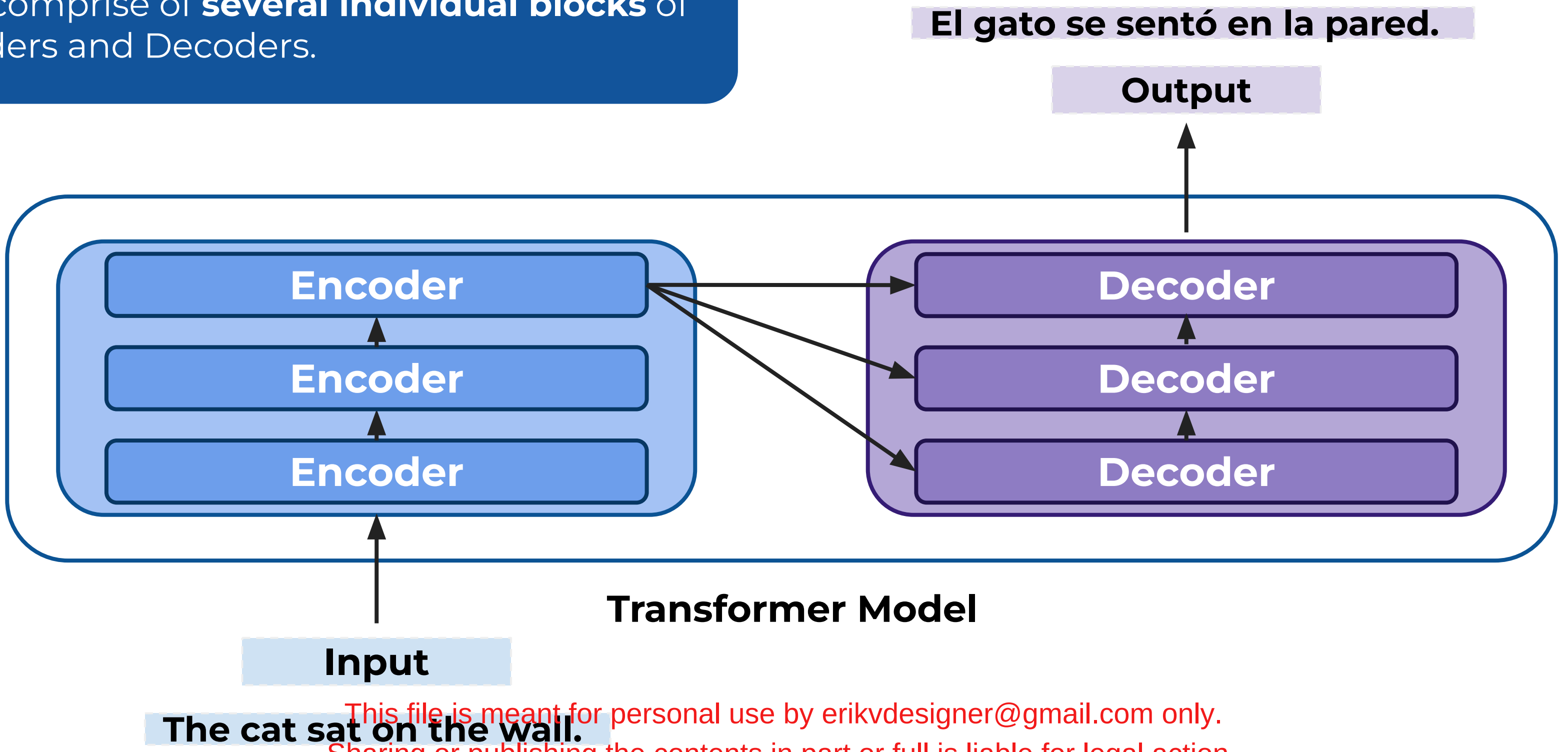
Decoder Stage

Input

The cat sat on the wall.

Transformers - Working

In reality, the **Encoder** and **Decoder** stage each comprise of **several individual blocks** of Encoders and Decoders.



El gato se sentó en la pared.

Output

Encoder

Encoder

Encoder

Decoder

Decoder

Decoder

Transformer Model

Input

The cat sat on the wall.

Transformers Quiz

Which of the following best describes the purpose of positional encoding in transformer ?

A — It introduces randomness to word embeddings to enhance model generalization.

B — It helps the model differentiate between words with similar semantic meanings.

C — It provides the model with information about the order of words in a sequence.

D — It reduces the computational complexity of the self-attention mechanism.

Transformers Quiz

Which of the following best describes the purpose of positional encoding in transformer ?

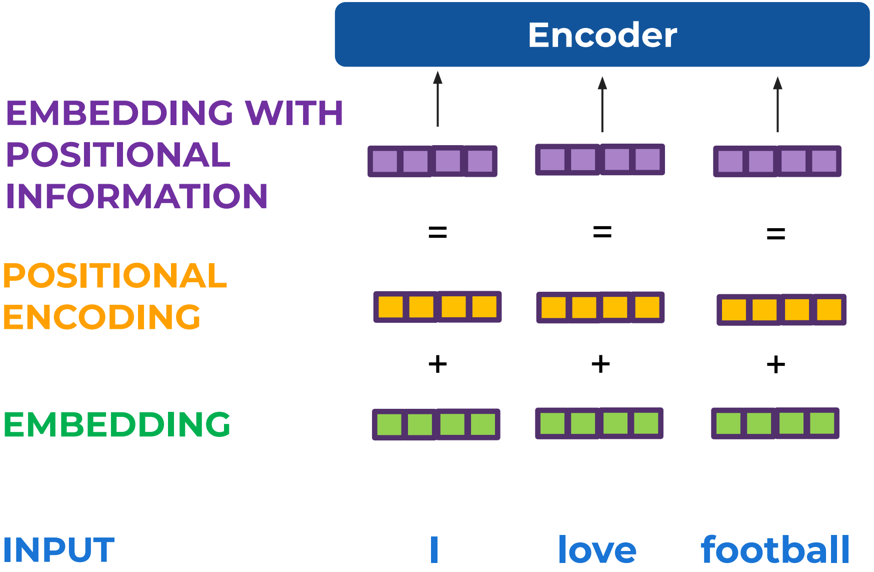
A It introduces randomness to word embeddings to enhance model generalization.

B It helps the model differentiate between words with similar semantic meanings.

C It provides the model with information about the order of words in a sequence.

D It reduces the computational complexity of the self-attention mechanism.

Positional Encoding



Encoder

EMBEDDING WITH POSITIONAL INFORMATION

=

POSITIONAL ENCODING

+

EMBEDDING

INPUT

I love football

Positional Encoding is a way to account for **the order of the words in the input sequence.**

Positional Encoding is a vector added to each input embedding.

These vectors follow a specific pattern that the model learns, which helps it determine the position of each word, or the distance between different words in the sequence.

Transformers Quiz

Which of the following best describes the purpose of self-attention mechanism in transformer?

A It helps the model focus on relevant parts of the input sequence when making predictions.

B It reduces the computational complexity of the neural network.

C It enables the model to generate synthetic data for training purposes.

D It increases the model's ability to generalize to unseen data.

Transformers Quiz

Great Learning
POWER AHEAD

Which of the following best describes the purpose of self-attention mechanism in transformer?

A 🟩 It helps the model focus on relevant parts of the input sequence when making predictions.

B It reduces the computational complexity of the neural network.

C It enables the model to generate synthetic data for training purposes.

D It increases the model's ability to generalize to unseen data.

Self-Attention Mechanism

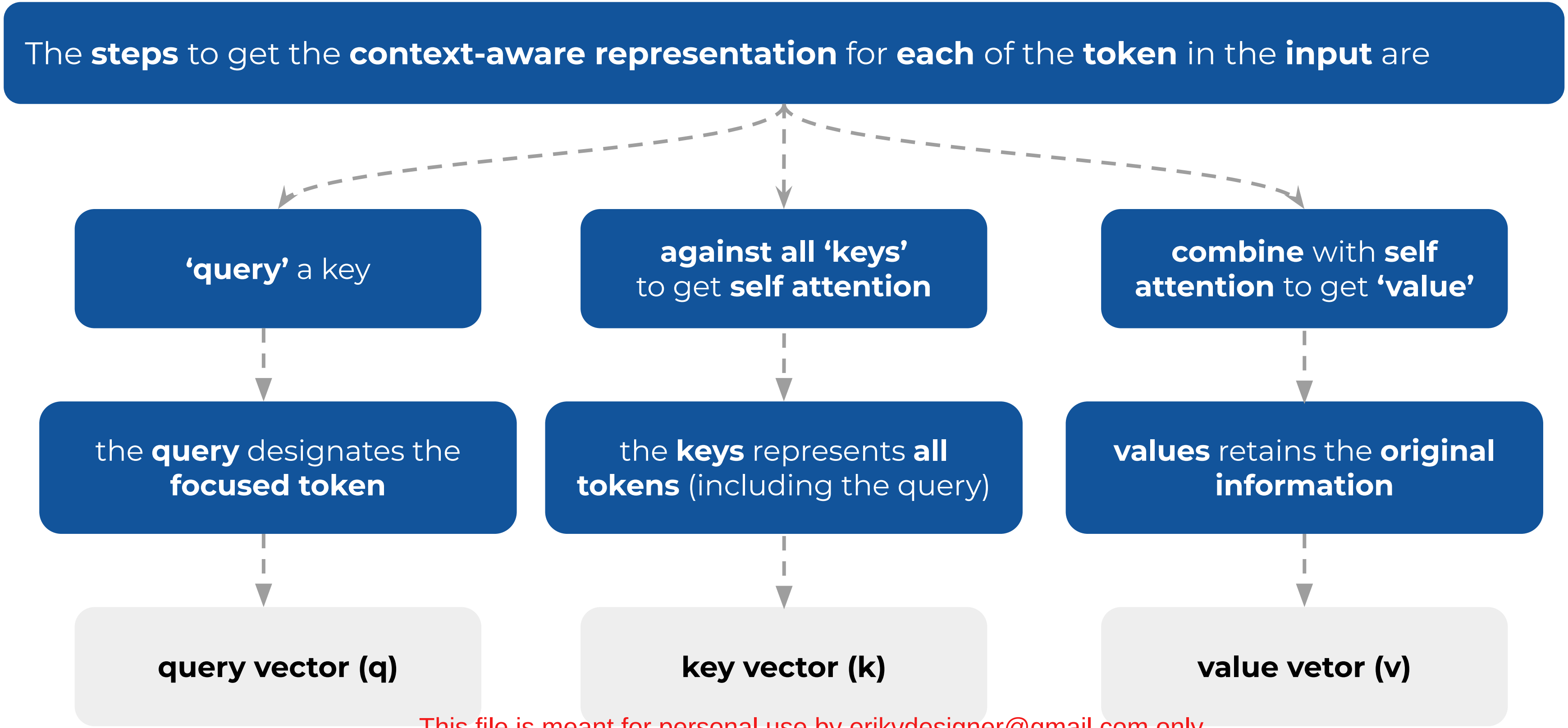
The **self-attention mechanism** lies at the **core** of **transformer models**

Self attention allows us to generate a **context-aware representation** of **each token** in the input

The **context-aware representation** of **each token** is generated with respect to all other tokens in the input

The context-aware representation **focuses** on the **relevant parts of the input** for a given task

Self-Attention - Computation



The **steps** to get the **context-aware representation** for **each** of the **token** in the **input** are

‘query’ a key

the **query** designates the **focused token**

query vector (q)

against all ‘keys’ to get **self attention**

the **keys** represents **all tokens** (including the query)

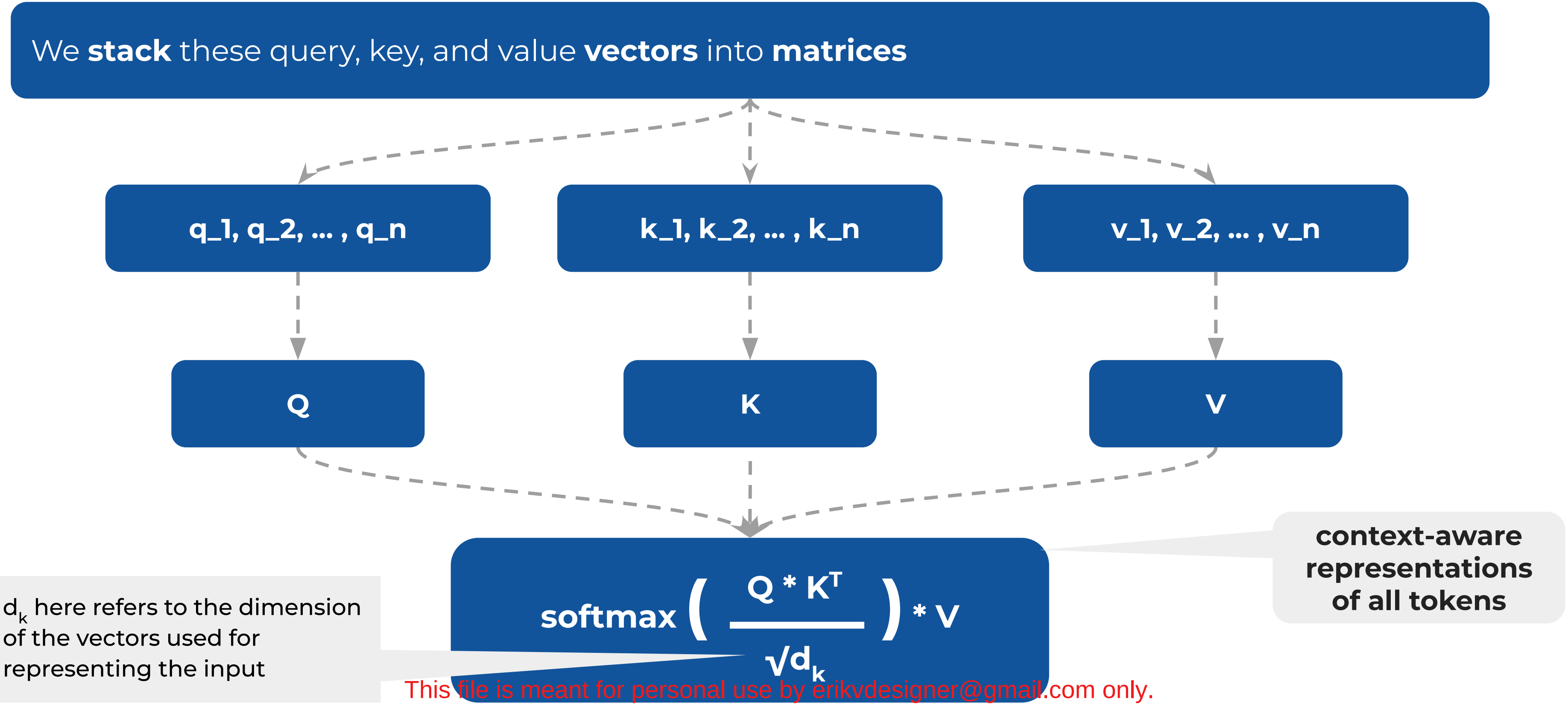
key vector (k)

combine with **self attention** to get **‘value’**

values retains the **original information**

value vetor (v)

Self-Attention - Computation



We **stack** these query, key, and value **vectors** into **matrices**

q_1, q_2, … , q_n

k_1, k_2, … , k_n

v_1, v_2, … , v_n

Q

K

V

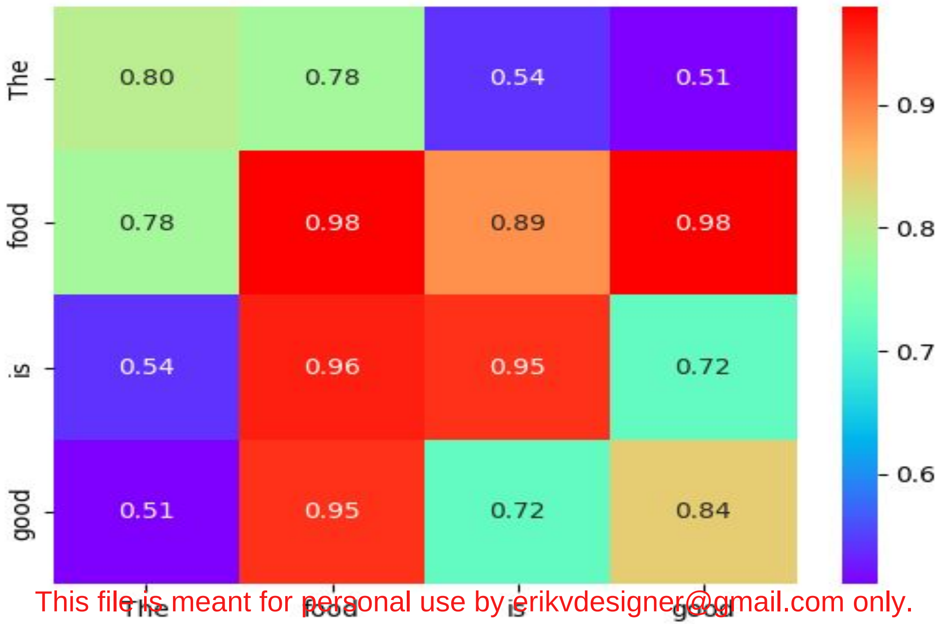
$$\text{softmax}\left(\frac{Q * K^T}{\sqrt{d_k}}\right) * V$$

d_k here refers to the dimension of the vectors used for representing the input

context-aware representations of all tokens

Self-Attention - Example

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$



Transformers Quiz

In a transformer model with multi-head attention, if there are 8 attention heads and each head has a dimensionality of 64, what is the dimension of the multi-head attention output after concatenation?

A 64

B 128

C 256

D 512

Transformers Quiz

In a transformer model with multi-head attention, if there are 8 attention heads and each head has a dimensionality of 64, what is the dimension of the multi-head attention output after concatenation?

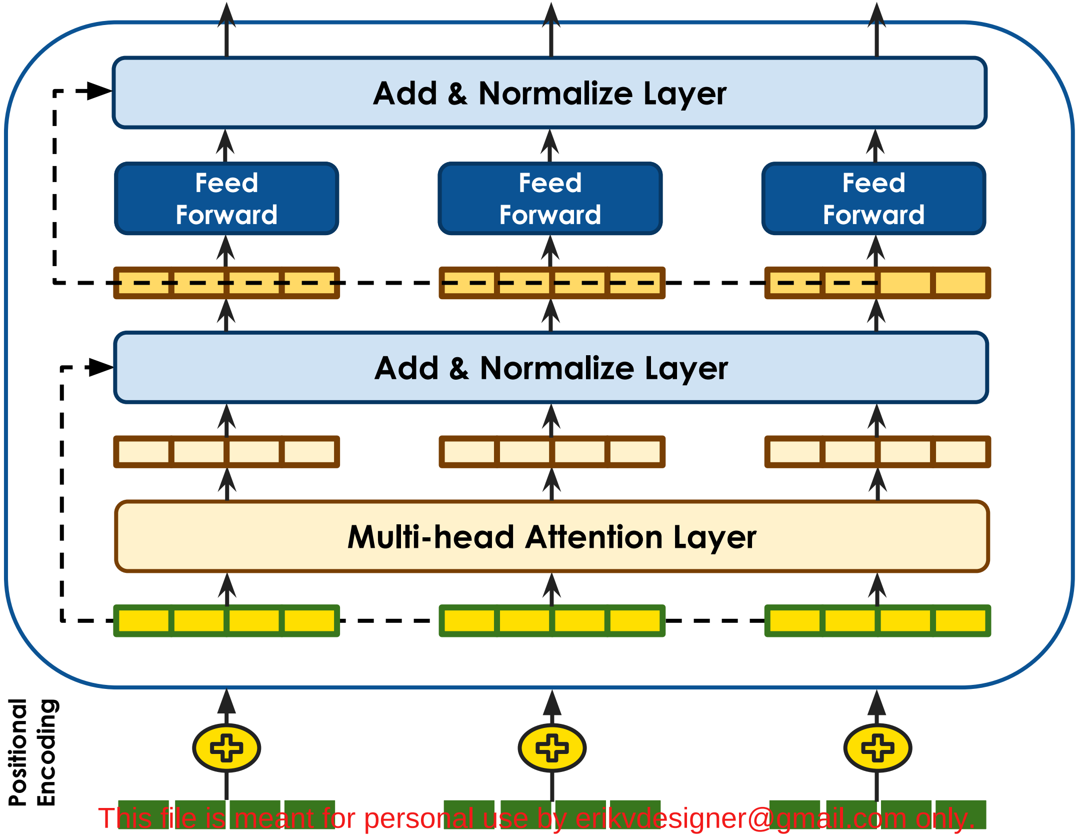
A 64

B 128

C 256

[✓ D 512]

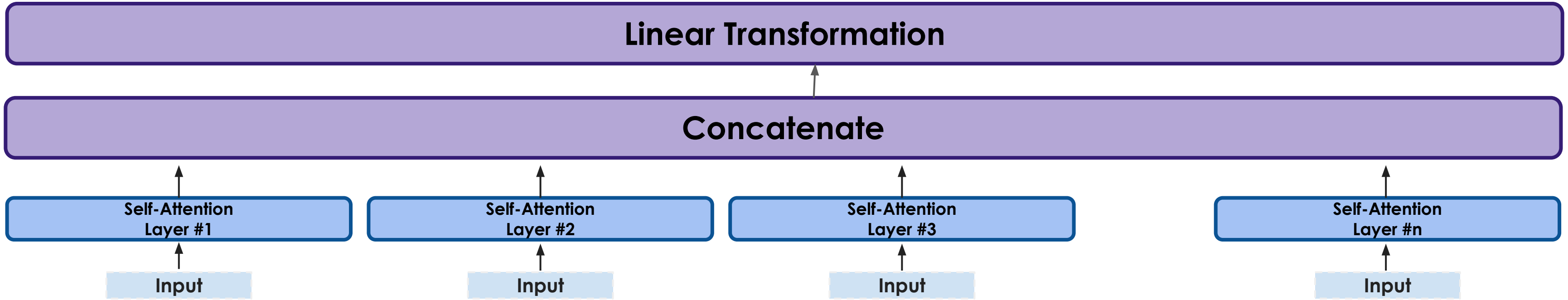
Transformer Architecture - Encoder Block



Multi-Head Attention

The output of each self-attention layer is taken and concatenated

The linear transformation layer is merely a fully-connected layer of neurons



Each head produces a 64 dimensional vector. Since there are 8 such heads, after concatenation, it will result into a 8*64 = 512 dimensional vector.

Transformers Quiz

Why is masking necessary in the decoder of a Transformer model for sequence-to-sequence tasks?

A To limit the decoder's access to information from the future positions

B To enhance the attention mechanisms focus on relevant information

C To prevent overfitting during training

D To increase the model's capacity to handle longer sequences

Transformers Quiz

Why is masking necessary in the decoder of a Transformer model for sequence-to-sequence tasks?

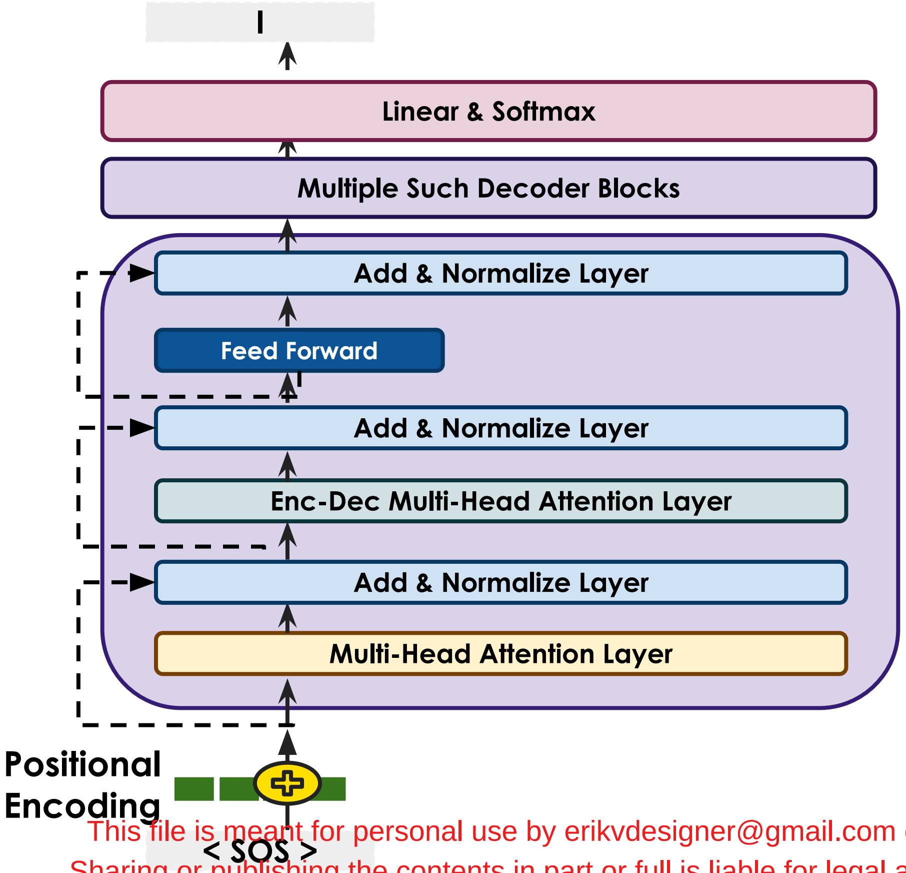
[✓ A To limit the decoder's access to information from the future positions]

B To enhance the attention mechanisms focus on relevant information

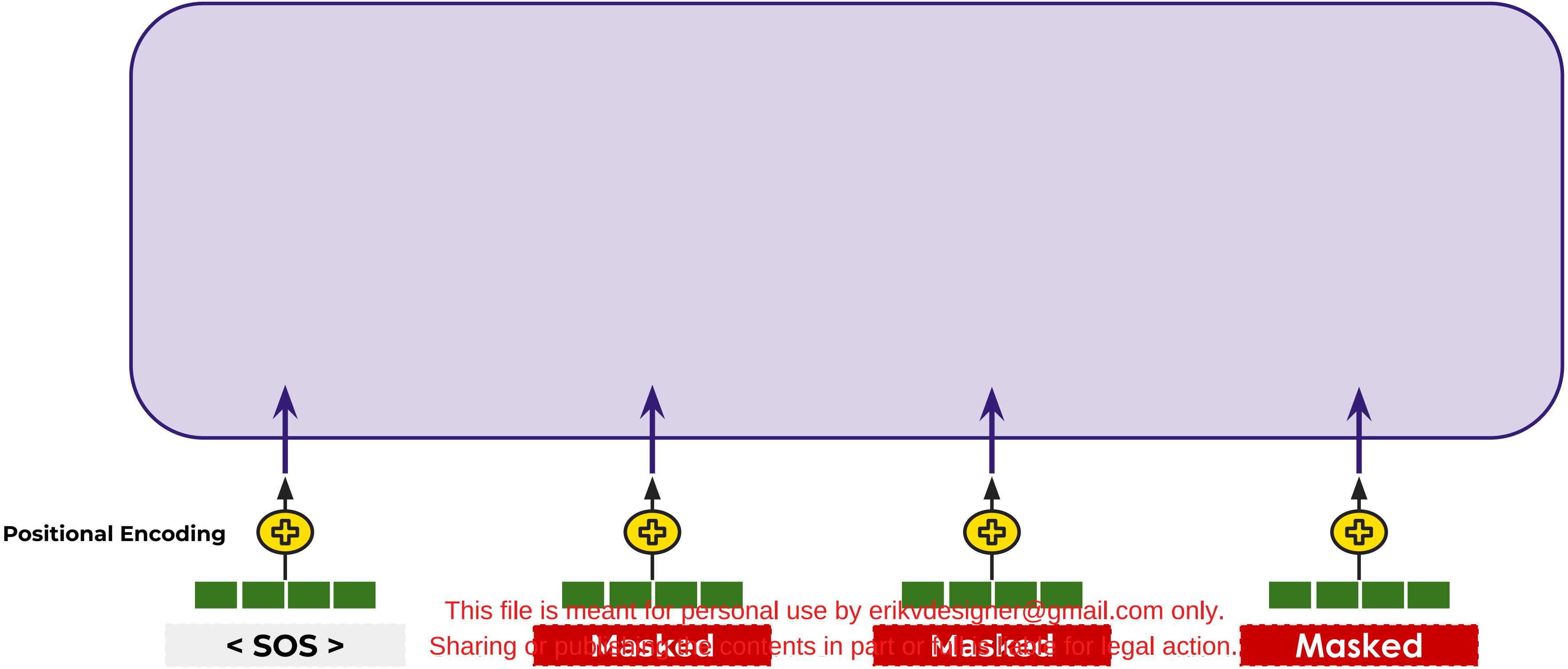
C To prevent overfitting during training

D To increase the model's capacity to handle longer sequences

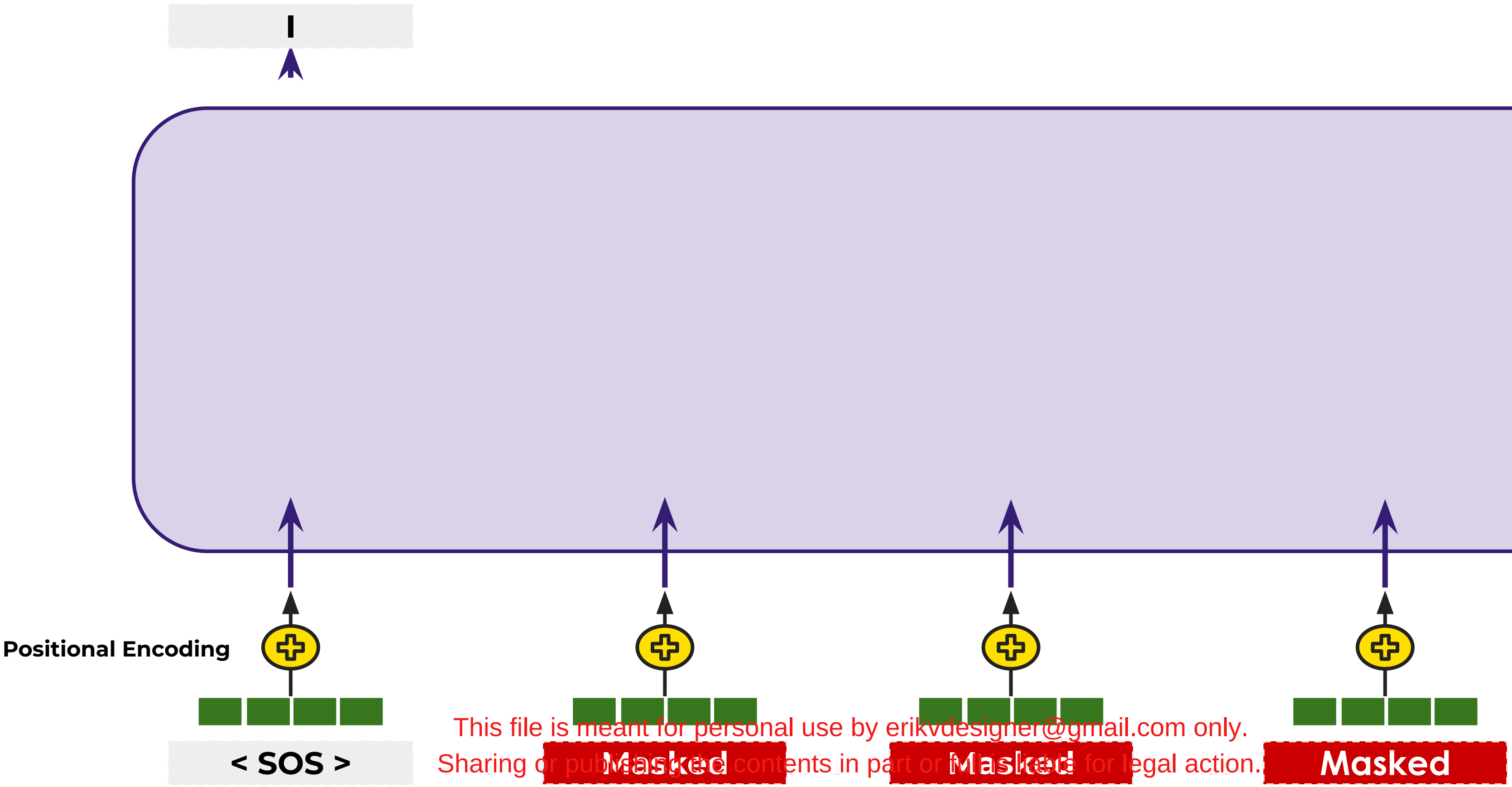
Transformer Architecture - Decoder Block



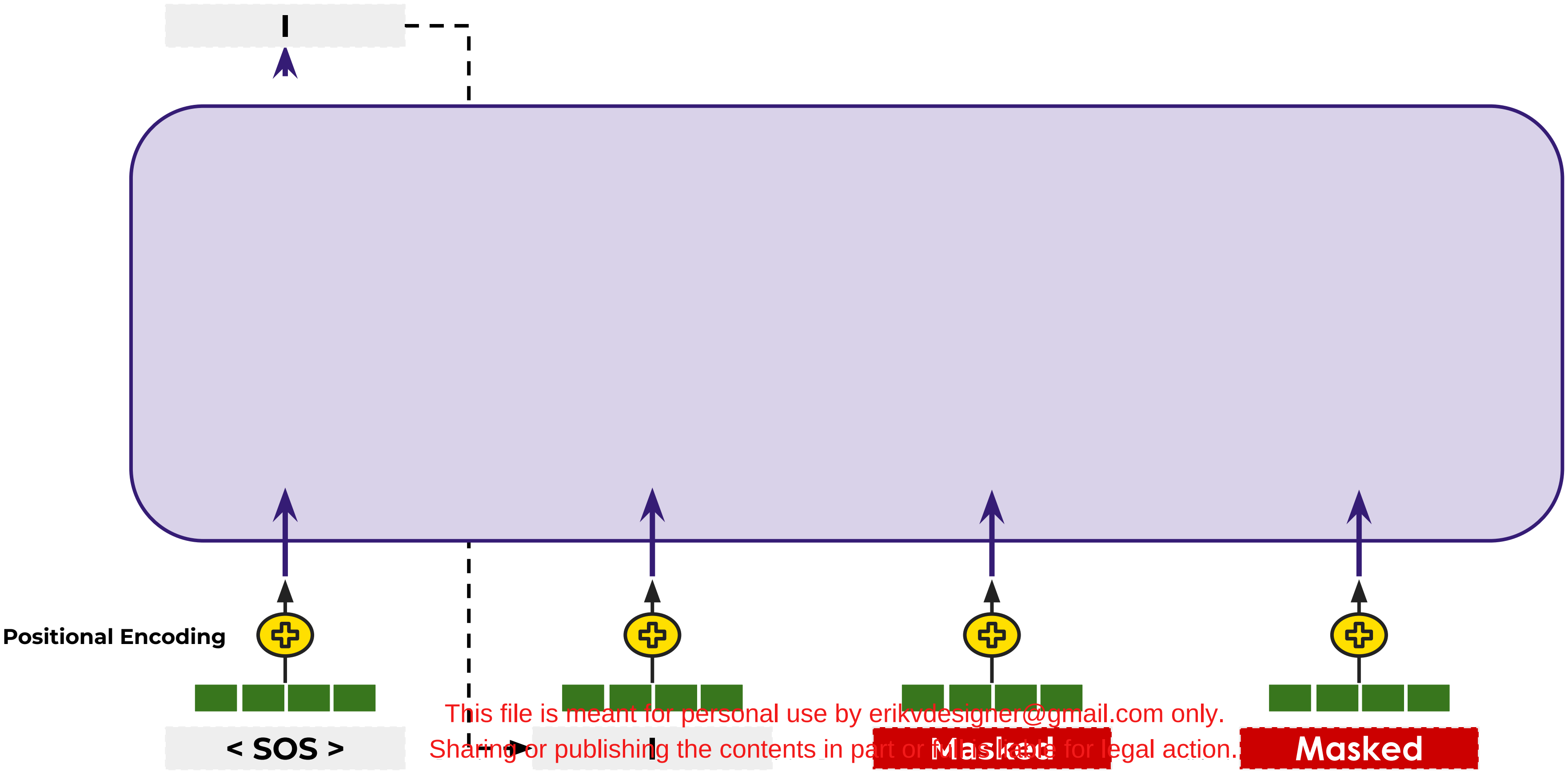
Decoder - Masking



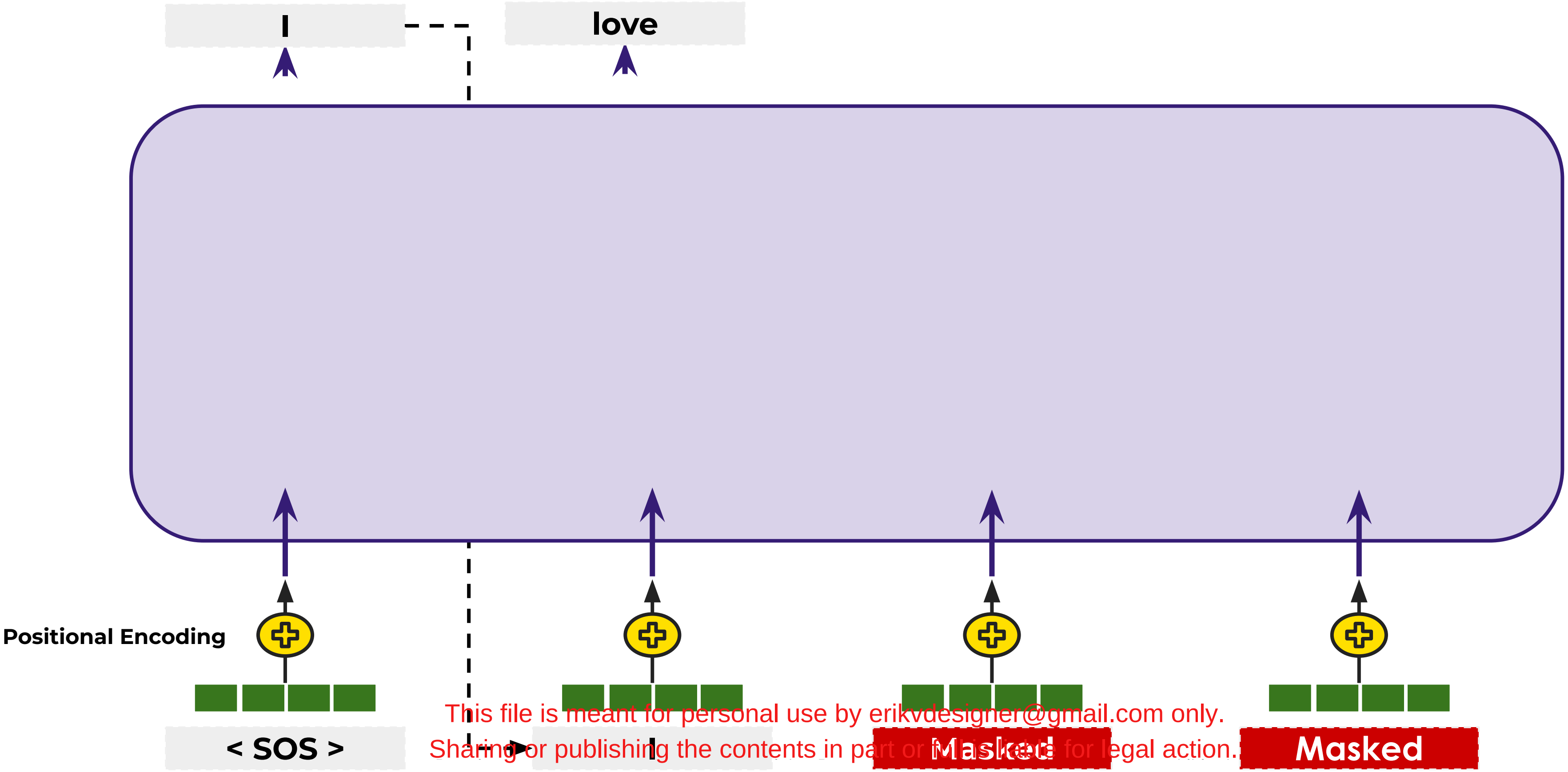
Decoder - Masking



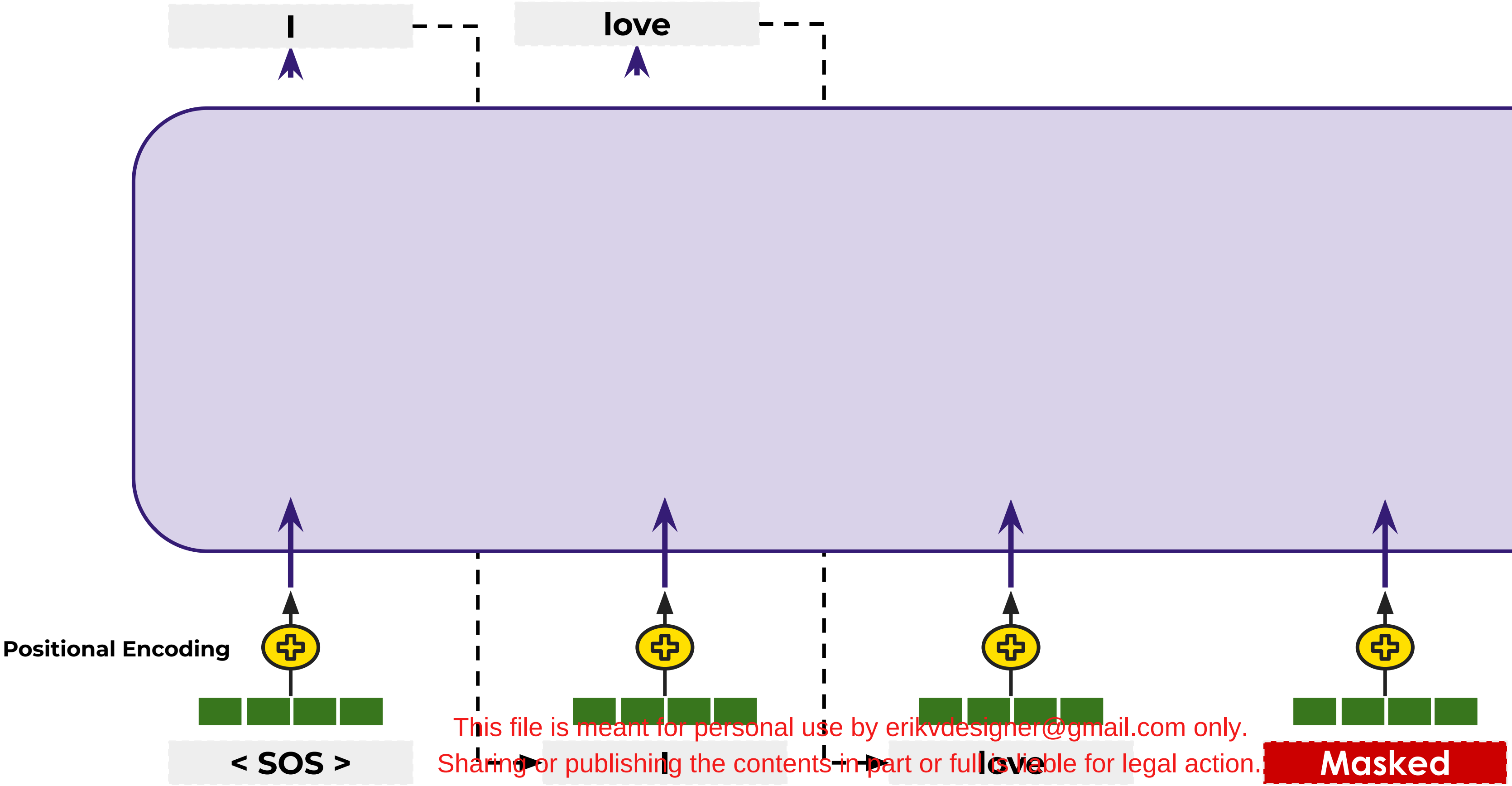
Decoder - Masking



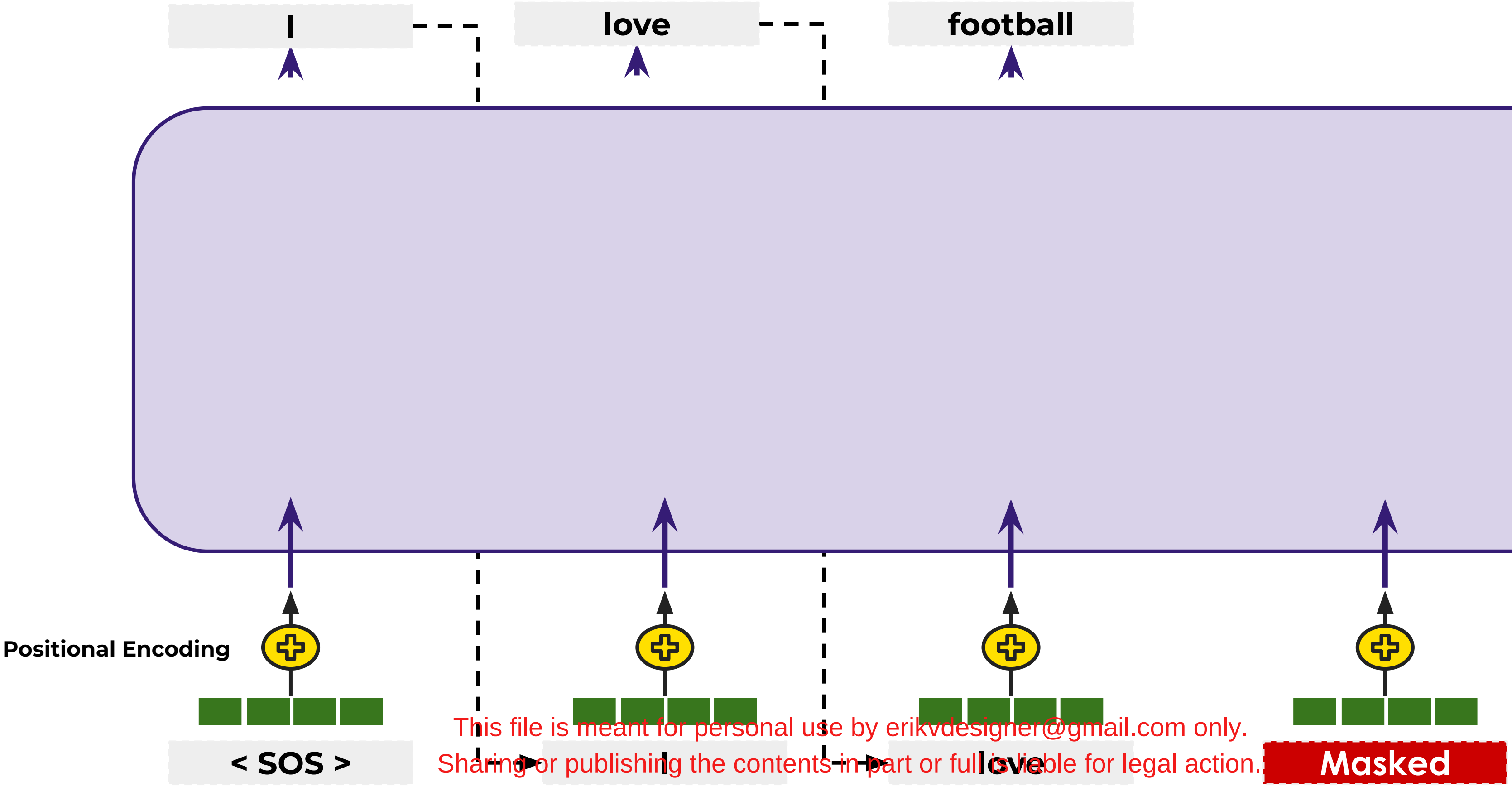
Decoder - Masking



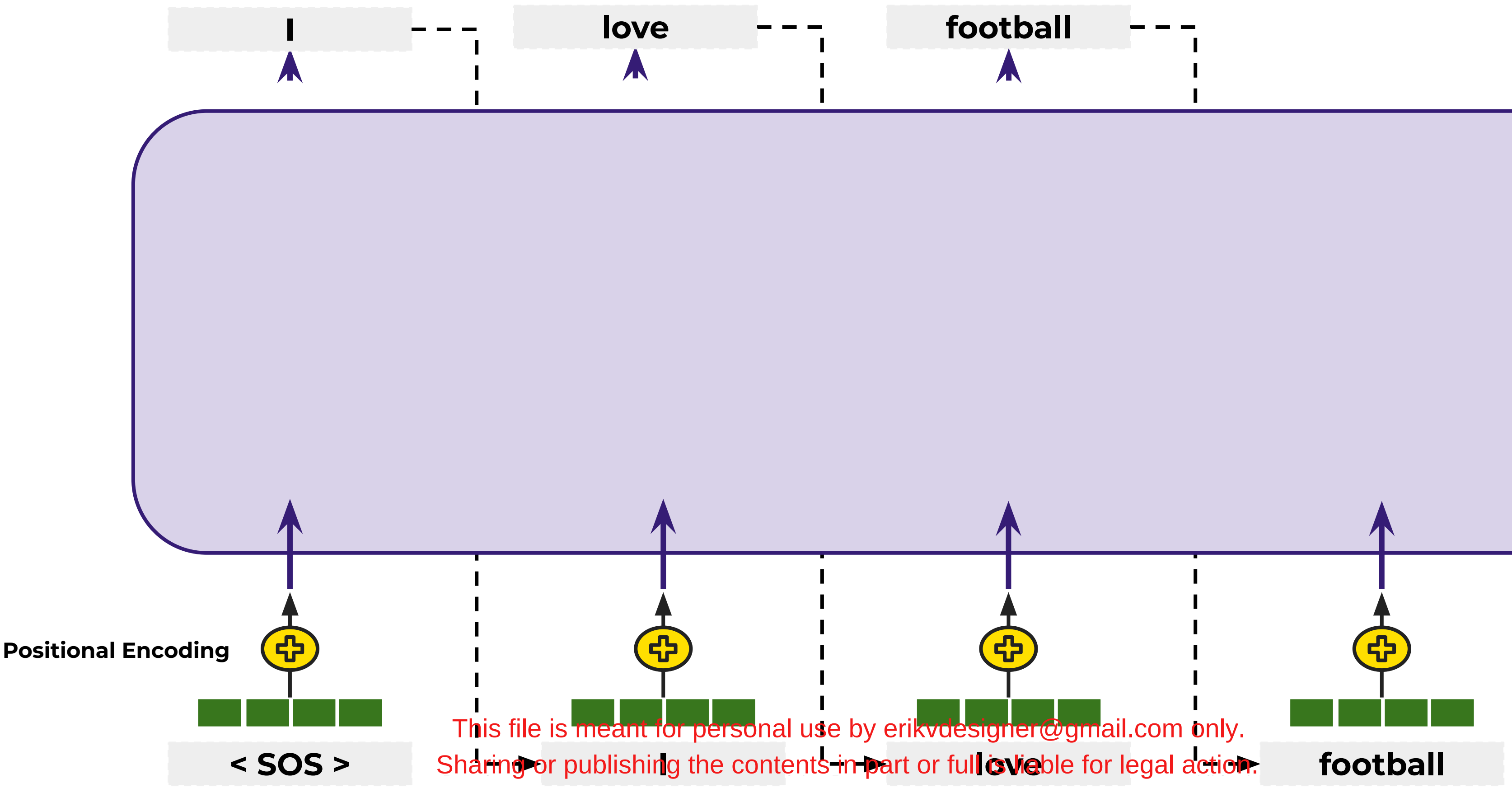
Decoder - Masking



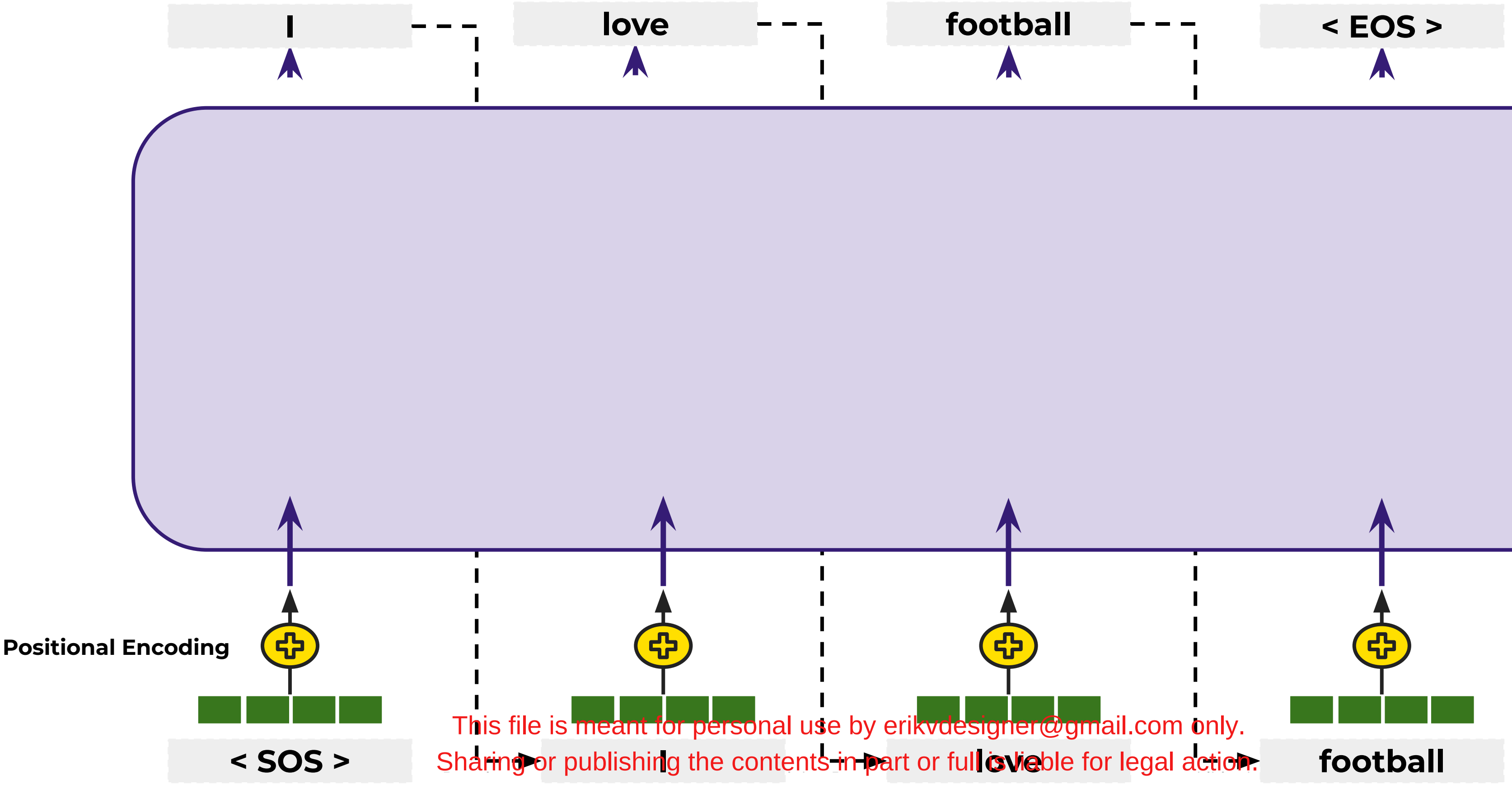
Decoder - Masking



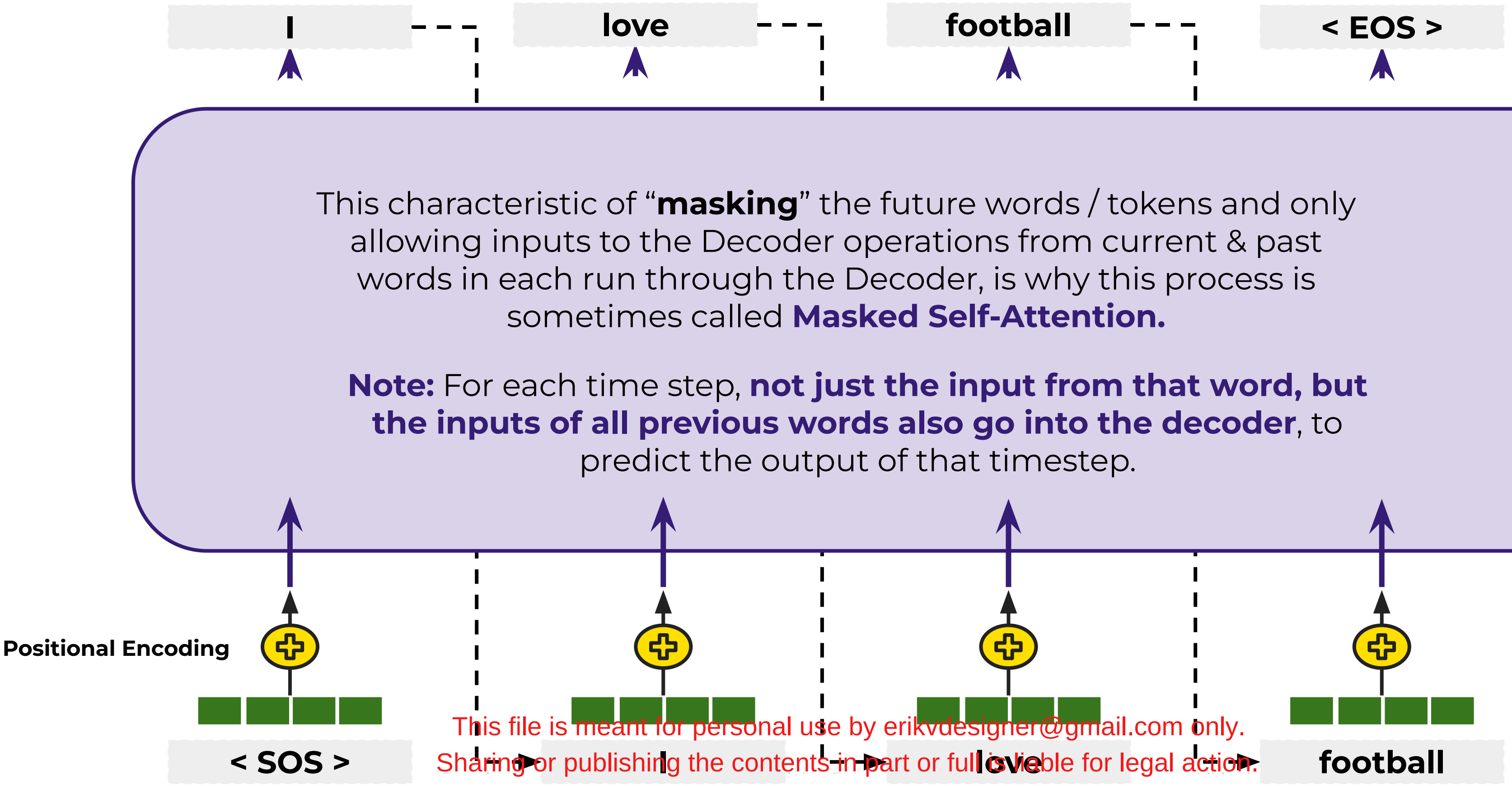
Decoder - Masking



Decoder - Masking



Decoder - Masking



This characteristic of “**masking**” the future words / tokens and only allowing inputs to the Decoder operations from current & past words in each run through the Decoder, is why this process is sometimes called **Masked Self-Attention.**

Note: For each time step, **not just the input from that word, but the inputs of all previous words also go into the decoder**, to predict the output of that timestep.

Transformers Quiz

Which of the following accurately describes the origin of the query, key, and value in encoder-decoder attention mechanisms?

A The query, key, and value are all derived solely from the encoder

B The query is derived from the decoder, while the key and value are from the encoder

C The query and value are derived from the encoder, while the key is from the decoder

D The query, key, and value are all derived from the decoder

Transformers Quiz

Which of the following accurately describes the origin of the query, key, and value in encoder-decoder attention mechanisms?

A The query, key, and value are all derived solely from the encoder

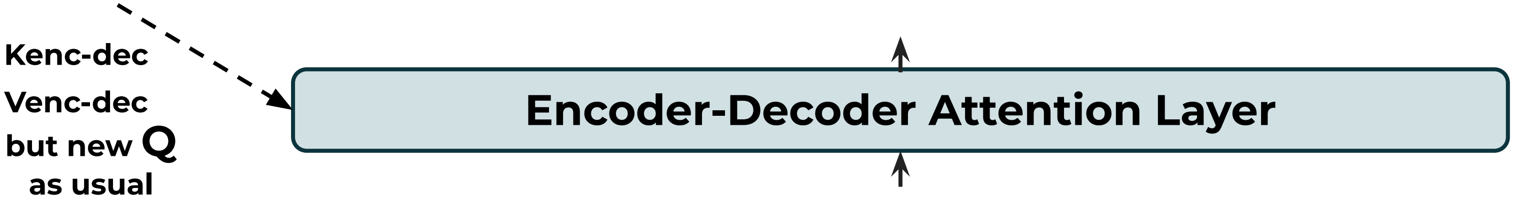
[✓ B] The query is derived from the decoder, while the key and value are from the encoder

C The query and value are derived from the encoder, while the key is from the decoder

D The query, key, and value are all derived from the decoder

The Encoder-Decoder Attention Layer

The difference from normal Self-Attention is that in this layer, **the K and V vectors are not generated from the input embeddings to this layer**, the way they were in the normal Self-Attention layer.



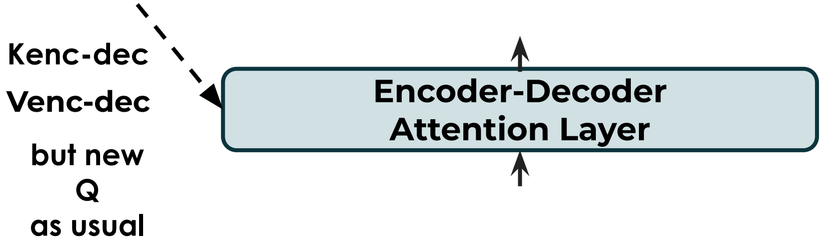
In fact, we utilize a **K encoder-decoder (K enc-dec)** and a **V encoder-decoder (V enc-dec)** in this layer, whose source is from the **final output of the Encoder stage**.

The Encoder-Decoder Attention Layer

We directly utilize **the final embedding vectors** generated at the end of the Encoder stage, and multiply those with weight matrices to get **K enc-dec** & **V enc-dec**. These get used as K and V in this Encoder-Decoder Attention Layer.

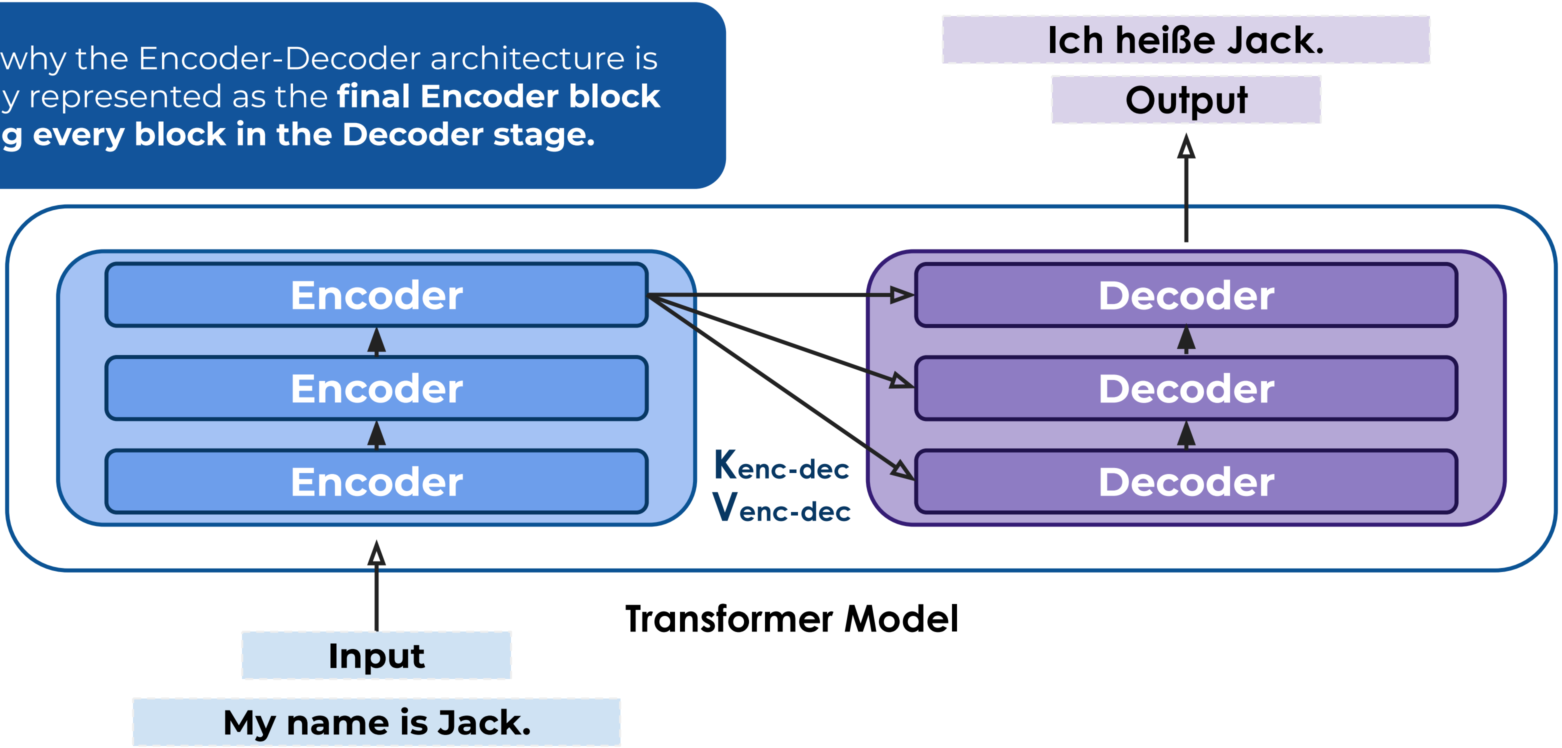
It is only the Q vector that this layer creates from the input to it, the way that normally happens in the Self-Attention Layer (where all three of K, Q & V are directly created from the input embeddings to the layer).

It is also important to mention, that the **Q for < SOS >** (Dec Pos 0) for example, **only relies on the K enc-dec & V enc-dec of the word "I"** (Enc Pos 1) from the input, to predict the word "Ich". This happens for every Decoder word.



The Encoder-Decoder Attention Layer

This is why the Encoder-Decoder architecture is actually represented as the **final Encoder block feeding every block in the Decoder stage.**



The arrows from the final Encoder block to each Decoder block represent the **K enc-dec & V enc-dec from the final Encoder layer being used in the Encoder-Decoder Attention Layer of each Decoder block** in the Decoder stage.

Transformers Quiz

Which of the following best describes BERT’s architecture ?

A BERT is a decoder-only model

B BERT is an encoder-only model

C BERT consists of both encoder and decoder layers

Transformers Quiz

Which of the following best describes BERT's architecture ?

A BERT is a decoder-only model

B BERT is an encoder-only model

C BERT consists of both encoder and decoder layers

Different Transformer Architectures

There are broadly three-types of transformer models today, based on their usage of Encoder and Decoder blocks

Encoder-Decoder

Encoder-only

Decoder-only

Encoder-Decoder Transformer

There are broadly three-types of transformer models today, based on their usage of Encoder and Decoder blocks

Encoder-Decoder

Encoder-only

Decoder-only

Utilize the Encoder and Decoder blocks in tandem, similar to the original transformer architecture

Typically used in tasks where the output heavily relies on the input, like **Machine Translation** and **Text Summarization**

Examples: **T5** and **FLAN-T5**

Encoder-only Transformer

There are broadly three-types of transformer models today, based on their usage of Encoder and Decoder blocks

Encoder-Decoder

Encoder-only

Decoder-only

Utilize only Encoder blocks to generate continuous embeddings from the input

Typically used in discriminative tasks that require embeddings, like for **Text Classification** and **Semantic Search**

Examples: **BERT** and **DistilBERT**

Decoder-only Transformer

There are broadly three-types of transformer models today, based on their usage of Encoder and Decoder blocks

Encoder-Decoder

Encoder-only

Decoder-only

Utilize only Decoder blocks to auto-regressively predict* the next token based on the input

Typically used in generative tasks like **Sentence Completion** and **Question-Answering**

Examples: **GPT** and **Llama**

* Autoregressive prediction involves predicting future values based on past values

Great
Learning

Happy Learning !