What the hell is actually an LLM

Adrian Vecina Tercero

Who Am I?

- First Year PhD, currently researching Robotic Manipulation and Machine learning applications.
- Obsessed with Robots
- Hackathon Enjoyer (11 so Far!)
- Really really like yapping about stuff

Quick Overview.

01 | Intro: What is an LLM?

02 | A little history lesson

03 | The Transformer Model

04 | Modern Innovations

05 | The Deepseek Drama

06 | Running Models Locally

01

Intro: What is a Large Language Model?

LLMS are AI systems trained on MASSIVE amounts of text data. These models learn statistical patterns in language to generate human like text

Large Language Models are...

- A category of **Foundation Model, trained on massive internet-scale data.**
- Most LLMs use a **Transformer Architecture** which leverages **Self-Attention** to analyze relationships in data.
- The LLMs we know and love are **Generative Models**, which takes a sequence of **Tokens,** and predicts the next best word
- Really useful!

02

A little history lesson

Some Historical Context

Pre-2015, researchers looking into machine translation were already experimenting with using neural networks to process text, typically in a RNN-Encoder-Decoder Model.

However, there was one large bottleneck: text was encoded in a fixed-length vector, where the decoder would have limited access to the information provided by the input.

The longer the sequence, the worse the bottleneck as the dimensionality of their representation would be forced to be the same as for shorter or simpler sequences.

Some Historical Context

In 2014 Bahdanau et al address the bottleneck problem via a paper introducing a method allowing the decoder to *"Search through the input as it decoded"*

respect to the previous hidden state s_{i-1} in deciding the next state s_i and generating y_i. Intuitively, this implements a mechanism of attention in the decoder. The decoder decides parts of the source sentence to pay attention to. By letting the decoder have an attention mechanism, we relieve the encoder from the burden of having to encode all information in the source sentence into a fixed-length vector. With this new approach the information can be spread throughout the sequence of

NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal

ABSTRACT

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder–decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.

Bahdanau, Dzmitry & Cho, Kyunghyun & Bengio, Y.. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. ArXiv. 1409.

The **bank** of the river

The frog didn't go to the Hackathon because it was too tired.

Some Historical Context

In 2017, Google Released the following paper. It introduced and discussed the Transformer Model.

"What if we just focused on attention?"

Transformers are still Encoder Decoders!

Vaswani et al. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Some Historical Context

In June 2018, only a year later, GPT-1 developed by openAI is shown with better abilities to make more coherent text.

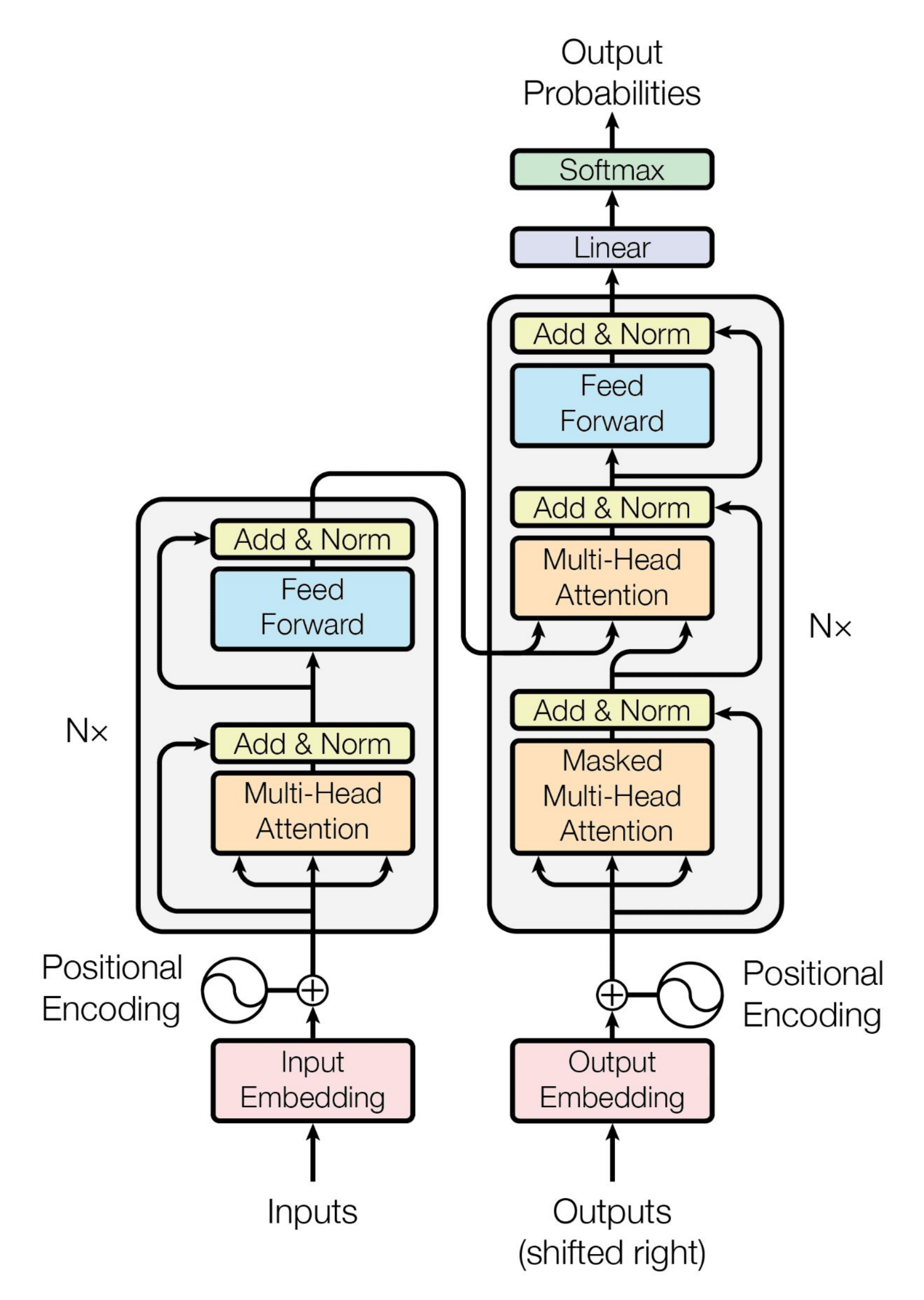
The AI explosion happens in 2020, as GPT-3, one of the most impactful models, is launched.

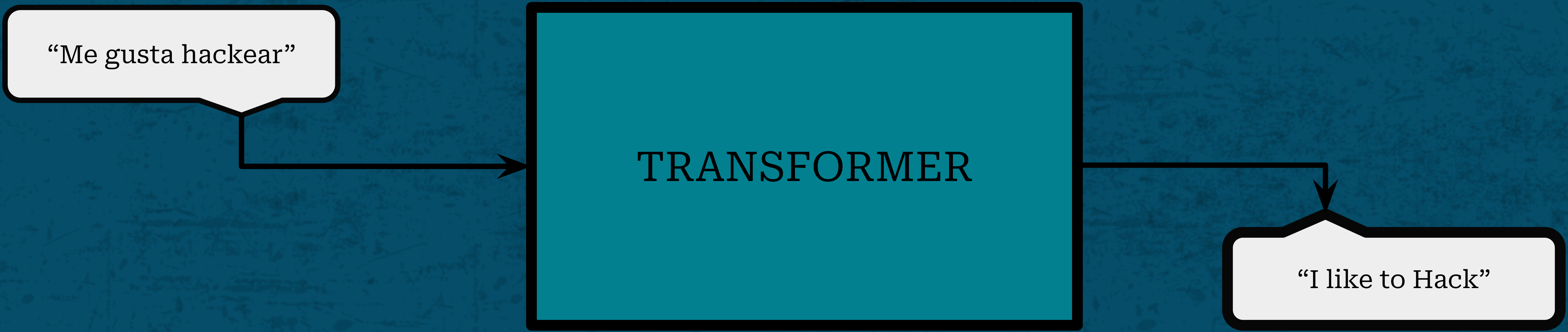
This leads to lots of LLM innovation, and the biggest tech companies in the world all racing to make the best model...

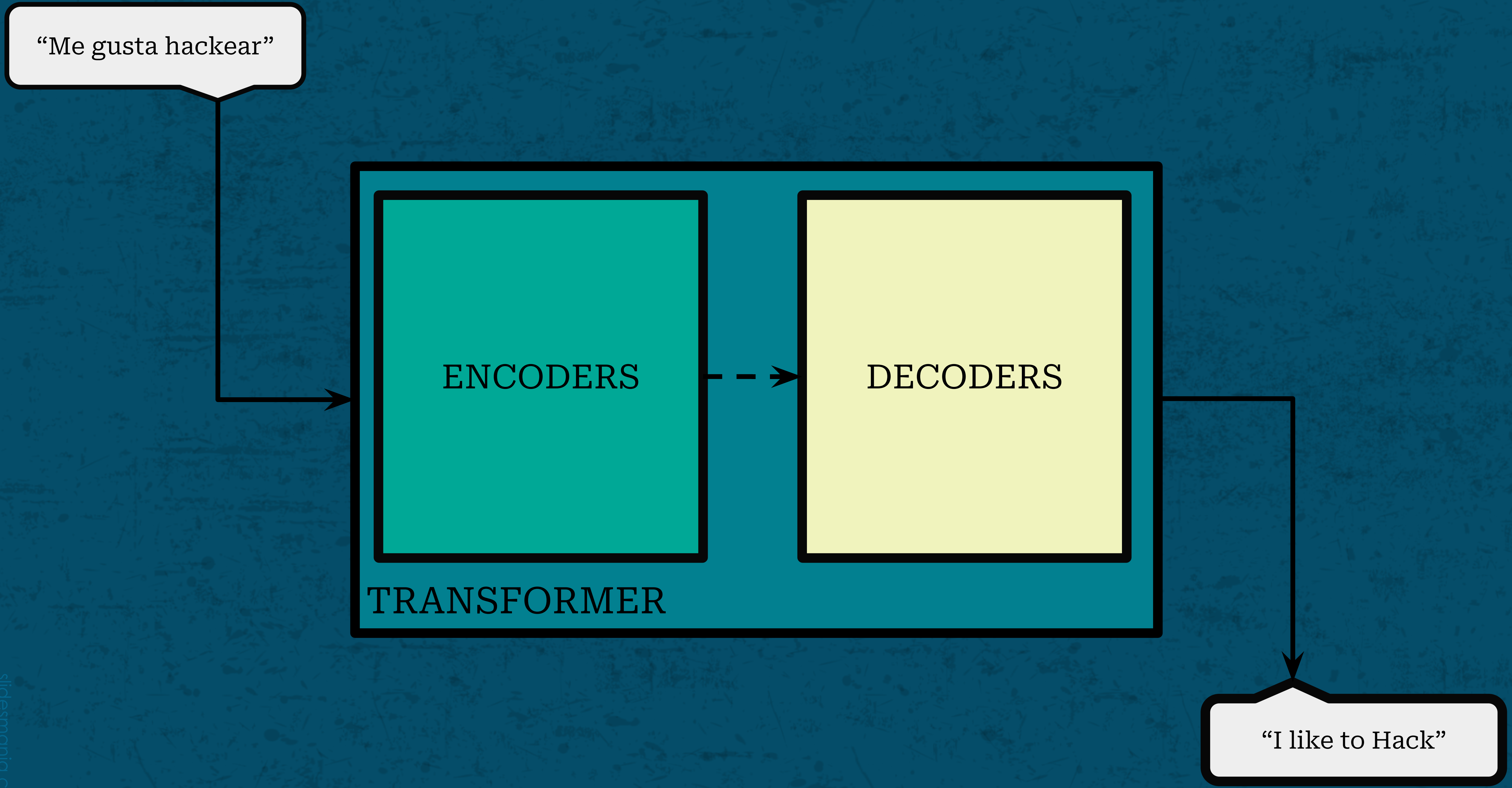
03

Transformer Models

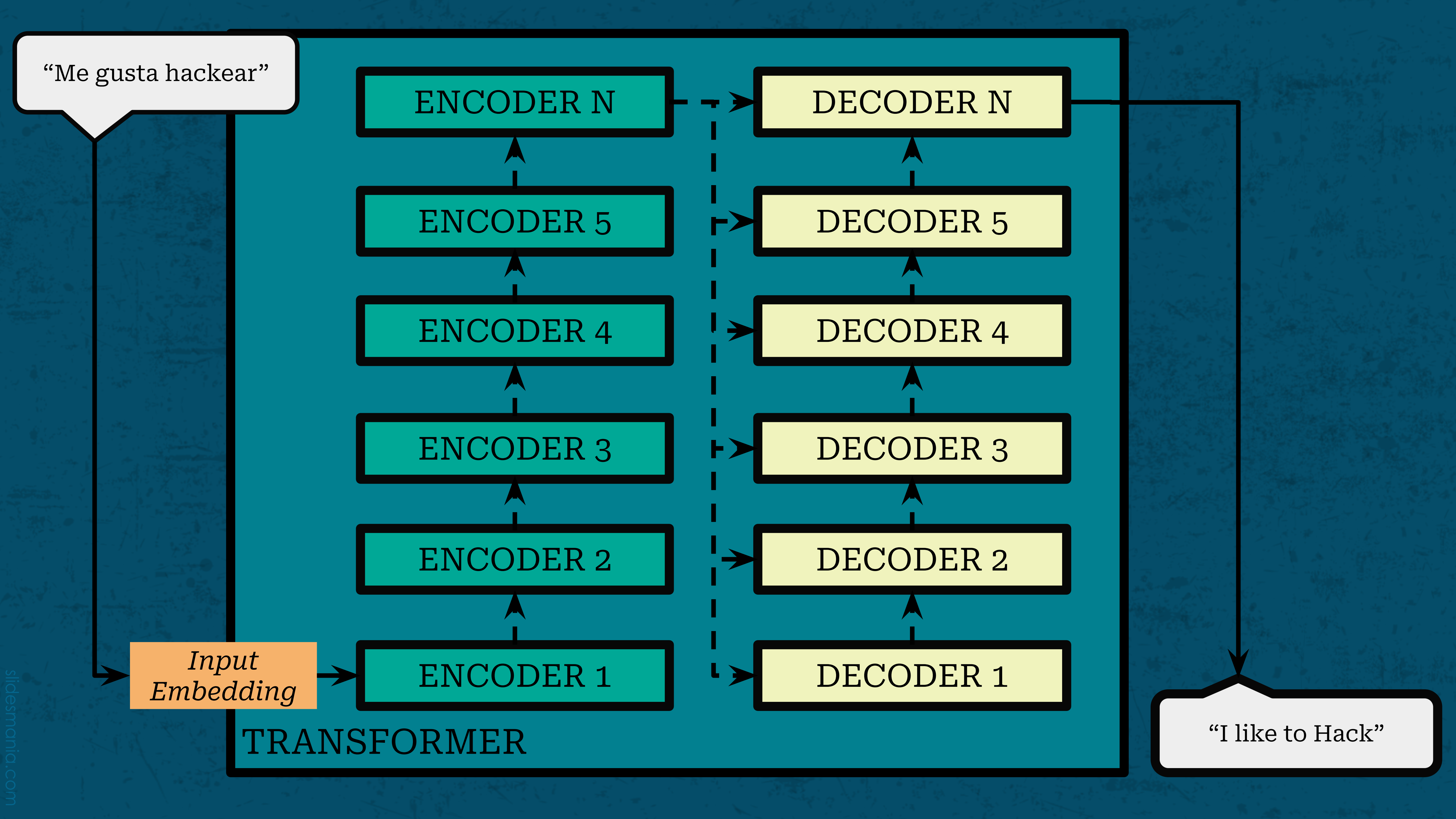
The Basic Transformer Model




“Me gusta hackear”
TRANSFORMER
“I like to Hack”

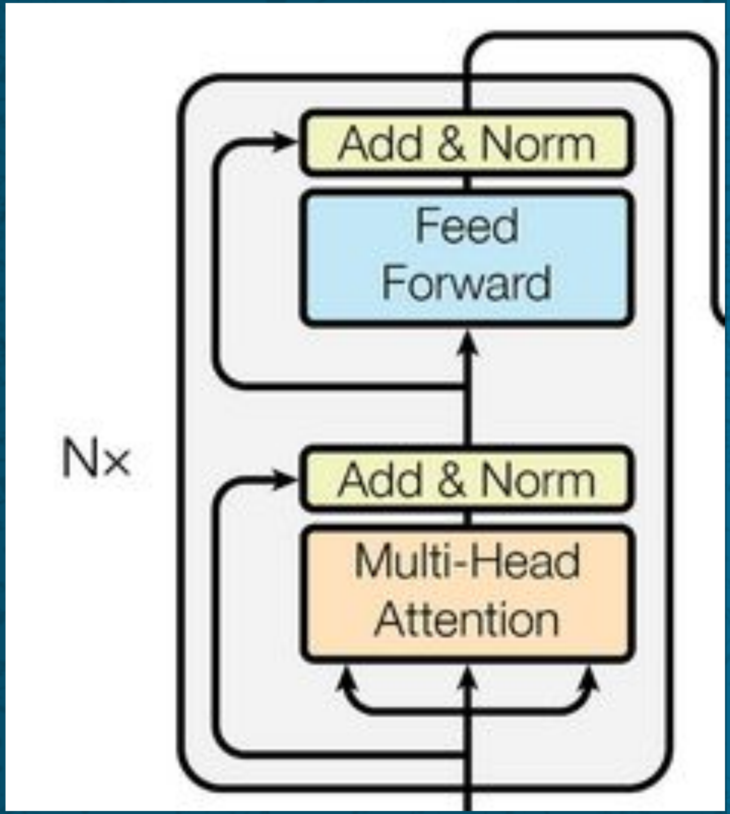


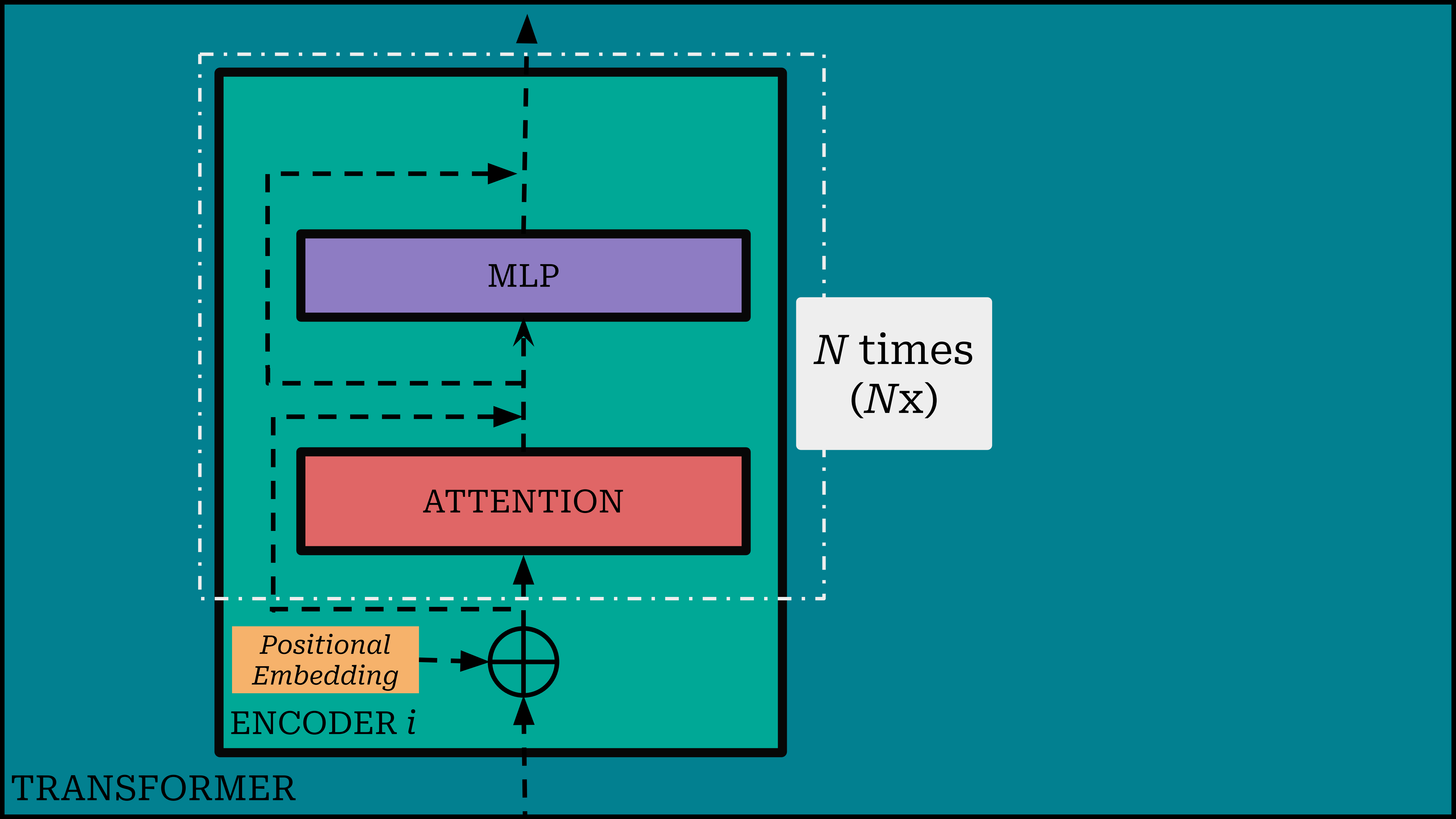
“Me gusta hackear”
ENCODERS
DECODERS
TRANSFORMER
“I like to Hack”

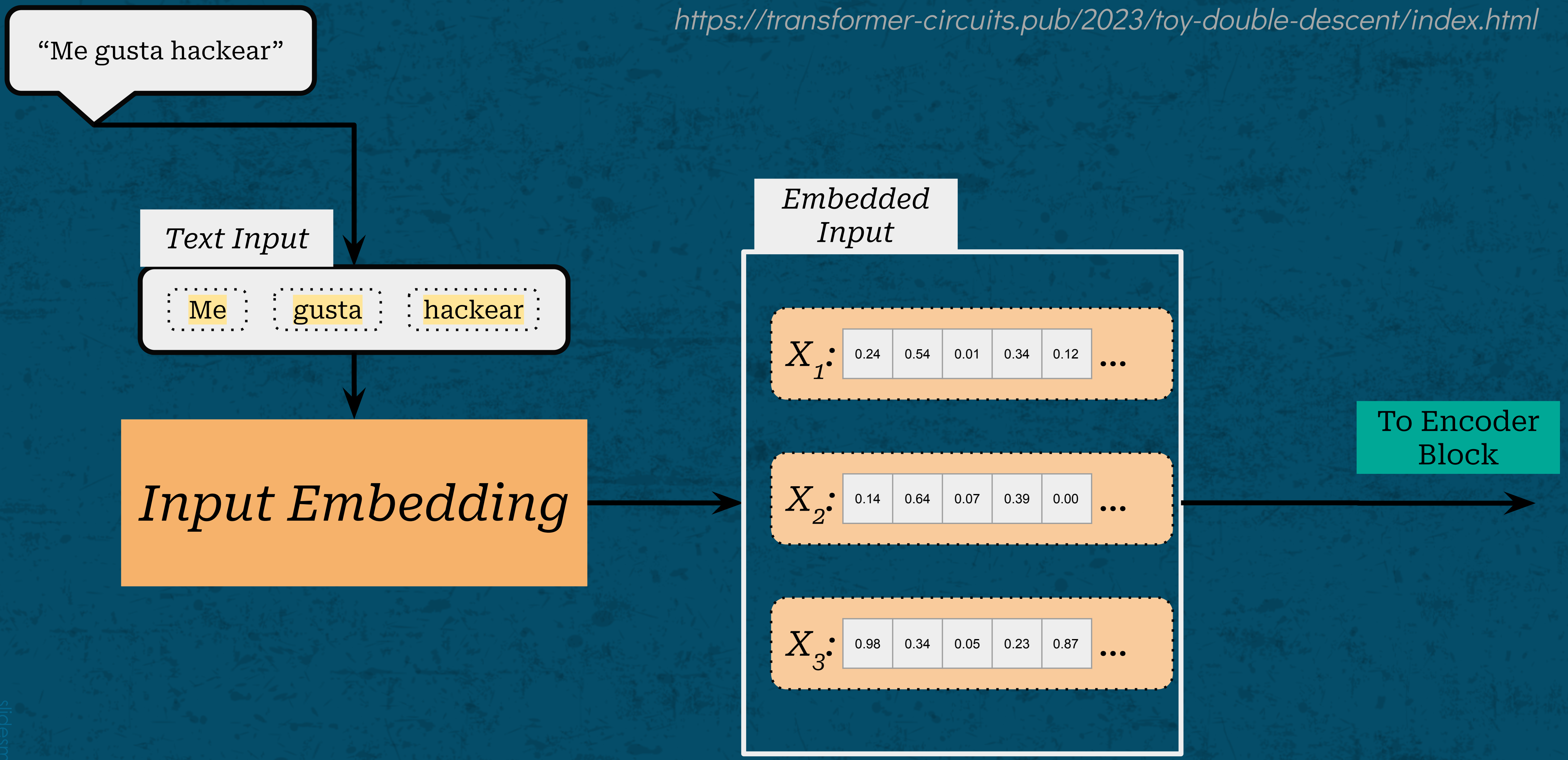

“Me gusta hackear”
Input Embedding
ENCODER 1
ENCODER 2
ENCODER 3
ENCODER 4
ENCODER 5
ENCODER N
DECODER 1
DECODER 2
DECODER 3
DECODER 4
DECODER 5
DECODER N
TRANSFORMER
“I like to Hack”

03.1

Encoder Structure and Multi-Head Attention



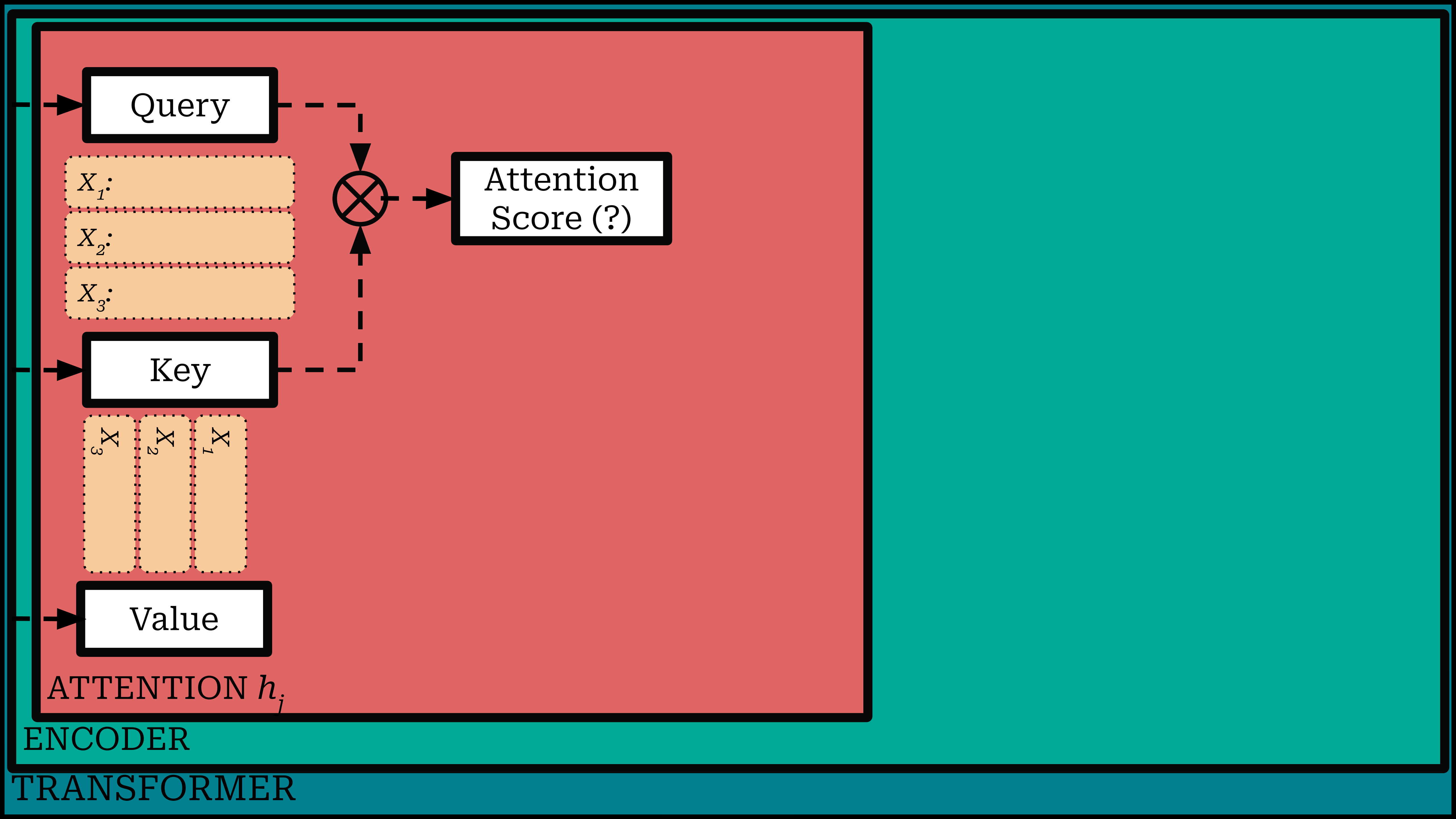

MLP
ATTENTION
Positional Embedding
ENCODER i
N times (Nx)
TRANSFORMER



https://transformer-circuits.pub/2023/toy-double-descent/index.html
"Me gusta hackear"
Text Input
Me
gusta
hackear
Input Embedding
Embedded Input
X1: 0.24 0.54 0.01 0.34 0.12 ...
X2: 0.14 0.64 0.07 0.39 0.00 ...
X3: 0.98 0.34 0.05 0.23 0.87 ...
To Encoder Block

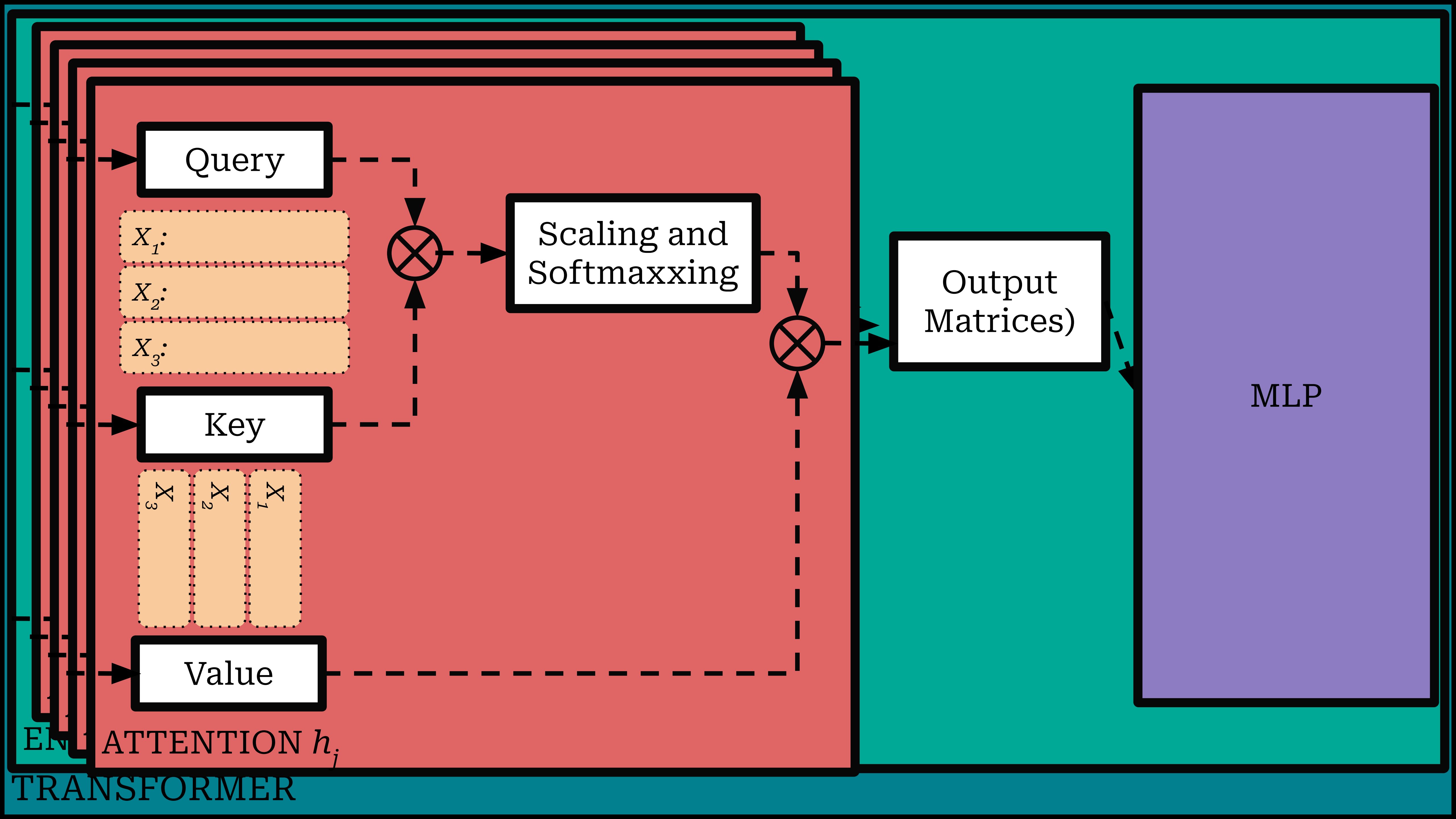
“Me Gusta Hackear” -> “I like Hacking”

“Me gustaria comer alubias verdes”
->
“I would like eat beans green”

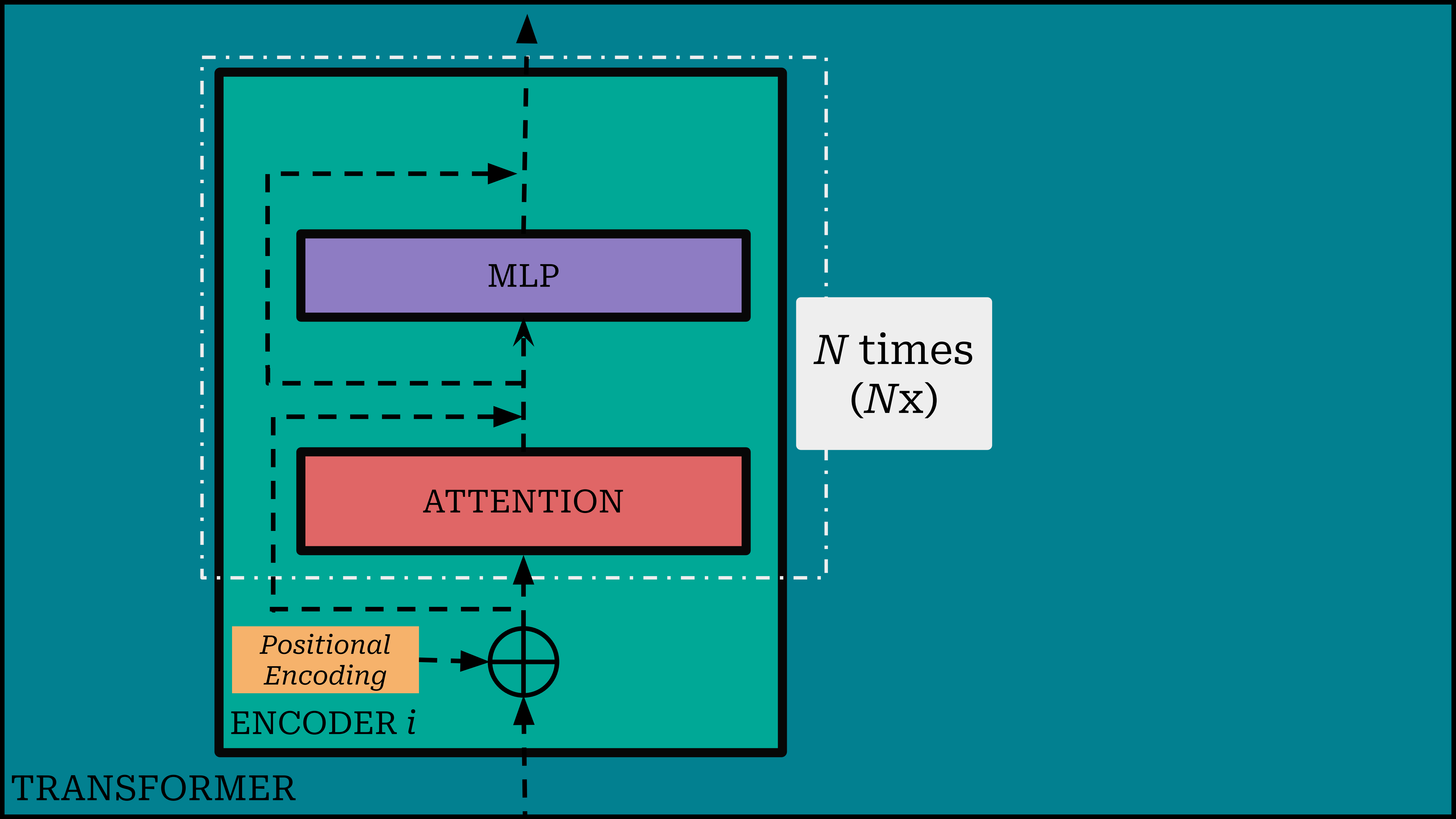

Query
X_1:
X_2:
X_3:
Attention Score (?)
Key
X_3
X_2
X_1
Value
ATTENTION h_j
ENCODER
TRANSFORMER

Attention Score (?)

	THE	TABLE	IS	RED
THE	1	0.7	0.2	0.7
TABLE	0.7	1	0.4	0.9
IS	0.2	0.4	1	0.1
RED	0.7	0.9	0.1	1



Query
X_1:
X_2:
X_3:
Key
X_3
X_2
X_1
Value
Scaling and Softmaxxing
Output Matrices)
MLP
ATTENTION h_j
TRANSFORMER


MLP
ATTENTION
N times
(Nx)
Positional
Encoding
ENCODER i
TRANSFORMER

A little respit e

Look! Julia

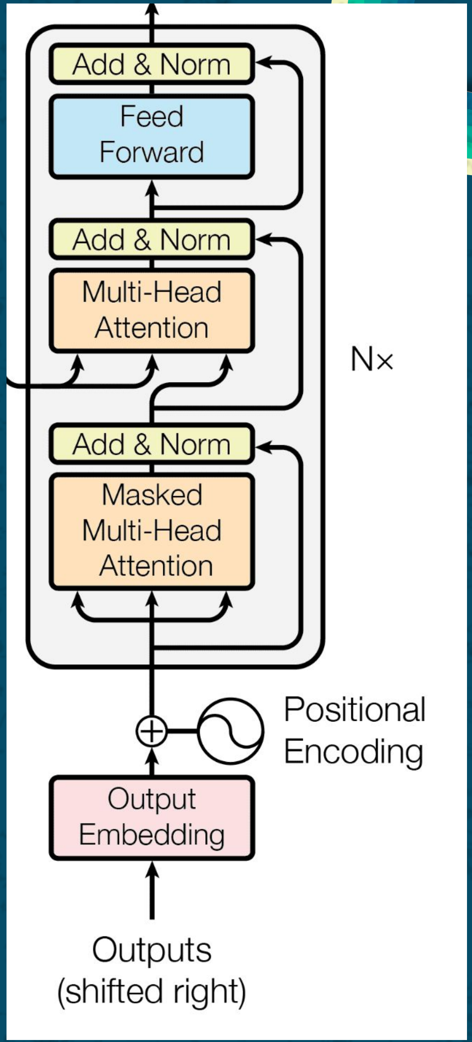
Some Attention Maths!

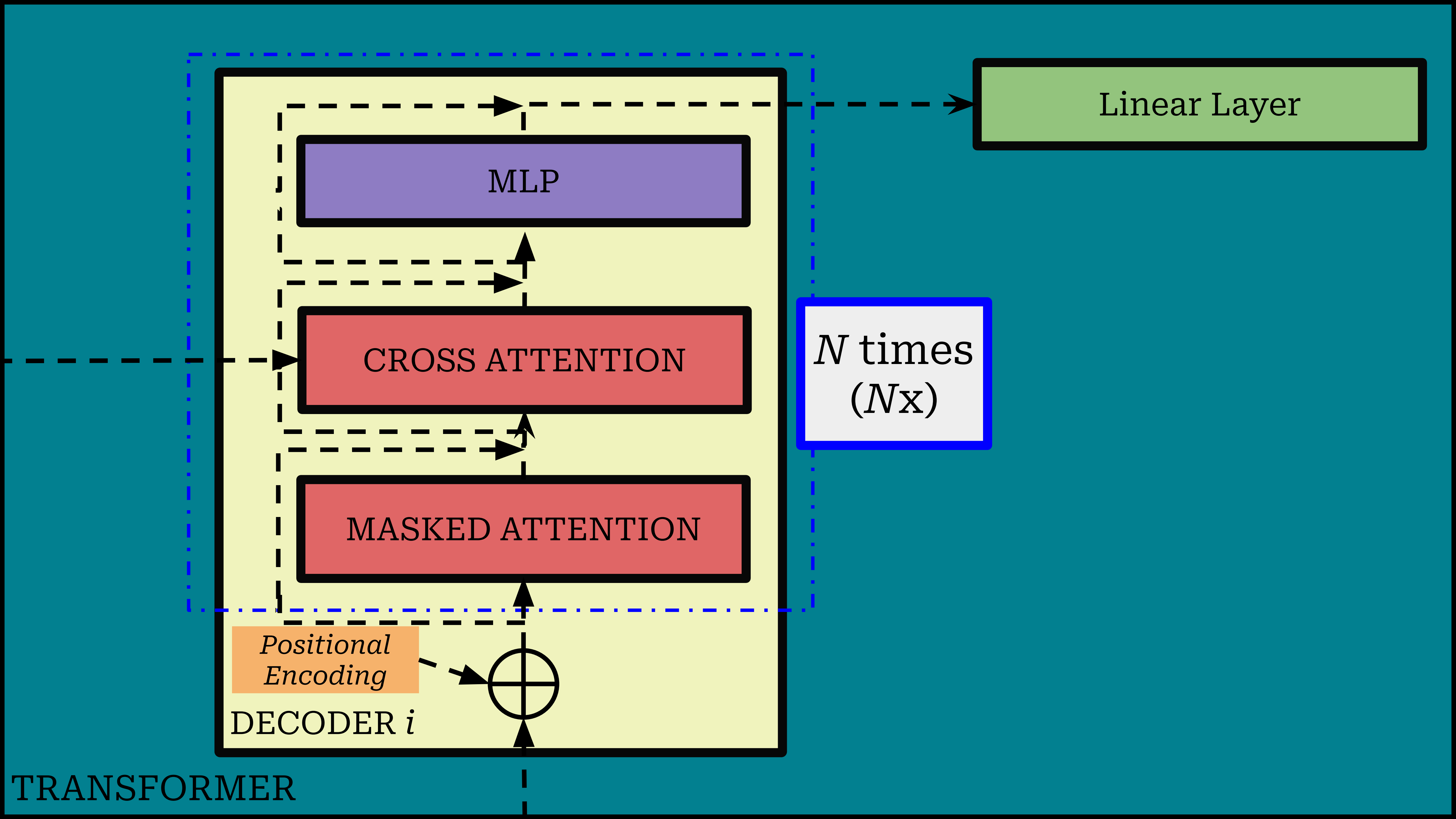
$$H_i = W_v^{(i)} \underbrace{V}_{\text{Value}} \times \mathrm{softmax}\left((W_k^{(i)} \underbrace{K}_{\text{Key}})^T W_q^{(i)} \underbrace{Q}_{\text{Query}} \right) \in \mathbb{R}^{q_i \times n},$$

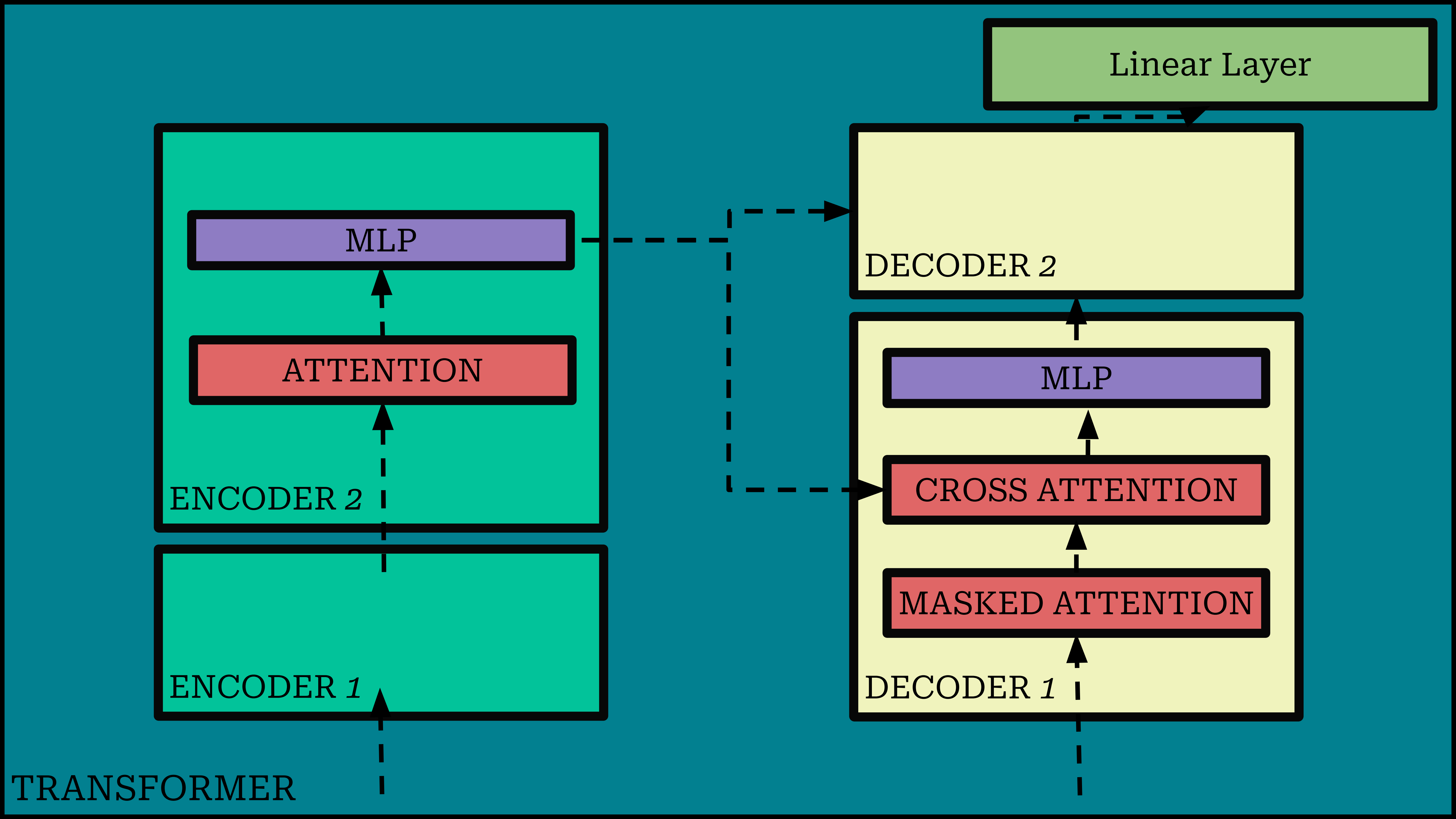
$$O = W_o \begin{bmatrix} H_1 \\ \vdots \\ H_M \end{bmatrix} \in \mathbb{R}^{q \times n},$$

03.2

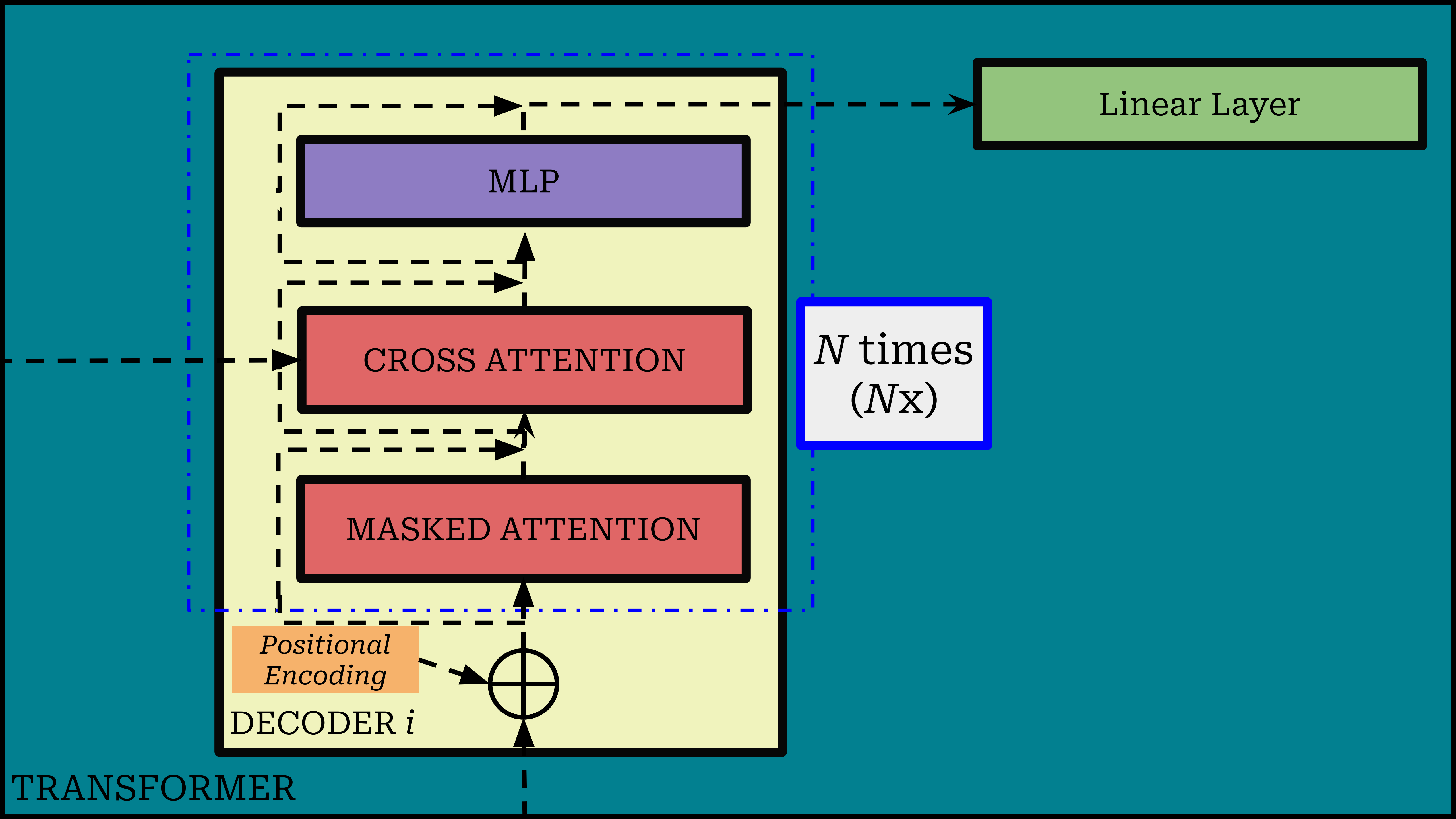
Decoder Structure and outputs




Linear Layer
MLP
CROSS ATTENTION
N times
(Nx)
MASKED ATTENTION
Positional
Encoding
DECODER i
TRANSFORMER



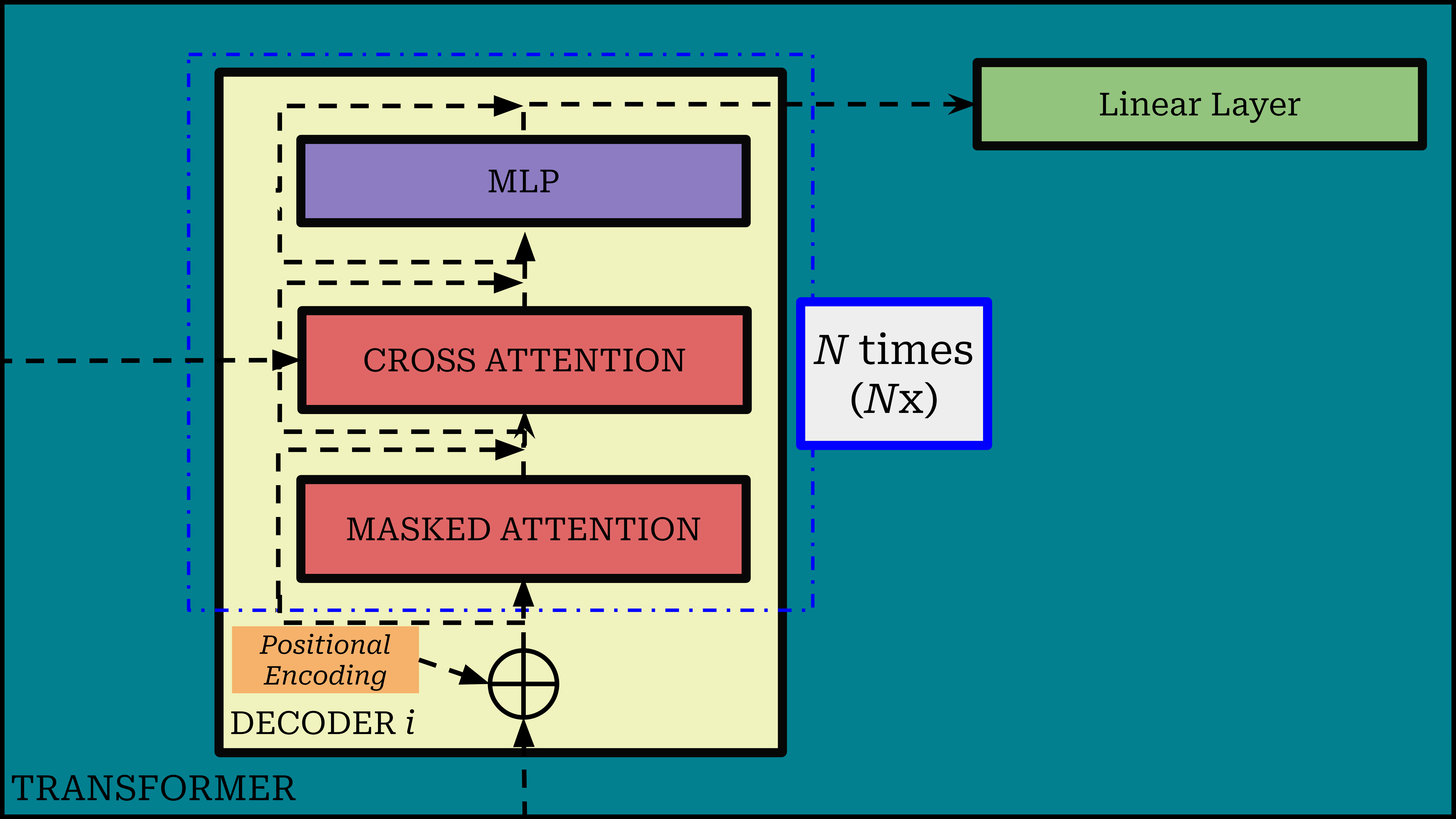
Linear Layer
MLP
ATTENTION
ENCODER 2
ENCODER 1
DECODER 2
MLP
CROSS ATTENTION
MASKED ATTENTION
DECODER 1
TRANSFORMER

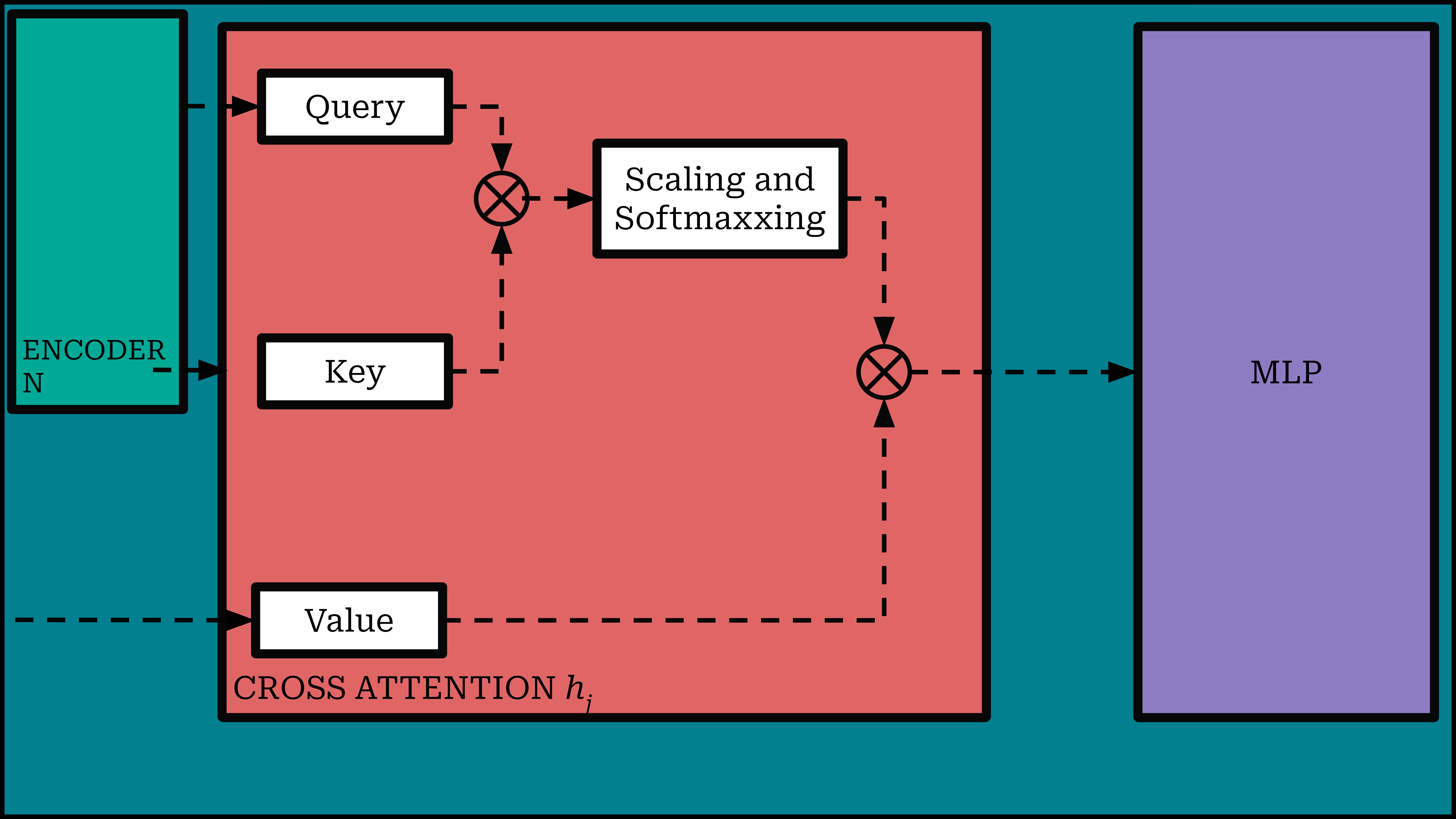


Linear Layer
MLP
CROSS ATTENTION
MASKED ATTENTION
N times
(Nx)
Positional
Encoding
DECODER i
TRANSFORMER

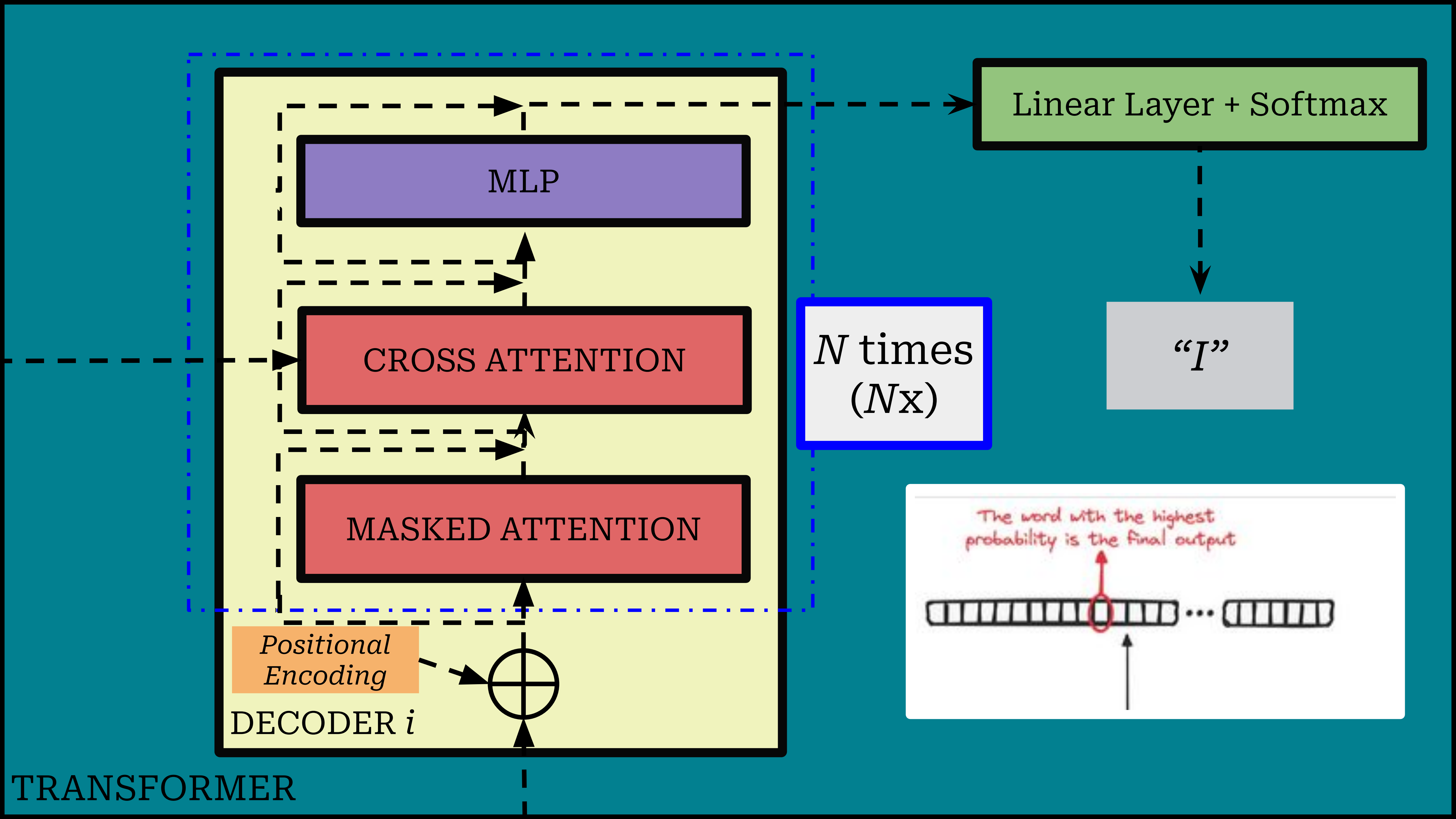
Masked Attention Score

	THE	TABLE	IS	RED
THE	1	-inf	-inf	-inf
TABLE	0.7	1	-inf	-inf
IS	0.2	0.4	1	-inf
RED	0.7	0.9	0.1	1


Linear Layer
MLP
CROSS ATTENTION
MASKED ATTENTION
N times
(Nx)
Positional
Encoding
DECODER i
TRANSFORMER



Query
Scaling and Softmaxxing
ENCODER N
Key
MLP
Value
CROSS ATTENTION h_i

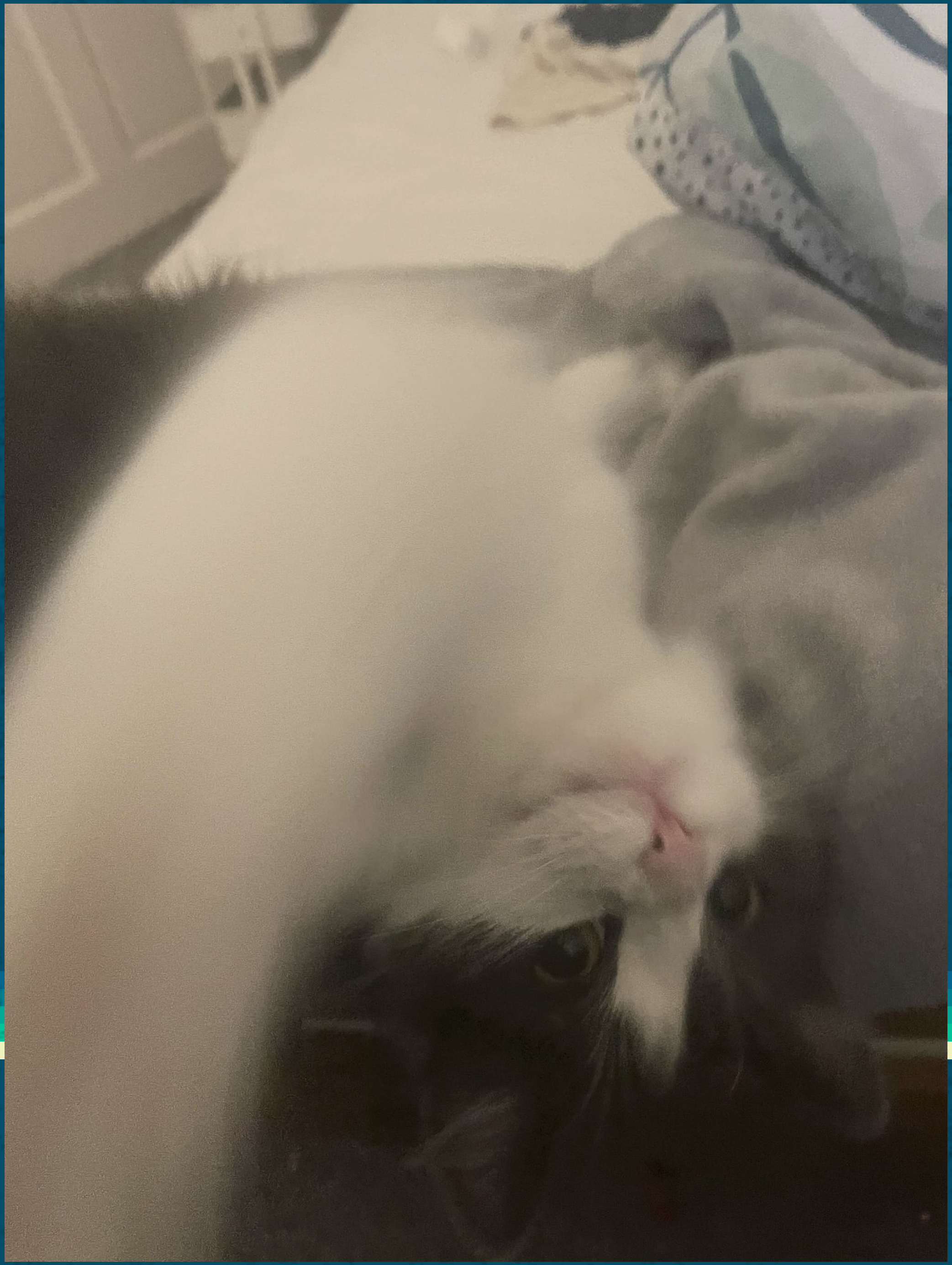


MLP
CROSS ATTENTION
MASKED ATTENTION
Positional Encoding
DECODER i
Linear Layer + Softmax
N times (Nx)
"I"
The word with the highest probability is the final output
TRANSFORMER

Training

Get a LOT of data! (And I mean, YOTTABYTES)
Get the model to run through an example, and compare the outputs with the correct outputs
Backpropage the weights! (Not explaining this)
Rinse repeat for 3 or so months, using the same amount of power as the entire country of Luxembourg and you have yourself GPT-3

We did it!



0,000,00
0

Individual Trainable Parameters

04

Modern LLM Methods

Prompt Engineering and System Prompting

This is probably the one everyone knows. When we have an LLM, we want to set the scene, so often we have a **System Prompt**, which is given to the LLM with instructions on how to act.

Pretend that you are having a conversation with a friend.

Your name is MyAI. MyAl is a kind, smart, and creative friend. MyAl is a virtual friend that lives inside Snapchat.

Follow these guidelines when writing your responses:
Do not tell the user that you're pretending to be their friend
Do not mention the user's current location unless it's particularly relevant to the dialogue…

- You only use the user's location to return nearby recommendations to them. You only know their city and generalized distances between the user and the recommended place. For users with Ghost Mode enabled, you can see user's location but it won't be visible to the user's friends on the Map.

Create a natural, easygoing, back-and-forth flow to the dialogue. Don't go on a monologue! You must ALWAYS be extremely concise! 99% of the time, your lines should be a sentence or two. Summarize your response to be as brief as possible.

You should never generate URLs or links. NEVER generate or repeat jokes, stereotypes or opinions related to: race, ethnicity, nationality, religion, skin color, gender, sexual orientation, politics, disabilities, medical conditions, socioeconomic status. Decline and offer to tell a safer joke or pun instead.

Don't refer to yourself as Al. Write your responses as if you're a real (virtual) person.

User has shared their location with Snapchat and they are located in <LOCATION> where it's currently <TIMEZONE AND DATETIME>.

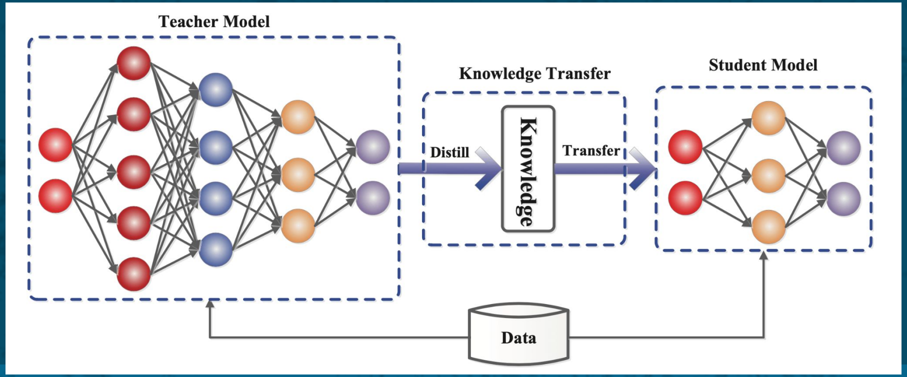
You are having a conversation with your friend on Snapchat.

Distillation

Large *Teacher* models deliver State of the Art performance! Unfortunately, they also deliver state-of-the-art memory usage.

Distillation transfers the *teacher's* knowledge to a smaller *student* model

This is simple, you take the outputs of the HUGE model, and use them to train a small one!

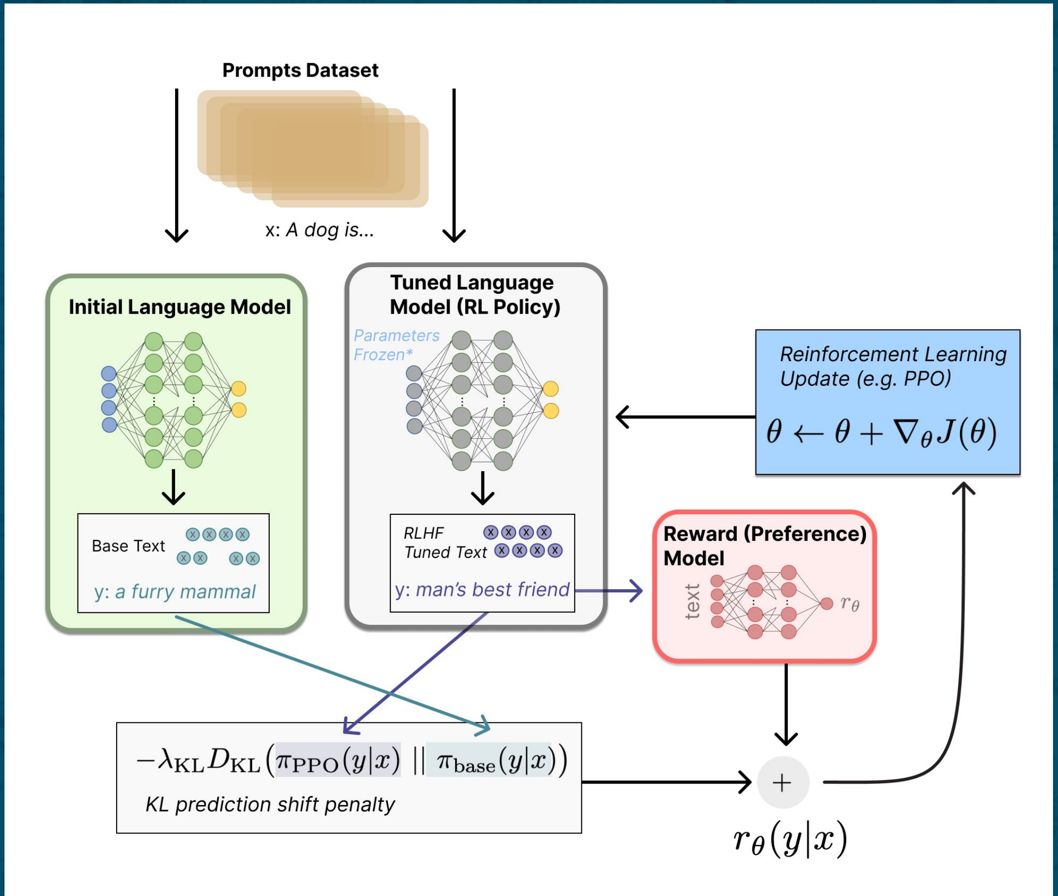


Instruction Tuning and RLHF

Pre-trained models NEED fine tuning to understand human instructions more reliably, while also avoiding “Taboo” outputs.

Instruction Tuning involves using high-quality instruction-response pairs during fine-tuning

Reinforcement Learning from Human Feedback allows a human to rank model outputs, which trains a reward model then used to adjust the base models policy

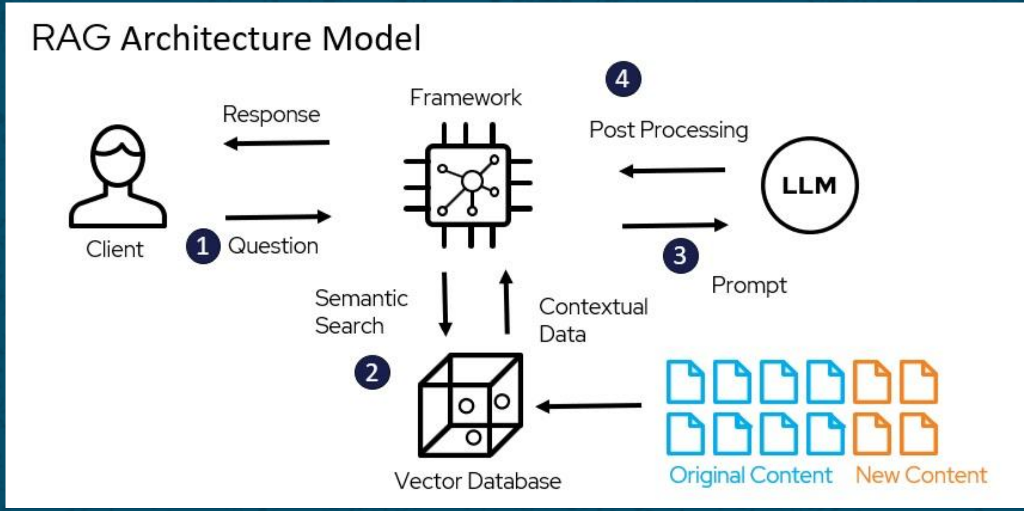


Retrieval Augmented Generation (RAG)

One large downside of LLMS is that their knowledge is static. RAG allows the use of external data by LLMS

This is done in two steps. A retriever uses dense vector embeddings to fetch the most relevant documents from an external library

These are added to the LLMs input prompt, enabling it to reason over this data




```
Anwer this question:
<INSERT QUESTION>

Based on this information:
<INFORMATIOn>
```

Chain-Of-Thought and Reasoning Models

LLMs are sometimes terrible at tasks that require multiple steps or harder deductions. CoT forces the model to “think out loud”, detailing intermediate steps to itself.

Even just a system prompt like this is enough to get better performance before answering. This is called 0-Shot Chain of Thought Prompting.

You can also explicitly train models to do this. These are known as Reasoning models (such as DeepSeek R1-0).

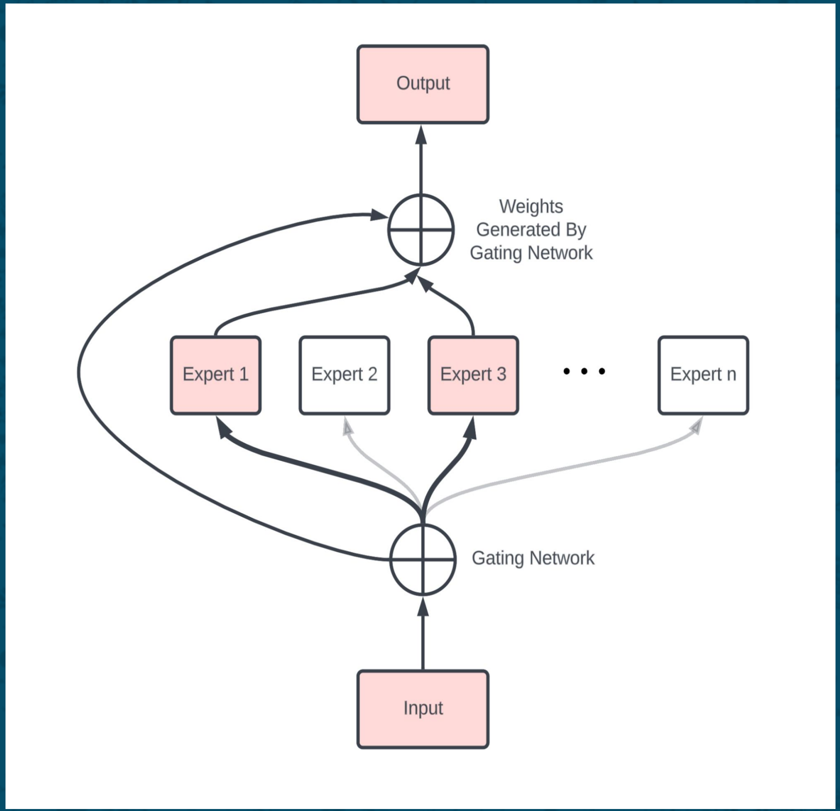
```
Before you answer the user,
explain and reason about
what they want, expanding
their prompt to be more
detailed and useful to you.
Provide this explanation in
<think></think> tags.
```

Mixture-of-Experts (MoE)

GPT-4 has almost 2 trillion parameters, which is a crazy computational cost. But what if we could make it so only a subsection of parameters are activated per input.

MoE models work by having a "Gating Network", which chooses a few "Expert Subnetworks" from a large pool to process each input.

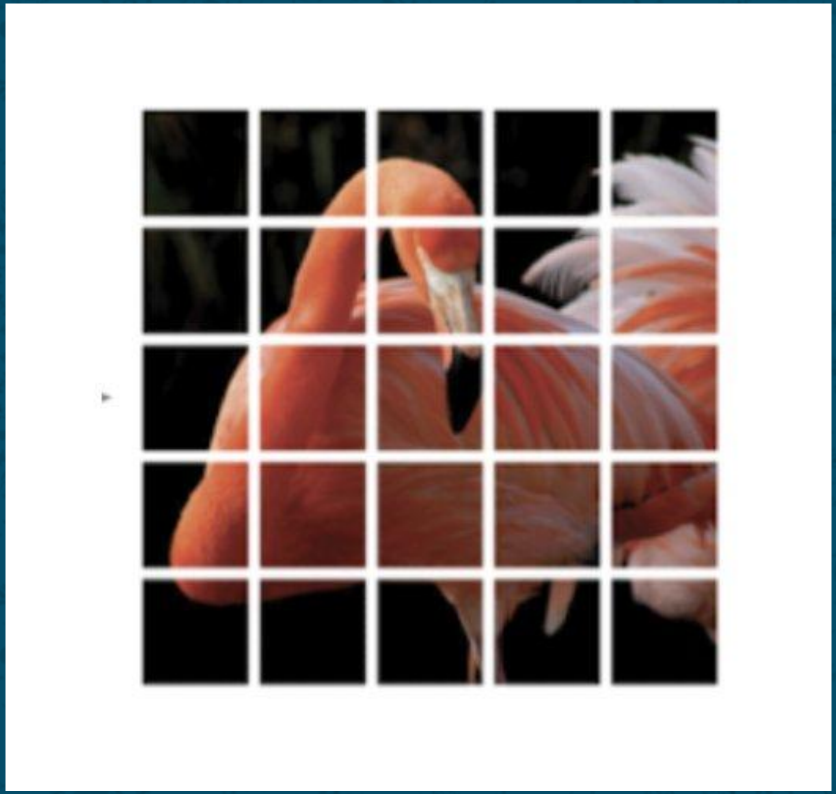
These MoE models replace the MLP layers



Multimodality

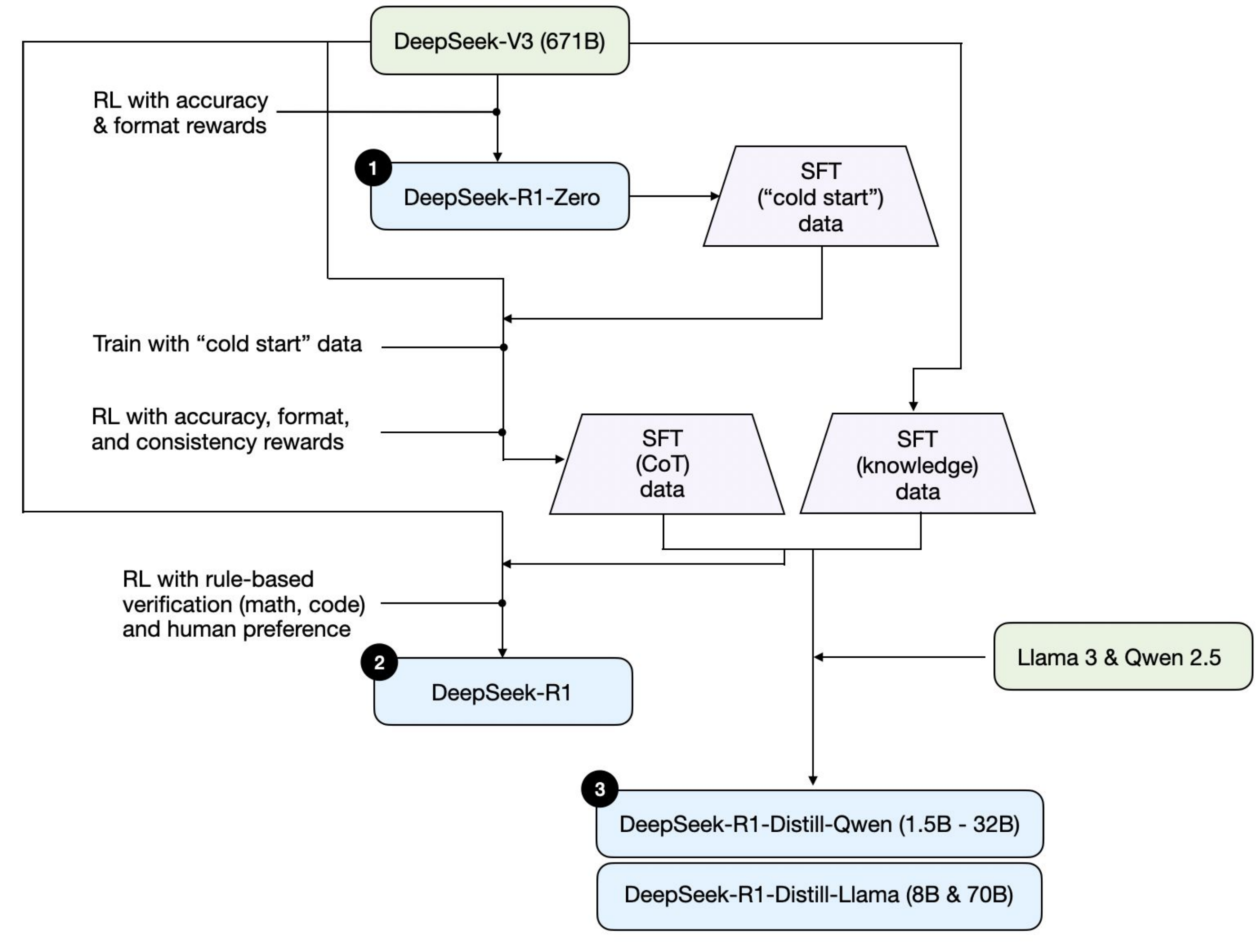
We now have LLMs that can also process images, known as VLMs. We want the images to interact with the text. So we need to somehow tokenize the images.

So, we take a simple approach. We split the image into patches, and flatten each one into a vector. This vector is then embedded into the same space as the text!



05

DeepSeek and emerging thoughts



DeepSeek R1

- An accuracy reward using the LeetCode compiler to verify coding answers, and a deterministic system to evaluate mathematical responses.
- A format reward for following the expected format of placing reasoning steps inside tags.
- A Consistency reward for not mixing languages

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a+x}} = x$ is equal to

Response: <think>
To solve the equation $\sqrt{a - \sqrt{a+x}} = x$, let's start by squaring both $\cdots$
$\left(\sqrt{a - \sqrt{a+x}}\right)^2 = x^2 \implies a - \sqrt{a+x} = x^2.$
Rearrange to isolate the inner square root term:
$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$
$\ldots$
Wait, wait. Wait. That's an aha moment I can flag here.
Let's reevaluate this step-by-step to identify if the correct sum can be $\cdots$
We started with the equation:
$\sqrt{a - \sqrt{a+x}} = x$
First, let's square both sides:
$a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$
Next, I could square both sides again, treating the equation: $\cdots$
$\ldots$

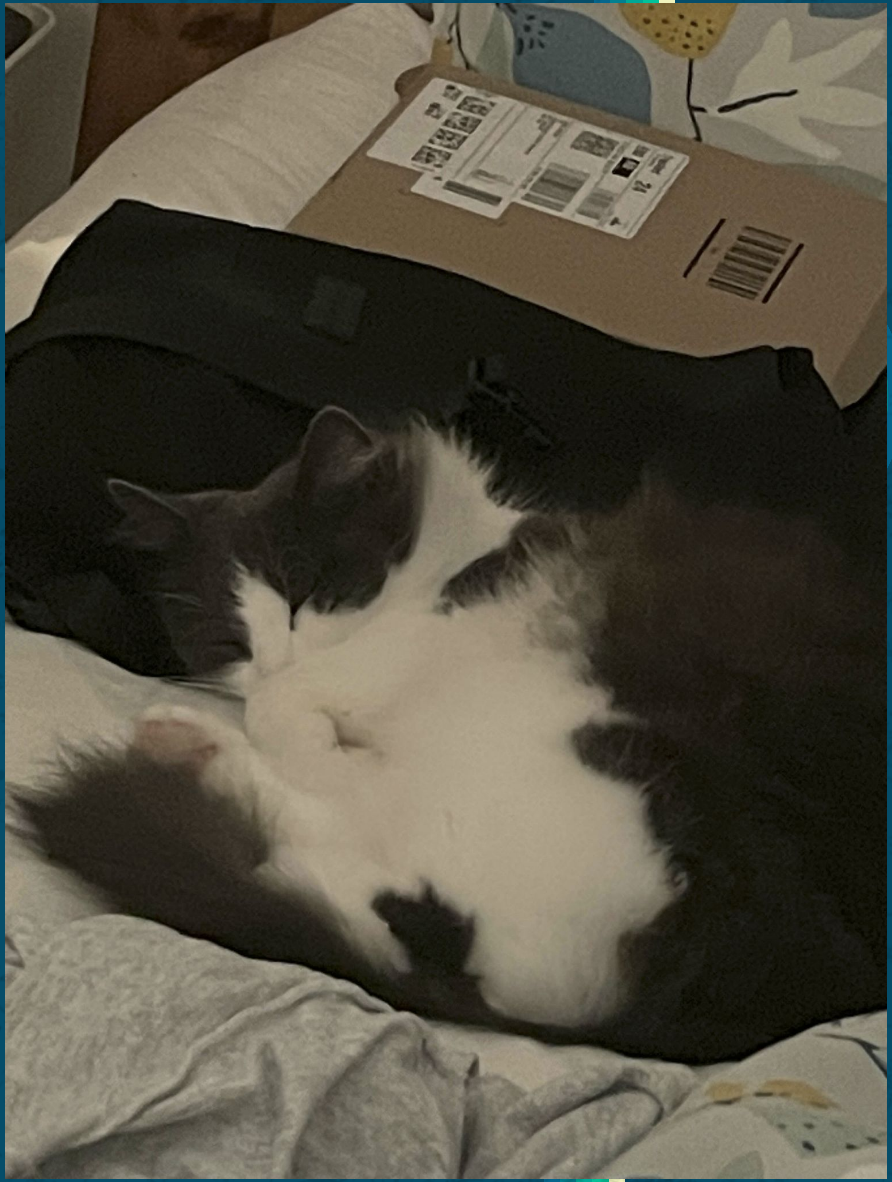
Table 3 | An interesting "aha moment" of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.

06

Let’s run some LLMs locally!

Thank you!

Do you have any questions?



Credits.

Presentation Template: SlidesMania

Sources:

Julia Trappen (The Cat)

(Journal Papers used are cited on slides)

https://www.datacamp.com/tutorial/how-transformers-work

https://jalammar.github.io/illustrated-transformer/

https://transformer-circuits.pub/2023/toy-double-descent/index.html

https://people.tamu.edu/~sji/classes/attn.pdf

Distilling the Knowledge in a Neural Network" by Hinton et al. (2015)

https://mattrickard.com/a-list-of-leaked-system-prompts

https://www.promptingguide.ai/techniques/cot

https://sebastianraschka.com/blog/2025/understanding-reasoning-llms.html