



VIS 2023



TransforLearn: Interactive Visual Tutorial for the Transformer Model

Lin Gao¹, Zekai Shao¹, Ziqin Luo¹, Haibo Hu², Cagatay Turkay³, Siming Chen^{1,4}

¹School of Data Science, Fudan University

²School of Big Data & Software Engineering, Chongqing University

³Centre for Interdisciplinary Methodologies, University of Warwick ⁴Shanghai Key Laboratory of Data Science

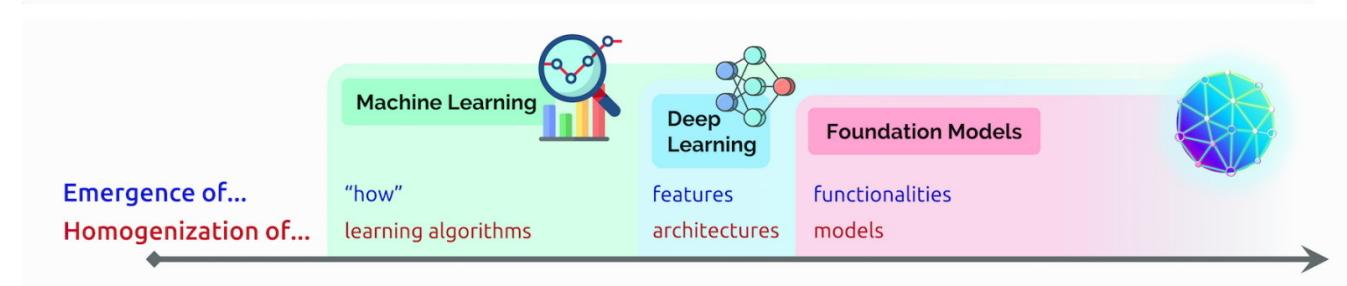
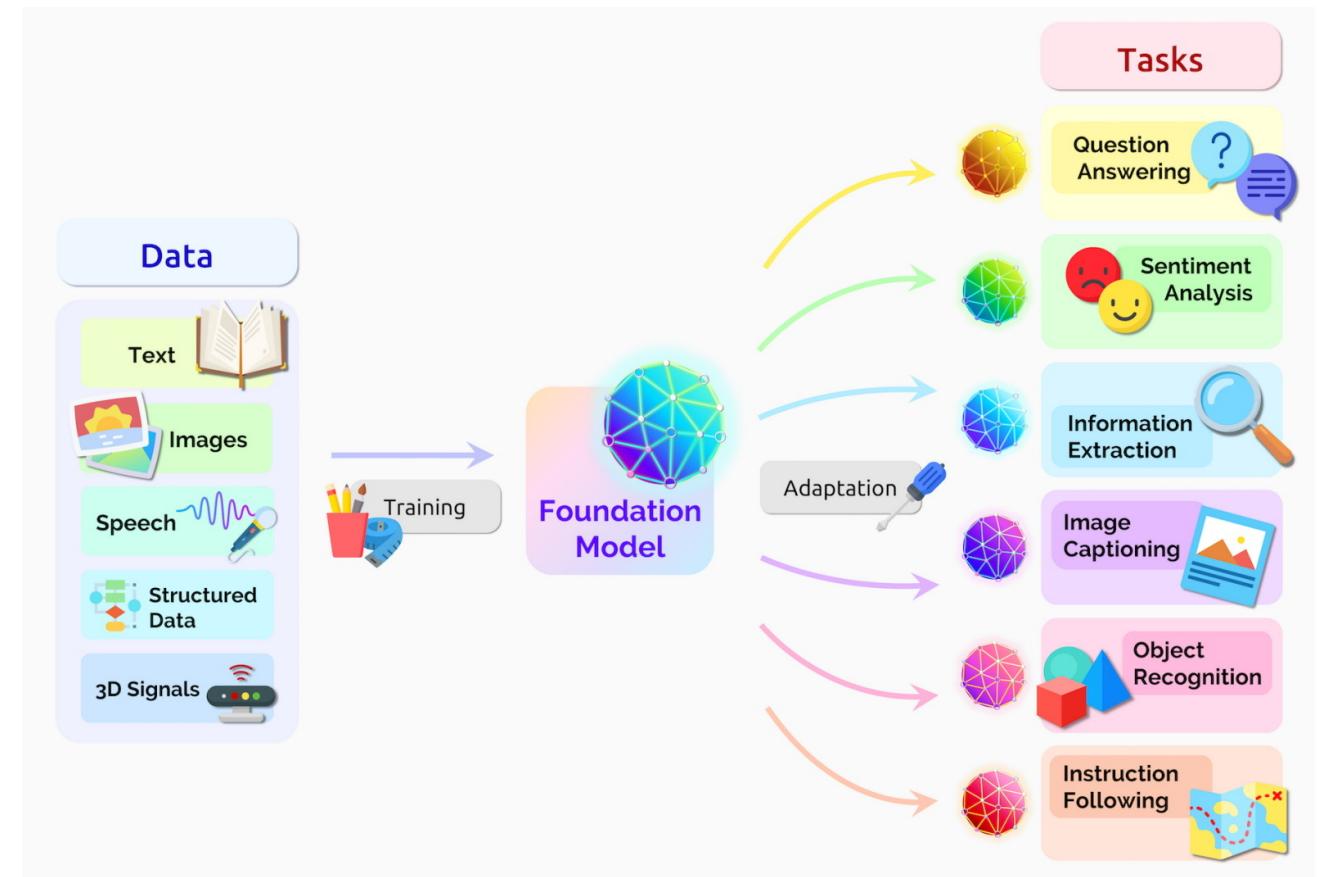


Background

Transformers are already used with many data sources for applications.

Transformers mark the next stage of AI's development, what some call the era of transformer AI.

The popularity of Transformer has sparked significant interest in learning its working mechanisms.





Background

Visualization for understanding deep learning models

- how the models make decisions & what they learned
 - model improvement & debugging



M2lens [1] (TVCG 2021)



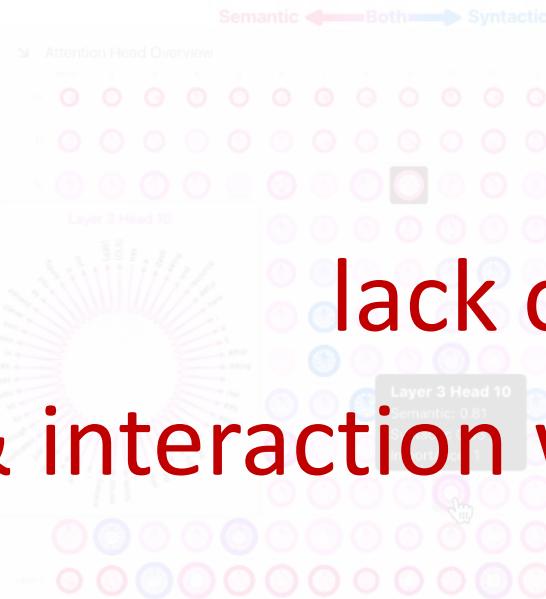
VEQA^[2] (TVCG 2023)



Background

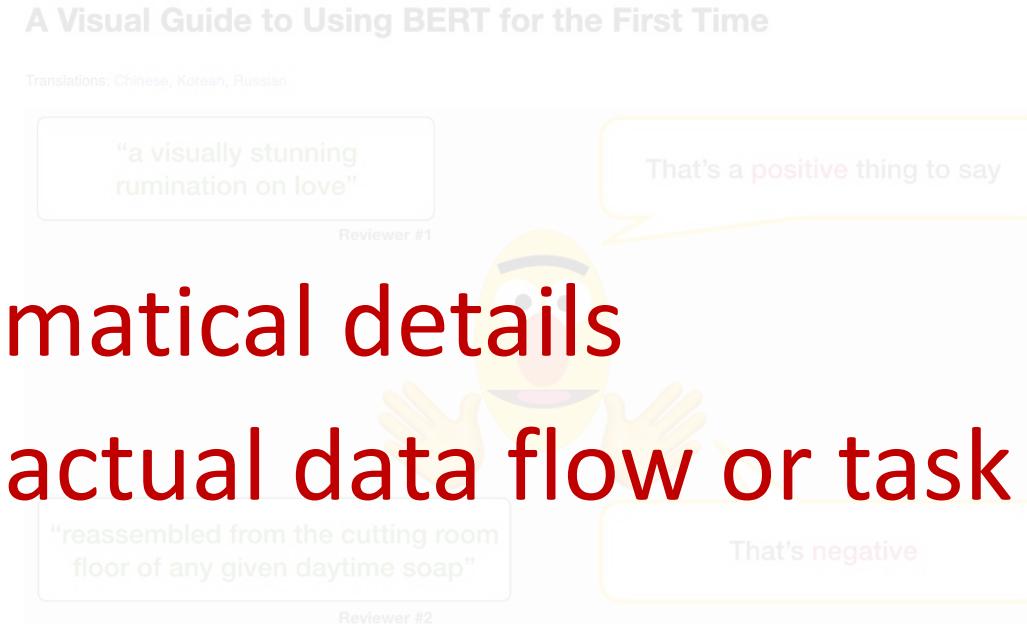
Visual interpretation of Transformers

- interpretation of embedding and attention mechanisms
- blogs & videos for tutorial



lack of mathematical details
& interaction with the actual data flow or task

Dodrio^[1] (ACL 2021)



Jalammar's blogs^[2]

[1] Wang Z J, Turko R, Chau D H. Dodrio: Exploring transformer models with interactive visualization[J]. arXiv preprint arXiv:2103.14625, 2021.

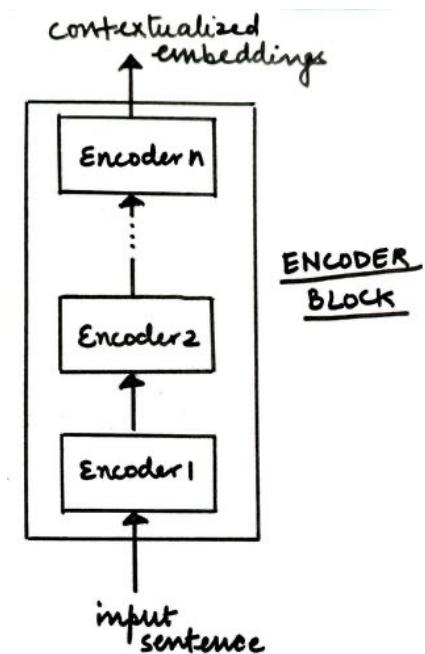
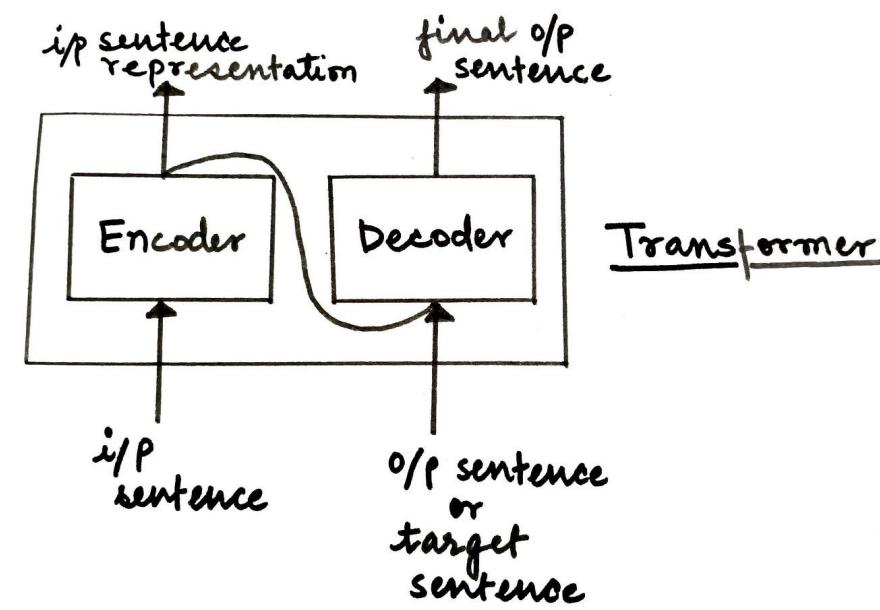
[2] <https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/>



Preliminary Study

For lecturers, they need to **manually** break down Transformer into multiple steps and discuss them in a sequence of slides.

- Theoretical learning -> **Dynamic thinking combined with practice**
- Contextual interactions
- Class engagement





Preliminary Study

Beginners face difficulties in comprehending and learning Transformers due to its **complex structure**, **data transformation** and **abstract downstream task**.

- Encoder/Decoder, Attention
- Embedding, Dimension
- Alignment, Process

Calculation & Principles

Sequence & Embedding

Multi-head
Self-Attention

Interaction & Combination

Specific Task Requirements

Data Flow
Between Layers

Encoder-Decoder

Proper Model Structure &
Layer Combination

Dimension Transformation

Alignment of
Output/Input

Layer Normalization

Input/Output Presentation

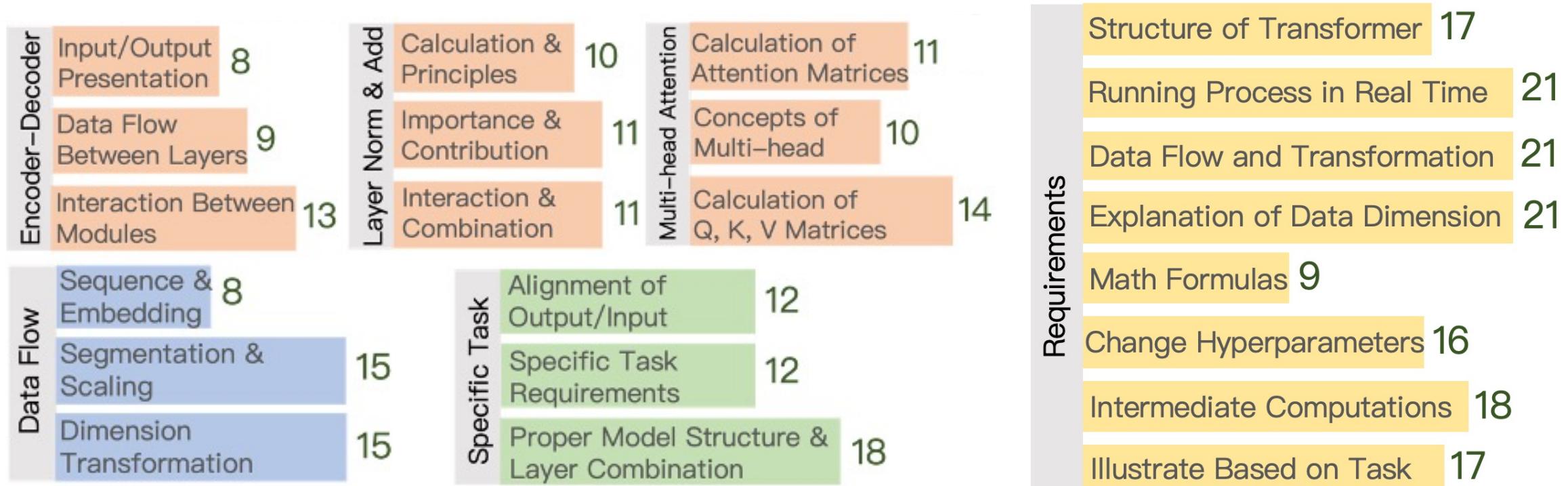
Segmentation & Scaling





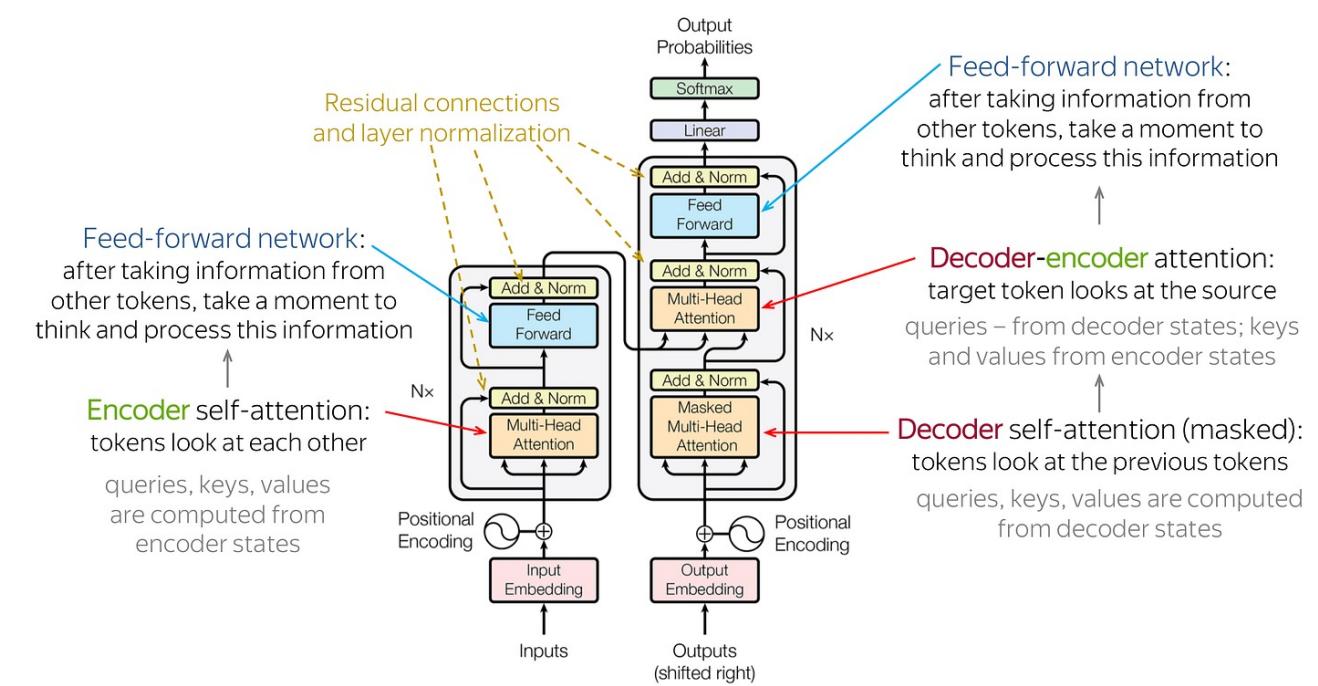
Preliminary Study

The survey asked about the **key challenges** in learning and applying Transformers from various aspects, and **what features would be helpful** in an interactive tool for beginners.



Consequently, an interactive visual tutorial is needed for deep learning **beginners** and **non-experts** to comprehensively learn about Transformers.

What can TransforLearn do?

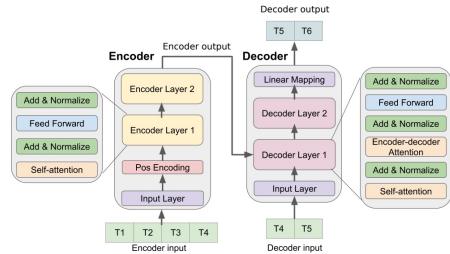


Why does Transformer has such a complex architecture^[1]

[1] <https://stats.stackexchange.com/questions/512242/why-does-transformer-has-such-a-complex-architecture>

Tasks & Requirements

Task-1



complex structure
& layer operations



Requirement-1

A **visual summary** of the model **architecture** and data flow.

Task-2



data flow &
transformation



Requirement-2

An **interactive interface** for **layer operations** and mathematical **formulas**.

Task-3



practical use
in downstream tasks



Requirement-3

Exploration mode between module levels based on downstream tasks.

Task-4



guidance &
feedback



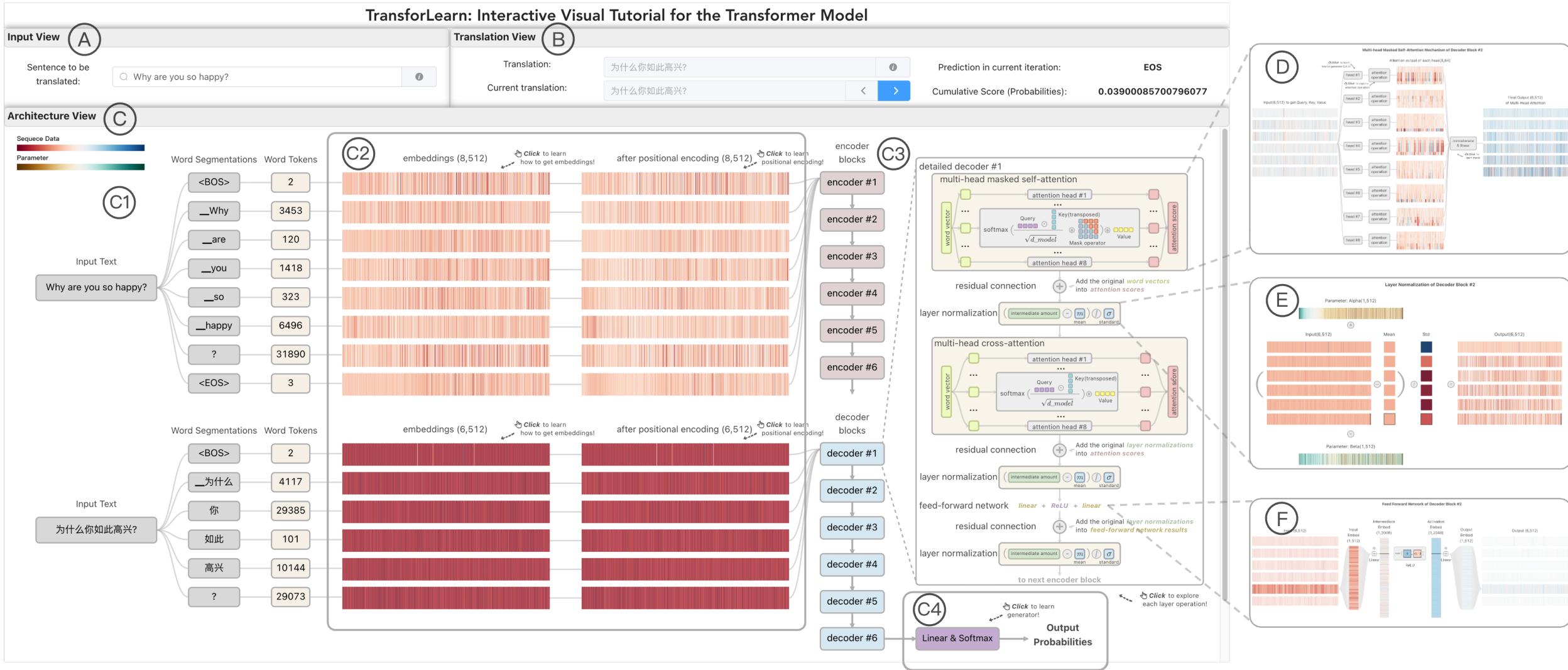
Requirement-4

Self-directed and **immersive** learning experiences.



Visual Design

TransforLearn: Interactive Visual Tutorial for the Transformer Model





Visual Design - Overview

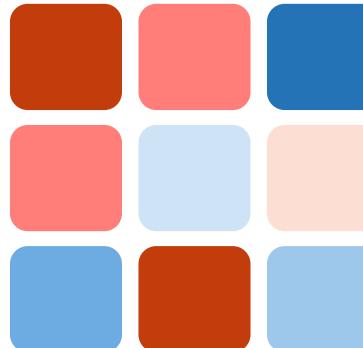
Architecture Overview



Module Detailed Views

$\begin{bmatrix} 11 & 20 & 109 \\ 21 & 54 & 37 \\ 74 & 11 & 60 \end{bmatrix}$

Sequence Data
Parameter Data



Breaking text into individual word segmentations.

Word Segmentations

Index in word token dictionary.

Word Tokens

Mapping words to dense vector representations.
embeddings (4,512)

Add positional information to original embeddings.

after positional encoding (4,512)

Add the original *layer normalizations* into *attention scores*

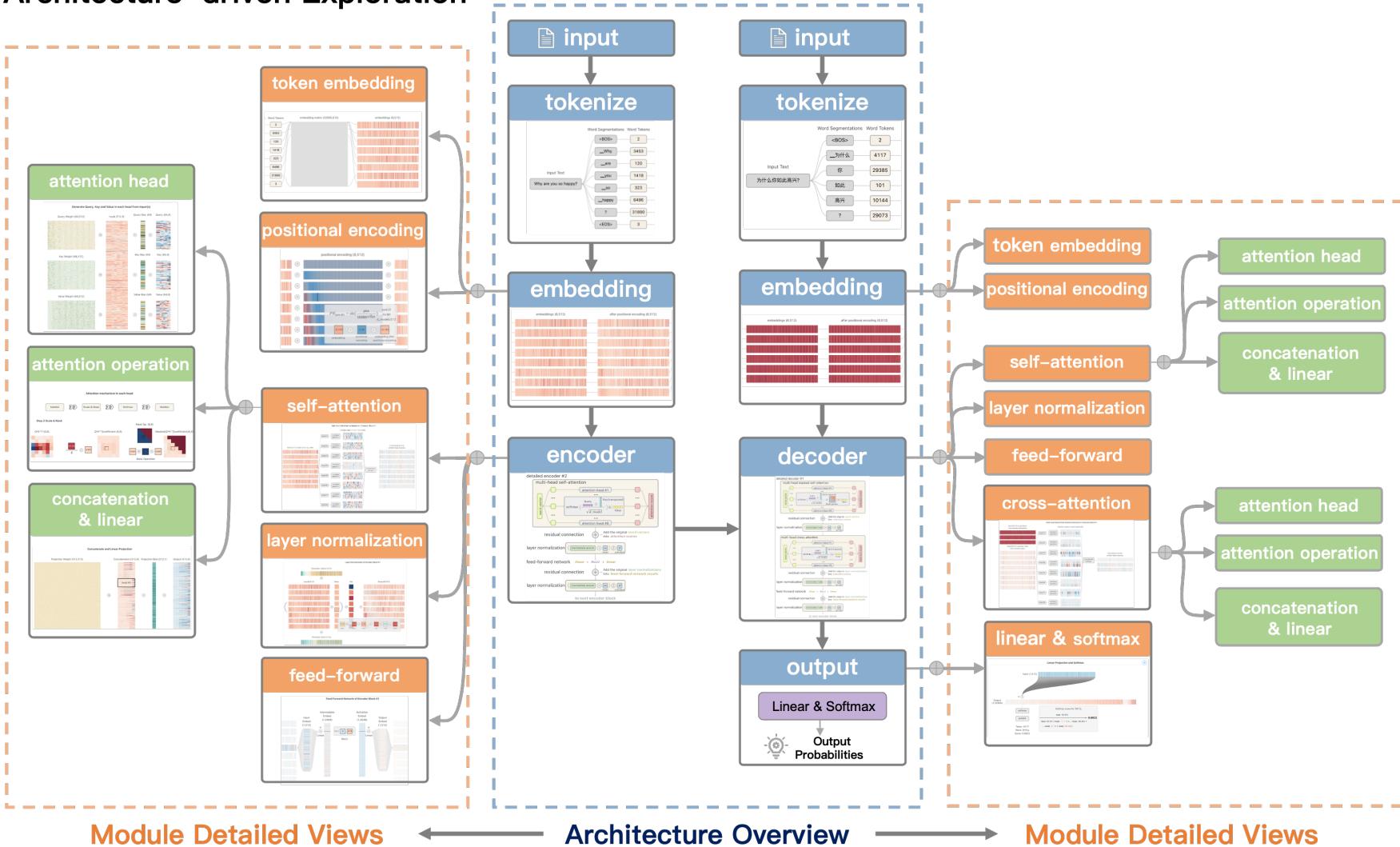
Add the original *layer normalizations* into *feed-forward network results*

Click to explore each layer operation!

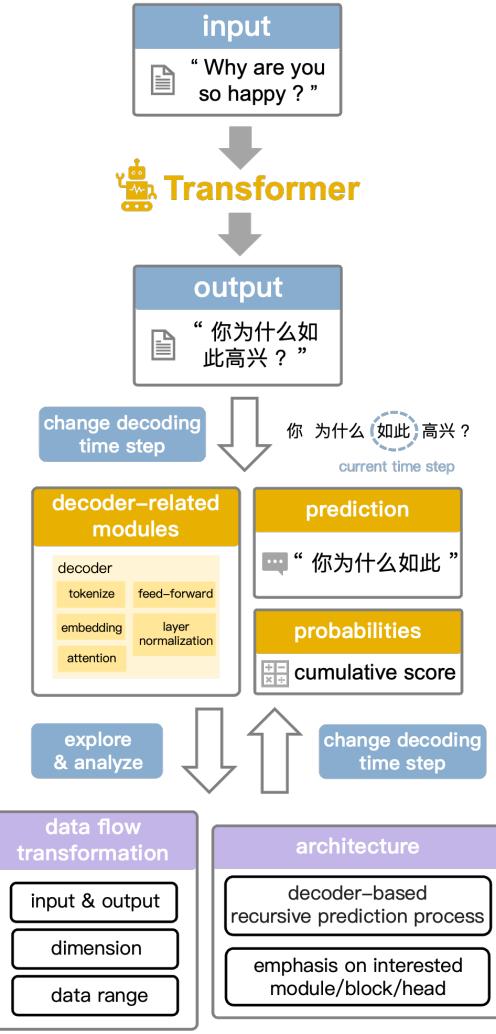
Click to learn how to generate Q,K,V!

Visual Design - Overview

Architecture-driven Exploration

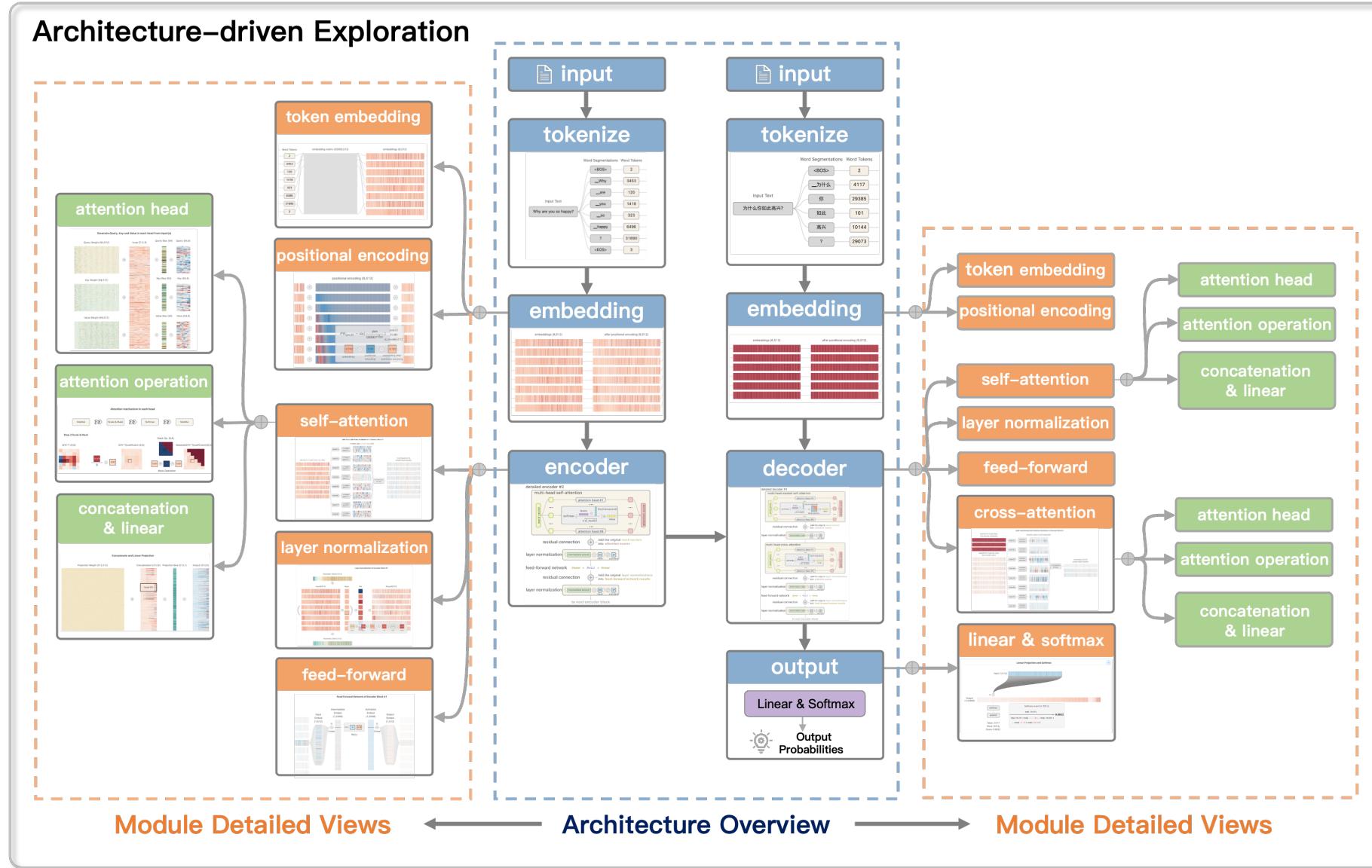


Task-driven Exploration



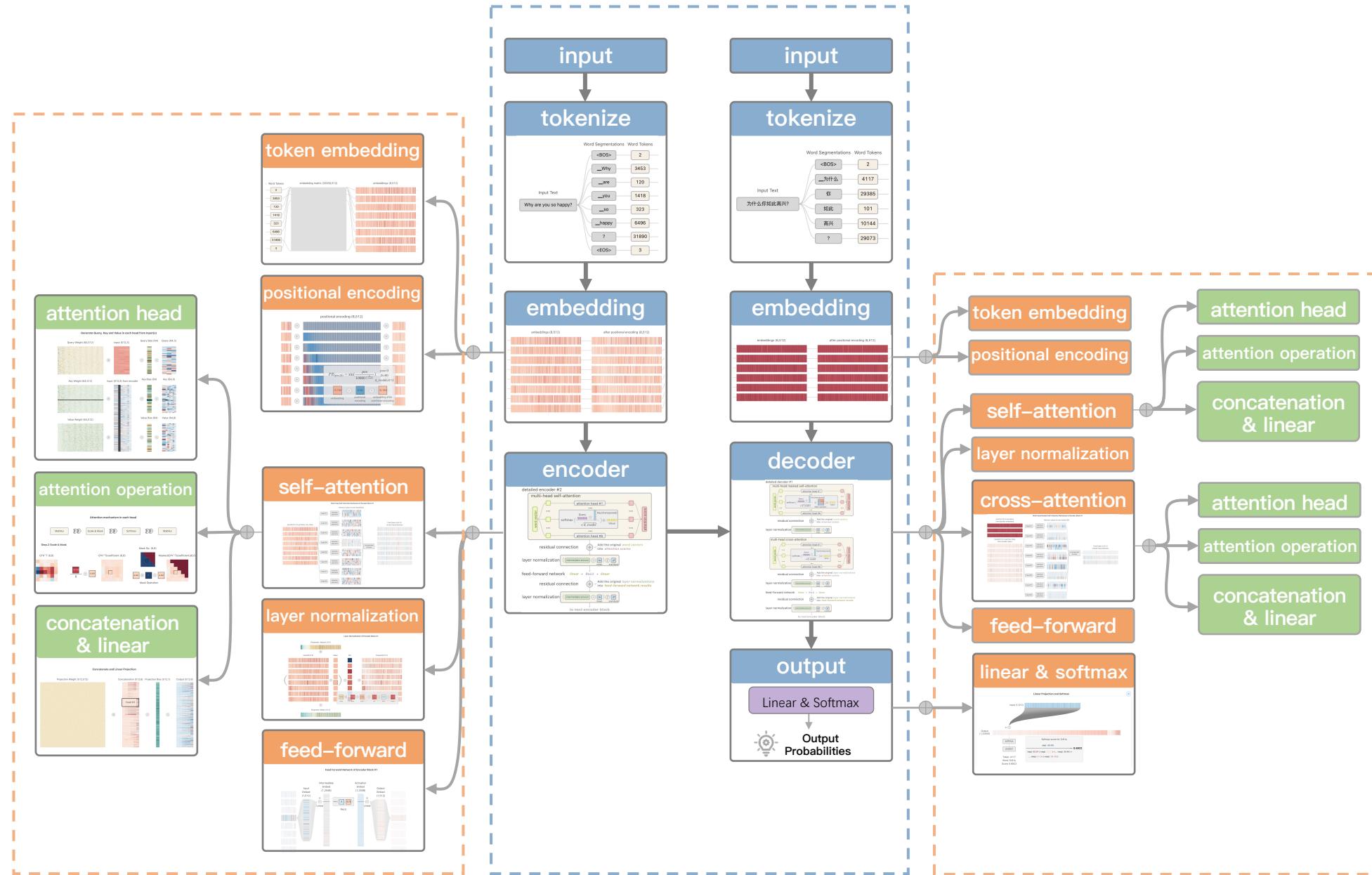


Architecture-driven Exploration





Model Overview



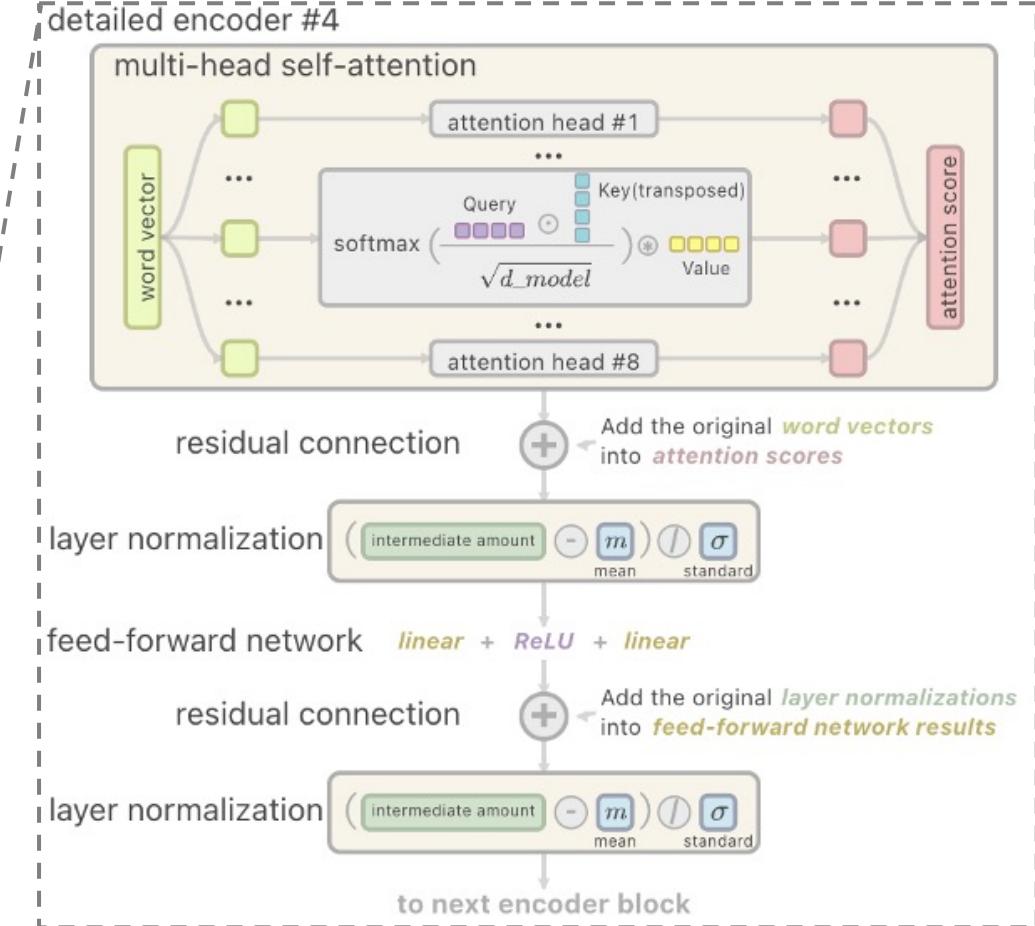
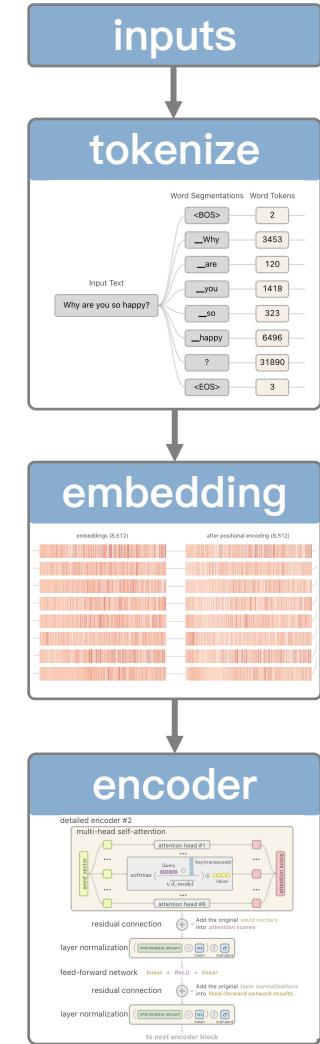
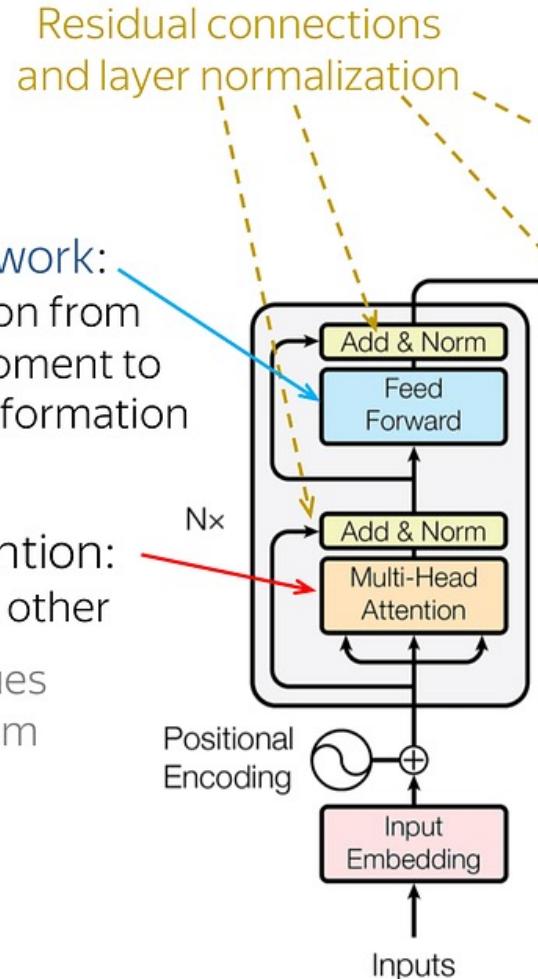


Encoder

Feed-forward network:
after taking information from
other tokens, take a moment to
think and process this information

Encoder self-attention:
tokens look at each other

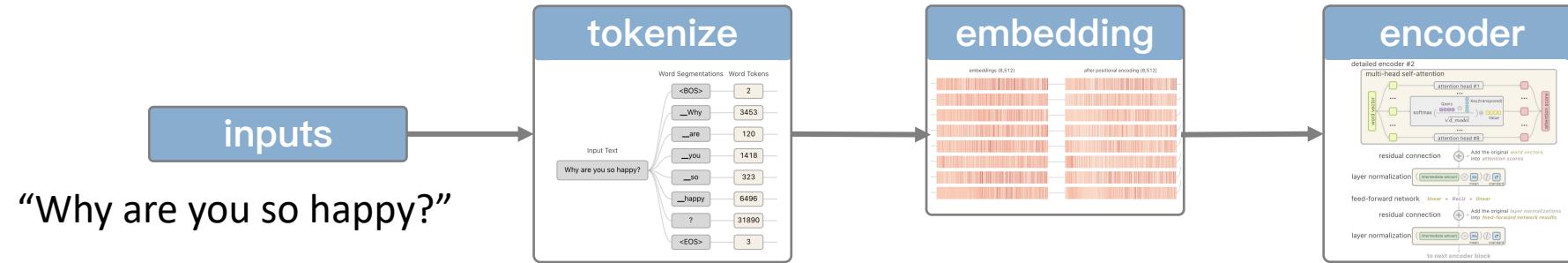
queries, keys, values
are computed from
encoder states



Encoder Block



Encoder



TransforLearn: Interactive Visual Tutorial for the Transformer Model

Input View

Sentence to be translated: ?

Translation View

Translation: 为什么你如此高兴? ?

Current translation: 为什么你如此高兴 < >

Prediction in current iteration: ?

Cumulative Score (Probabilities): **0.039872859081467996**

Architecture View

Sequence Data (blue bar)
Parameter (orange bar)

Word Segments Word Tokens
embeddings (8,512) Click to learn how to get embeddings! after positional encoding (8,512) Click to learn how to get embeddings!

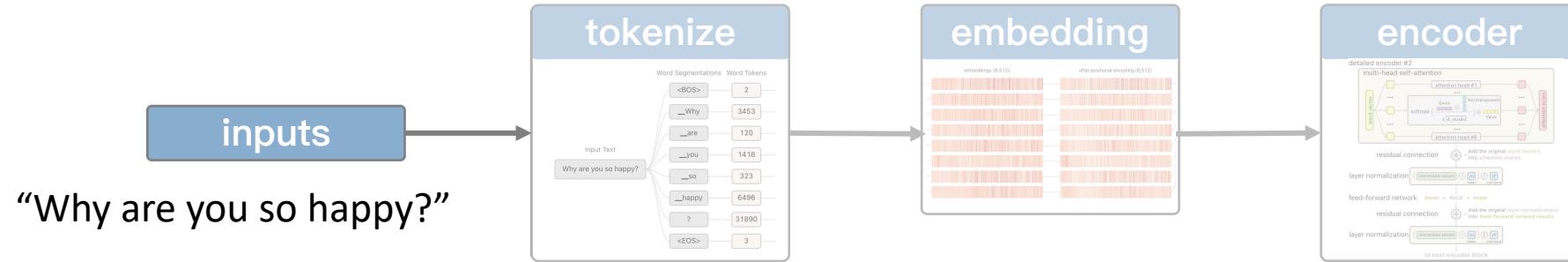
Input Text: Why are you so happy?
Word Segments Word Tokens
encoder blocks
encoder #1
encoder #2
encoder #3
encoder #4
encoder #5
encoder #6
decoder blocks
decoder #1
decoder #2

encoder #1
multi-head self-attention
attention head #1
...
softmax
Query () Key(transposed) () Value ()
...
attention head #8
residual connection
Add the original word vectors into attention scores
layer normalization
feed-forward network linear + ReLU + linear
residual connection
Add the original layer normalizations into feed-forward network results
layer normalization
to next encoder block

Hover over to explore each block!
Click to explore each layer operation!



Encoder



TransforLearn: Interactive Visual Tutorial for the Transformer Model

Input View

Sentence to be translated: Why are you so happy?

Translation View

Translation: 为什么你如此高兴?

Current translation: 为什么你如此高兴

Prediction in current iteration: ?

Cumulative Score (Probabilities): 0.039872859081467996

Architecture View

Sequence Data

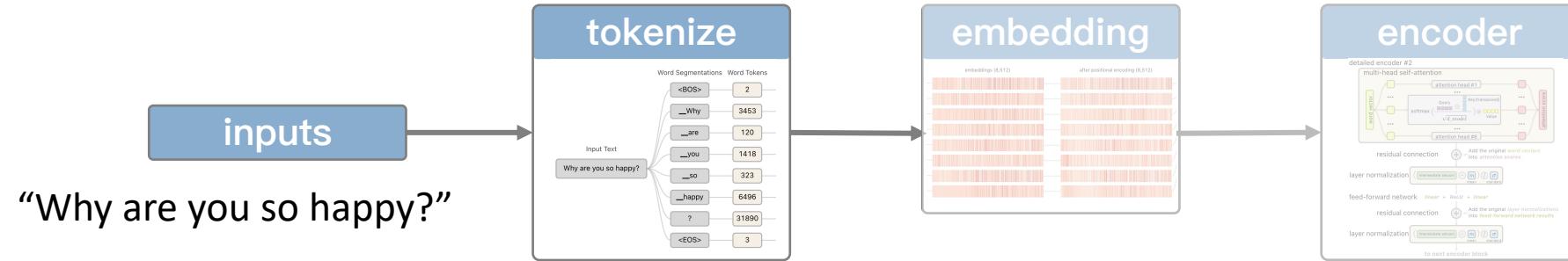
Parameter

Input Text

Why are you so happy?



Encoder



TransforLearn: Interactive Visual Tutorial for the Transformer Model

Input View

Sentence to be translated: Why are you so happy?

Architecture View

Sequence Data
Parameter

Word Segments Word Tokens

Input Text: Why are you so happy?

<BOS> 2
<Why> 3453
<are> 120
<you> 1418
<so> 323
<happy> 6496
? 31890
<EOS> 3

Word Segments Word Tokens

<BOS> 2
<为什么> 4117

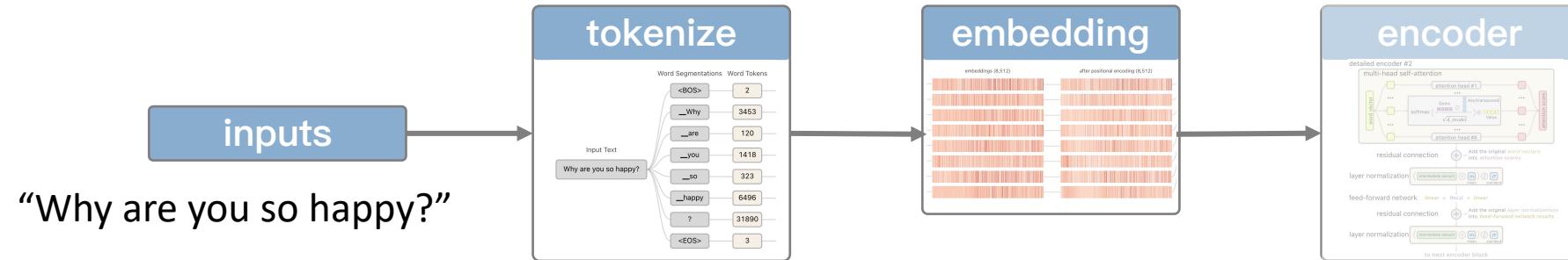
Translation View

Translation: 为什么你如此高兴?
Current translation: 为什么你如此高兴

Prediction in current iteration: ?
Cumulative Score (Probabilities): 0.039872859081467996



Encoder



TransforLearn: Interactive Visual Tutorial for the Transformer Model

Input View

Sentence to be translated: Why are you so happy?

Translation View

Translation: 为什么你如此高兴?
Current translation: 为什么你如此高兴
Prediction in current iteration: ?
Cumulative Score (Probabilities): 0.039872859081467996

Architecture View

Sequence Data (red bar)
Parameter (green bar)

Word Segments Word Tokens embeddings (8,512) after positional encoding (8,512)

Input Text: Why are you so happy?

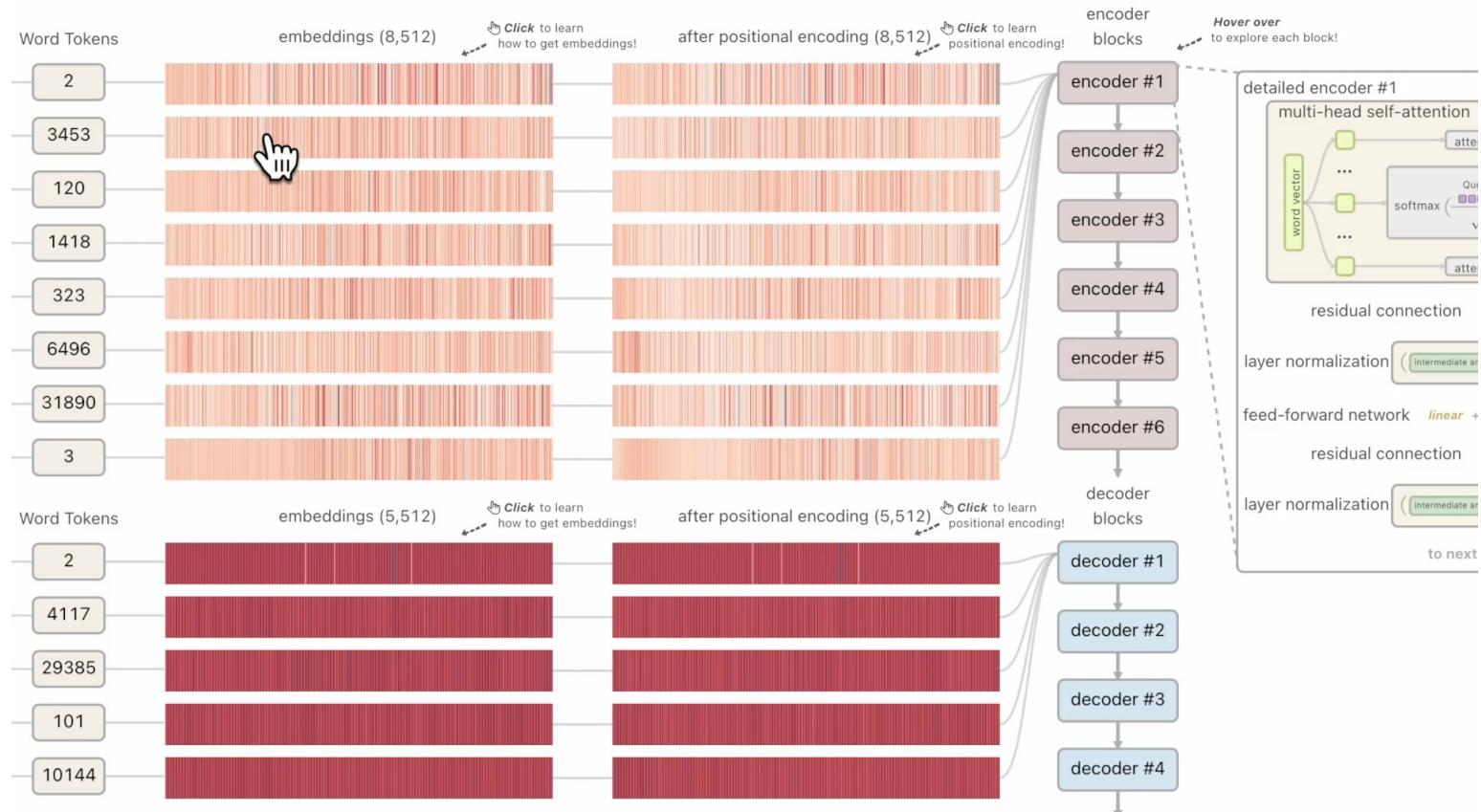
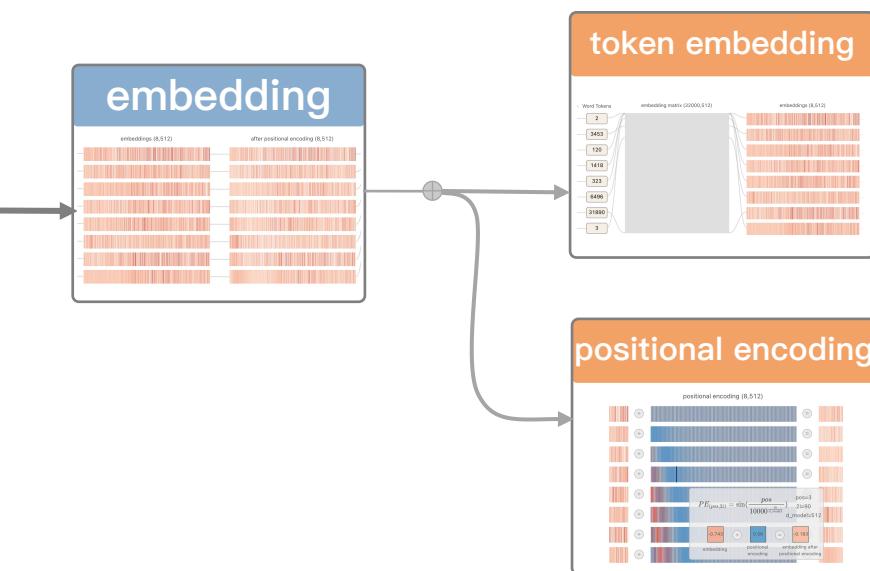
Word Segments Word Tokens embeddings (5,512) after positional encoding (5,512)

为什么 4117

This section provides an interactive visual tutorial for the Transformer Model. It includes three main views: Input View, Translation View, and Architecture View. The Input View shows the sentence "Why are you so happy?" being tokenized into words and their corresponding IDs. The Translation View shows the sentence being translated from English to Chinese. The Architecture View provides a detailed look at the internal structure of the model, showing the flow of data from input tokens through word embeddings and positional encodings, and into the multi-head self-attention and feed-forward network layers of the encoder. The interface also includes a color-coded legend for Sequence Data and Parameters, and a cumulative score for the current iteration.

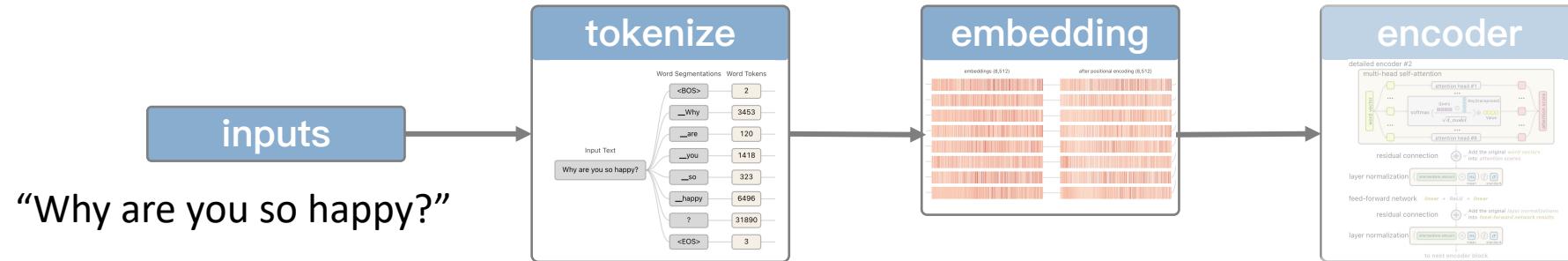


Encoder





Encoder



TransforLearn: Interactive Visual Tutorial for the Transformer Model

Input View

Sentence to be translated: Why are you so happy?

Translation View

Translation: 为什么你如此高兴? Current translation: 为什么你如此高兴 Prediction in current iteration: ? Cumulative Score (Probabilities): 0.039872859081467996

Architecture View

Sequence Data (red bar)

Parameter (blue bar)

Word Segments Word Tokens embeddings (8,512) after positional encoding (8,512)

Input Text: Why are you so happy?

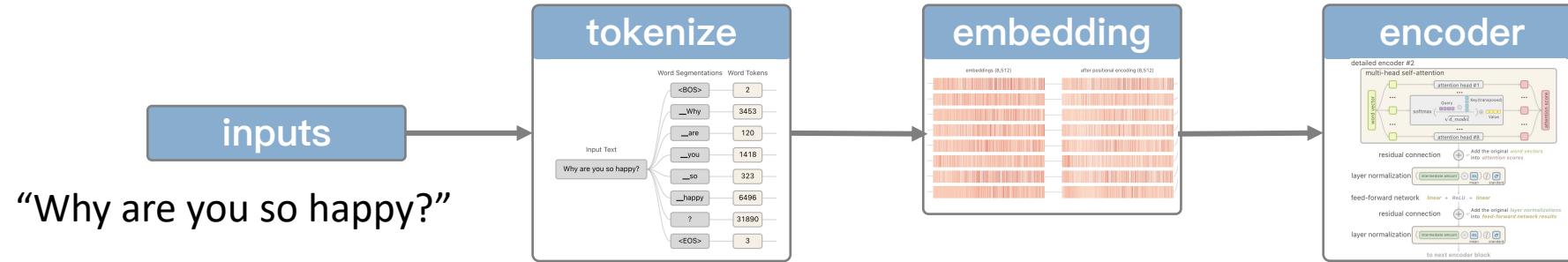
Word Segments Word Tokens embeddings (5,512) after positional encoding (5,512)

为什么 4117

The Architecture View displays the internal structure of the Transformer model. It shows the input text "Why are you so happy?" being tokenized into word segments and mapped to word tokens (<BOS>, 2, <Why>, 3453, <are>, 120, <you>, 1418, <so>, 323, <happy>, 6496, ?, 31890, <EOS>, 3). These tokens are then processed by two parallel embedding layers: one producing 8-dimensional embeddings (8,512) and another producing 5-dimensional embeddings (5,512). Each embedding layer is followed by a "Click to learn how to get embeddings!" button. The 8-dimensional embeddings are then processed by a multi-head self-attention mechanism, which consists of attention heads (1 to 8), residual connections, layer normalization, and a feed-forward network. The 5-dimensional embeddings are also processed by a similar mechanism. The final outputs are "embeddings (8,512)" and "embeddings (5,512)", followed by their respective "after positional encoding" states (8,512 and 5,512).



Encoder



TransforLearn: Interactive Visual Tutorial for the Transformer Model

Input View

Sentence to be translated: Why are you so happy?

Translation View

Translation: 为什么你如此高兴? Current translation: 为什么你如此高兴 Prediction in current iteration: ? Cumulative Score (Probabilities): 0.039872859081467996

Architecture View

Sequence Data (blue bar) Parameter (orange bar)

Input Text: Why are you so happy?

Word Segments and Word Tokens for the input sentence:

Word Segments	Word Tokens
<BOS>	2
_Why	3453
_are	120
_you	1418
_so	323
_happy	6496
?	31890
<EOS>	3

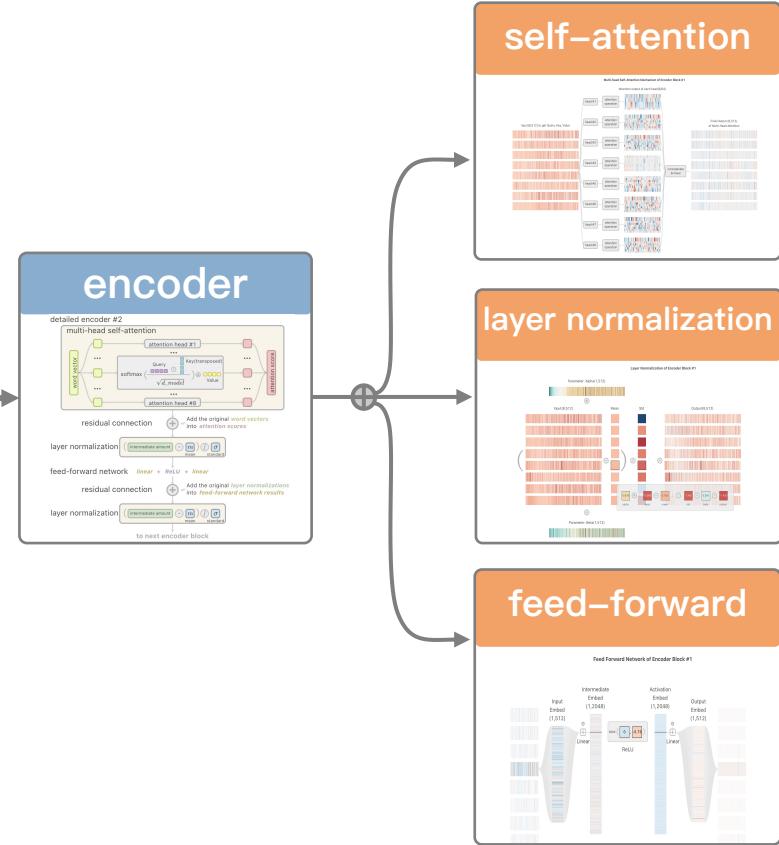
embeddings (8,512) and after positional encoding (8,512) visualizations.

encoder blocks (encoder #1 to encoder #6) and decoder blocks (decoder #1 to decoder #2) are shown, along with detailed views of the multi-head self-attention mechanism and feed-forward network operations.

Annotations include: Click to learn how to get embeddings!, Click to learn how to learn positional encoding!, Hover over to explore each block!, Click to explore each layer operation!, and various mathematical formulas for layer normalization and residual connections.



Encoder



TransforLearn: Interactive Visual Tutorial for the Transformer Model

Translation View

Translation: 为什么你如此高兴? Click to learn how to get embeddings!

Current translation: 为什么你如此高兴 < >

Prediction in current iteration: ?

Cumulative Score (Probabilities): 0.039872859081467996

Word Segments Word Tokens embeddings (8,512) Click to learn how to get embeddings! after positional encoding (8,512) Click to learn positional encoding! encoder blocks Hover over to explore each block!

encoder #1
encoder #2
encoder #3
encoder #4
encoder #5
encoder #6

Word Segments Word Tokens embeddings (5,512) Click to learn how to get embeddings! after positional encoding (5,512) Click to learn positional encoding! decoder blocks

decoder #1
decoder #2
decoder #3
decoder #4
decoder #5

detailed encoder #1 multi-head self-attention Click to explore each layer operation!

word vector ... attention head #1 ... attention score

Query \odot Key(transposed) \odot Value $\sqrt{d_{\text{model}}}$

residual connection Add the original word vectors into attention scores

layer normalization m σ

feed-forward network linear + ReLU + linear

residual connection Add the original layer normalizations into feed-forward network results

layer normalization m σ

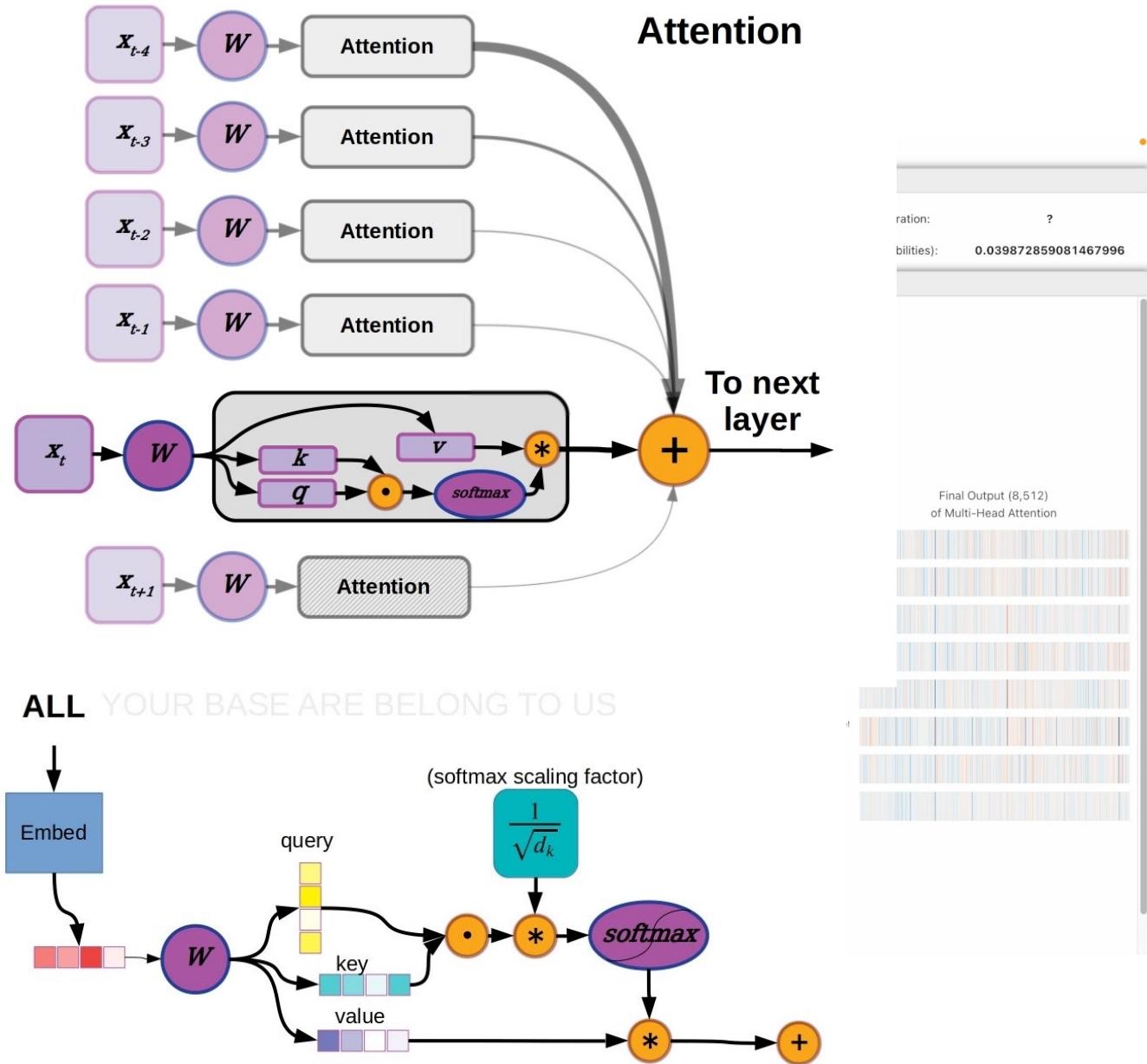
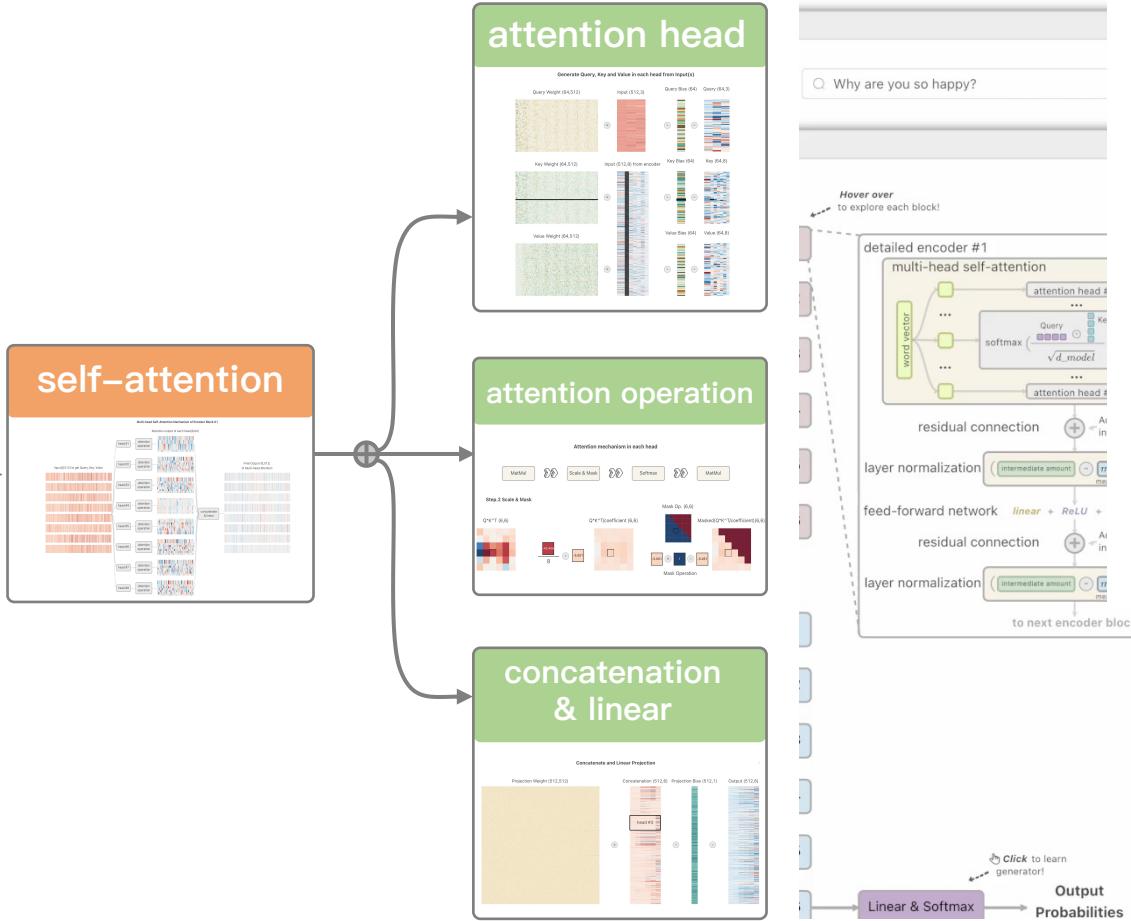
to next encoder block

Click to learn generator!

This figure illustrates the Transformer architecture, specifically focusing on the multi-head self-attention mechanism. It shows two parallel processing paths: one for English input and one for Chinese input. The English path consists of six encoder blocks, while the Chinese path consists of five decoder blocks. Each block takes word embeddings and applies positional encoding. The multi-head self-attention mechanism is detailed for the first encoder block, showing how query, key, and value matrices are derived from word vectors, scaled by the square root of the dimension, and combined through a softmax function. The resulting attention scores are then multiplied with the value matrix and summed. This process is followed by residual connections and layer normalization. The final output is passed to the next encoder block or the subsequent decoder block. The figure also includes a 'detailed encoder #1' box that provides a step-by-step breakdown of the multi-head self-attention process.

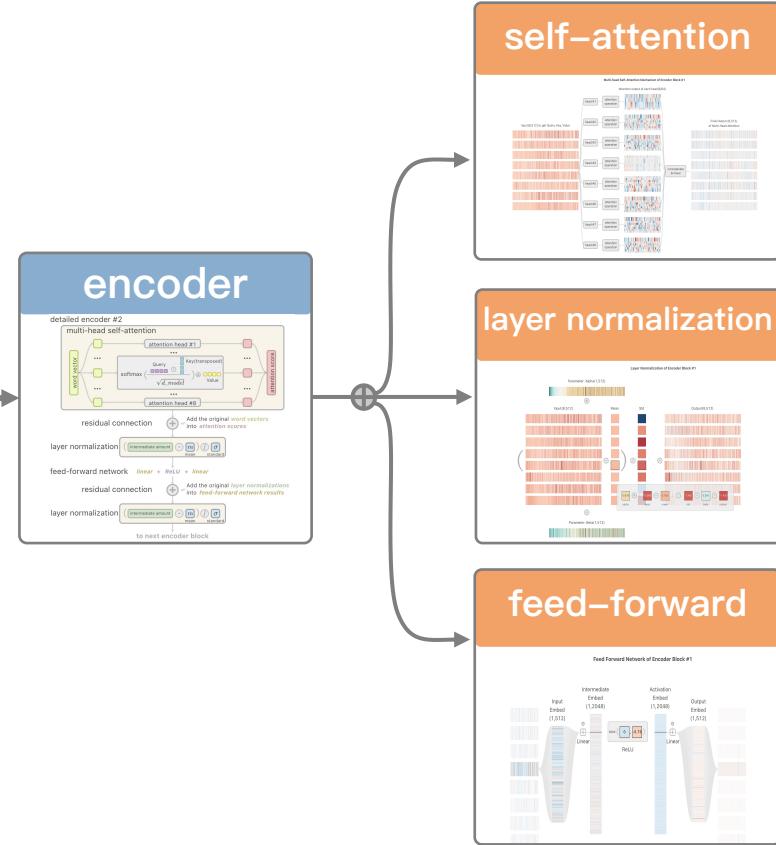


Encoder - Attention





Encoder



Translation View

Translation: 为什么你如此高兴? 为什么你如此高兴

Current translation: 为什么你如此高兴

Prediction in current iteration: ?

Cumulative Score (Probabilities): 0.039872859081467996

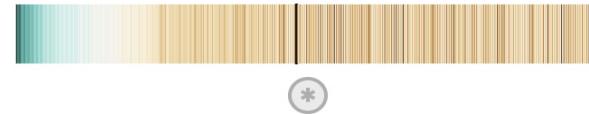
The figure illustrates the Transformer architecture for the sentence "Why are you so happy?" in Chinese and English. It shows the input words mapped to tokens, followed by word embeddings and positional encodings. The input tokens are: <BOS>, Why, are, you, so, happy, ?, <EOS>. The embeddings and positional encodings are shown as colored bars. The architecture consists of six encoder blocks and five decoder blocks. A detailed view of encoder block #1 is provided, showing the multi-head self-attention mechanism. Each block includes residual connections and layer normalization. The final output is a cumulative score of 0.039872859081467996.



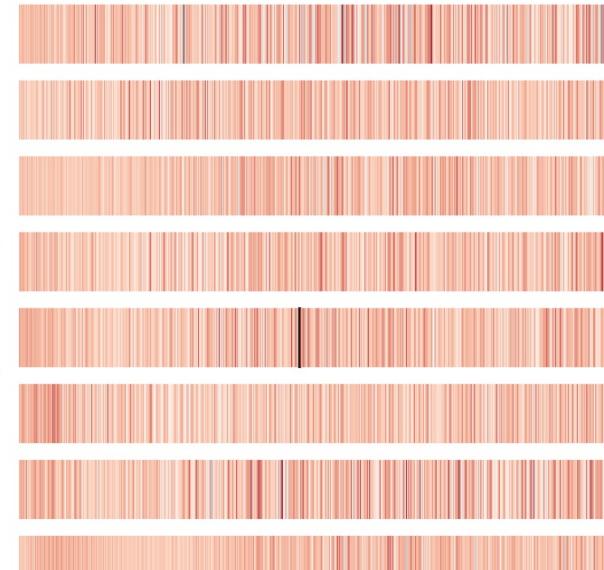
Encoder

Layer Normalization of Encoder Block #1

Parameter: Alpha(1,512)



Input(8,512)



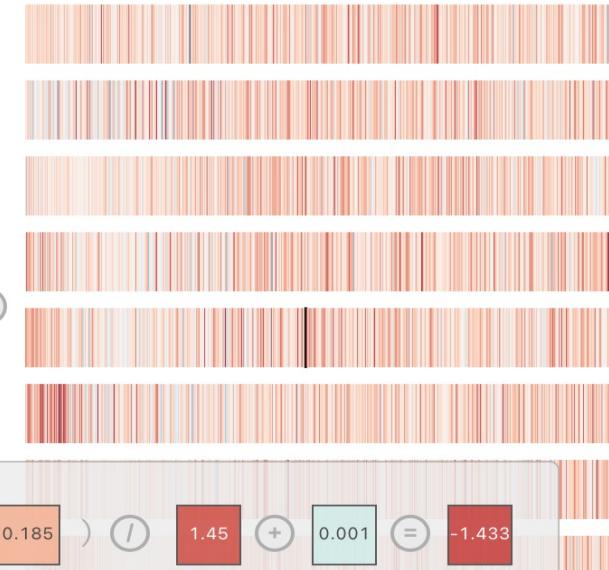
Mean



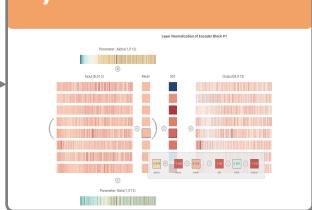
Std



Output(8,512)



layer normalization



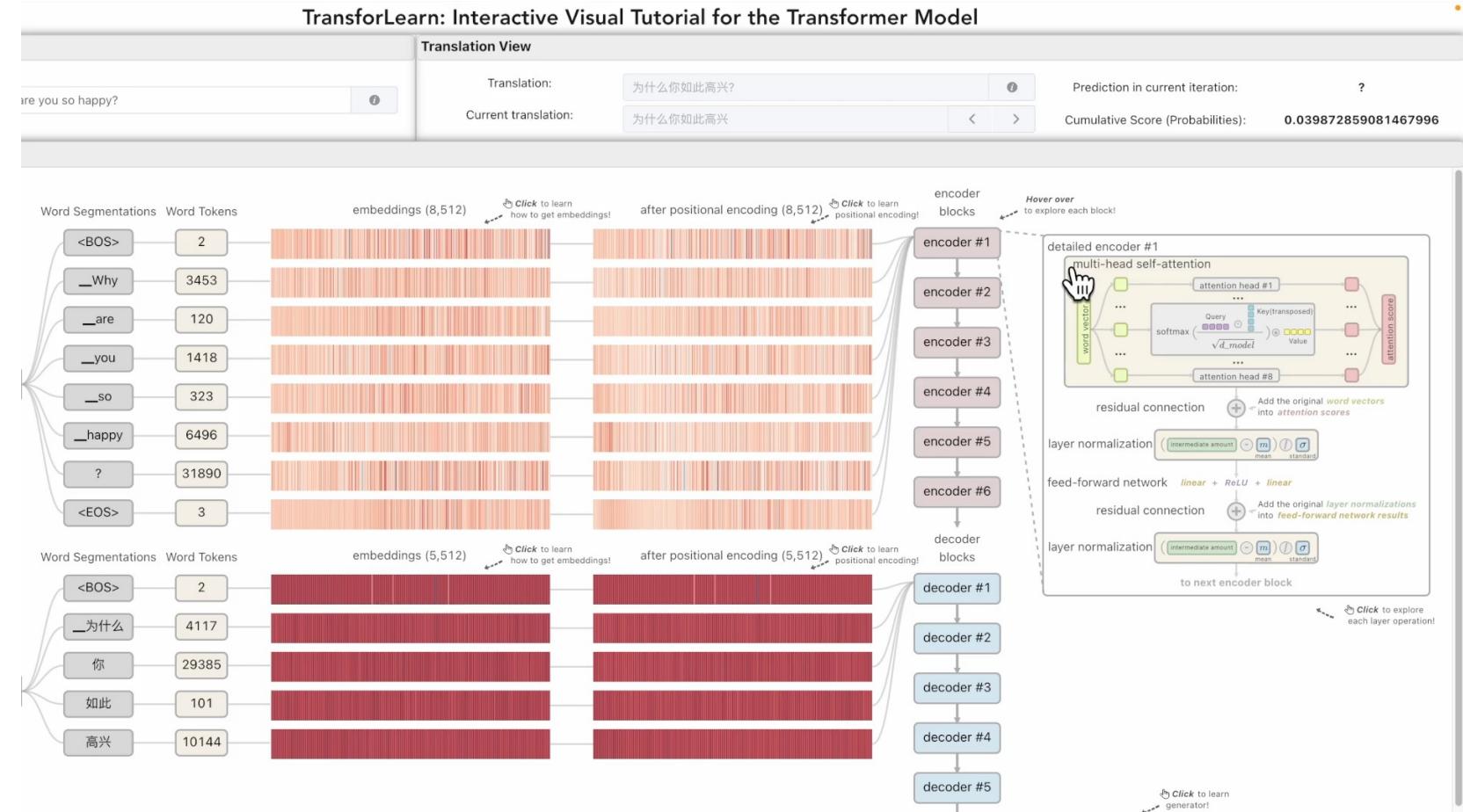
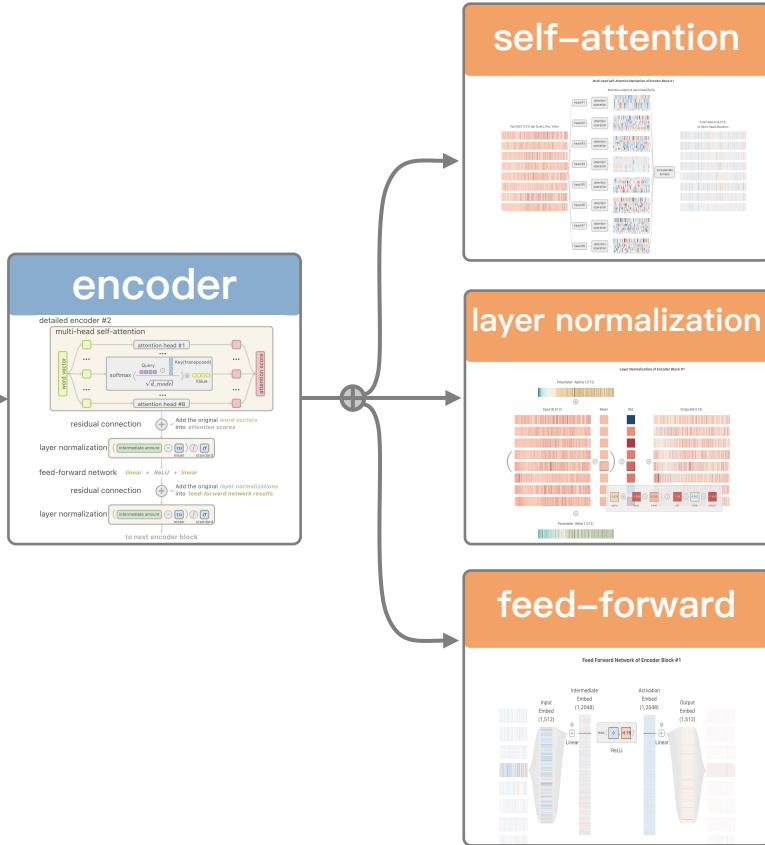
0.678 * -2.885 - 0.185) / 1.45 + 0.001 = -1.433

Parameter: Beta(1,512)



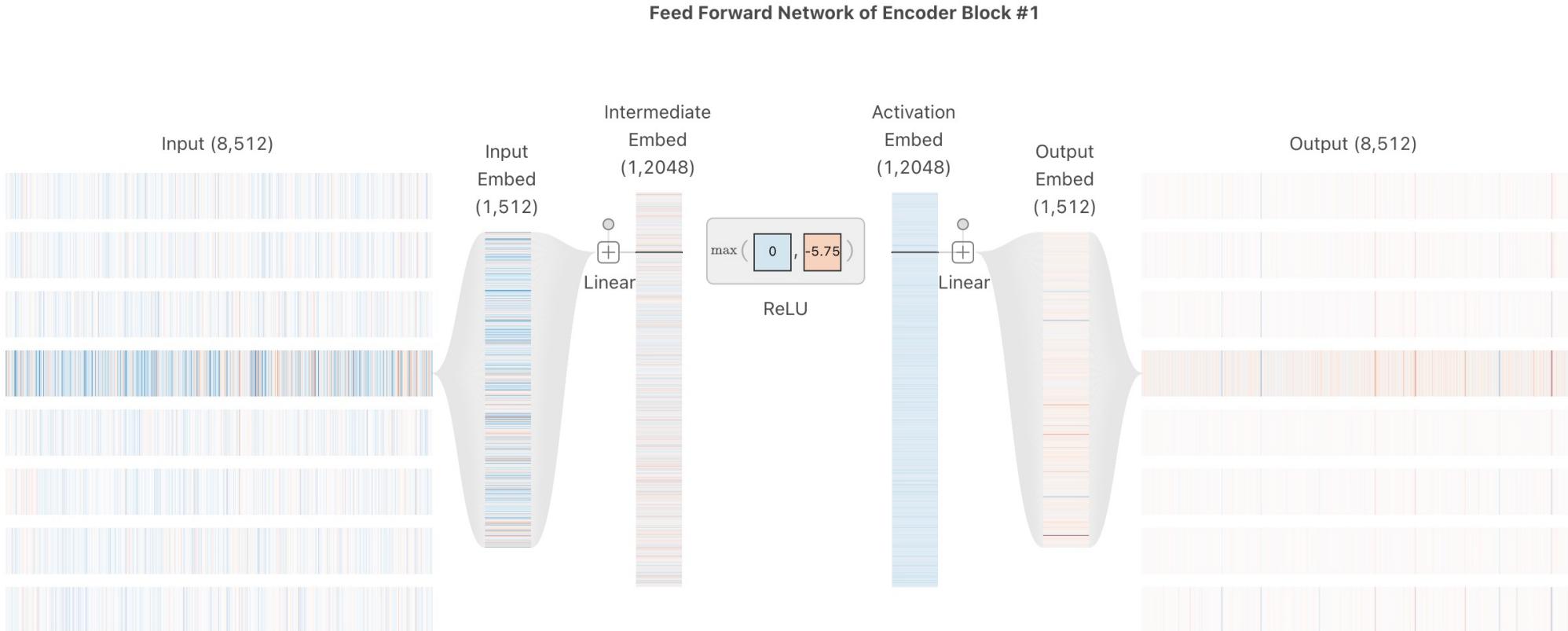
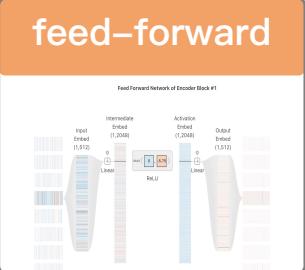


Encoder



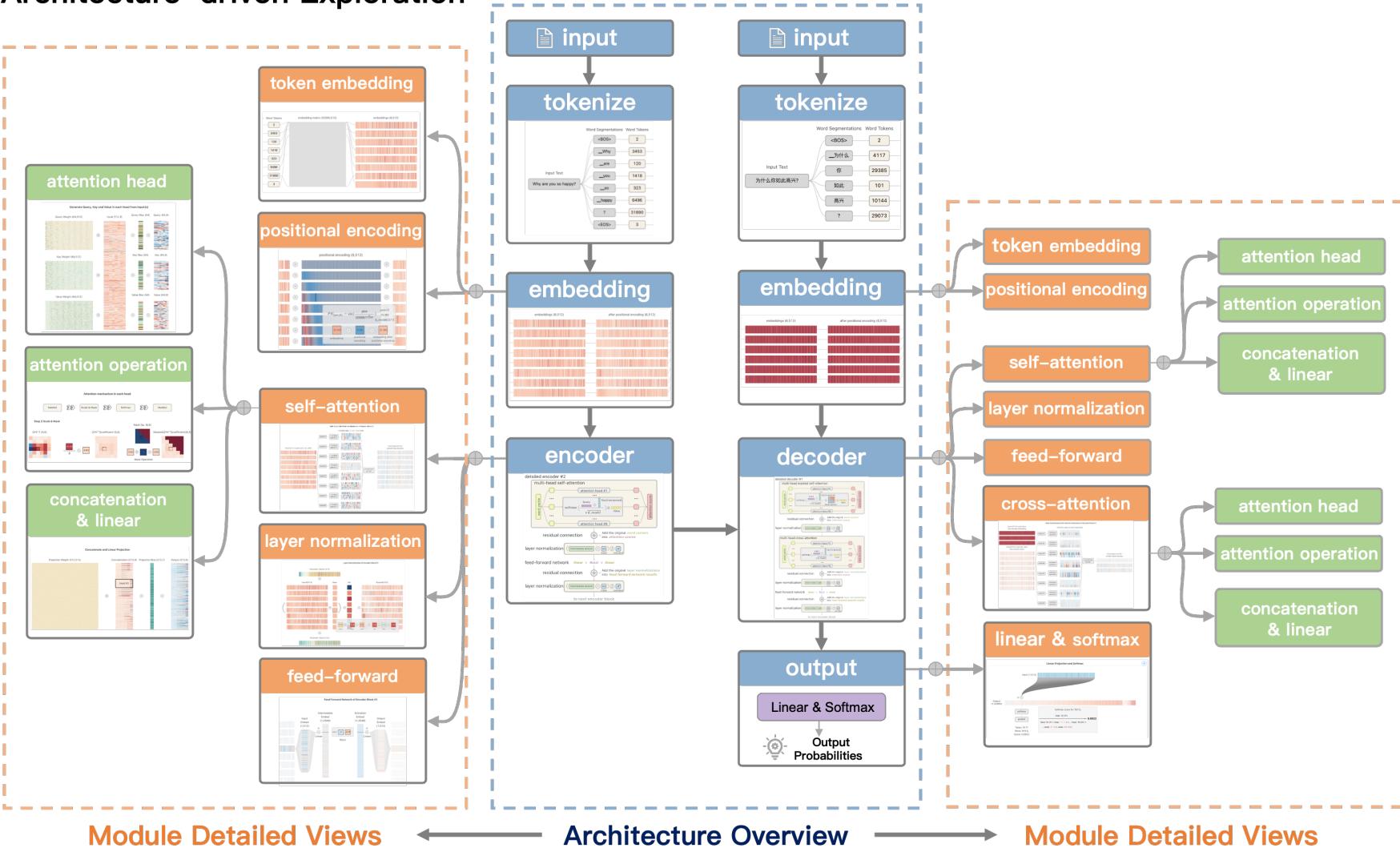


Encoder

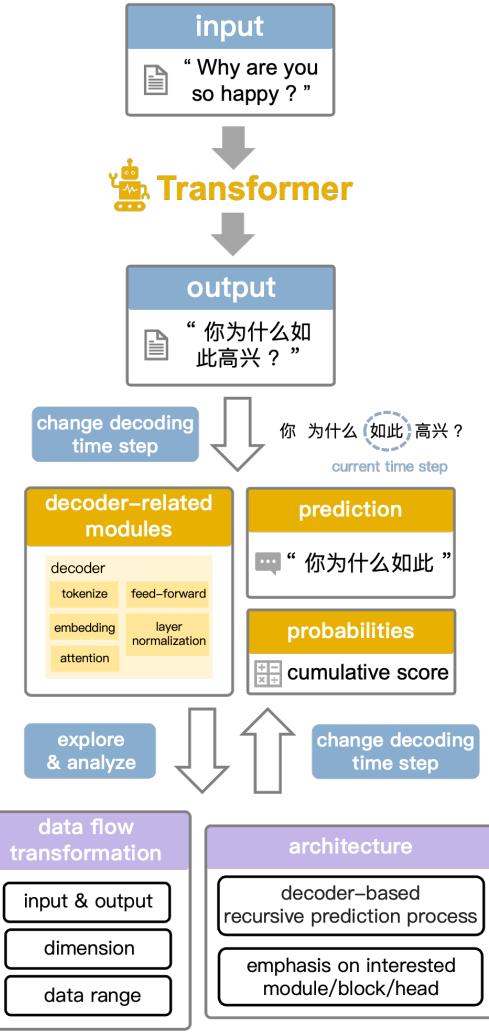


Visual Design - Overview

Architecture-driven Exploration

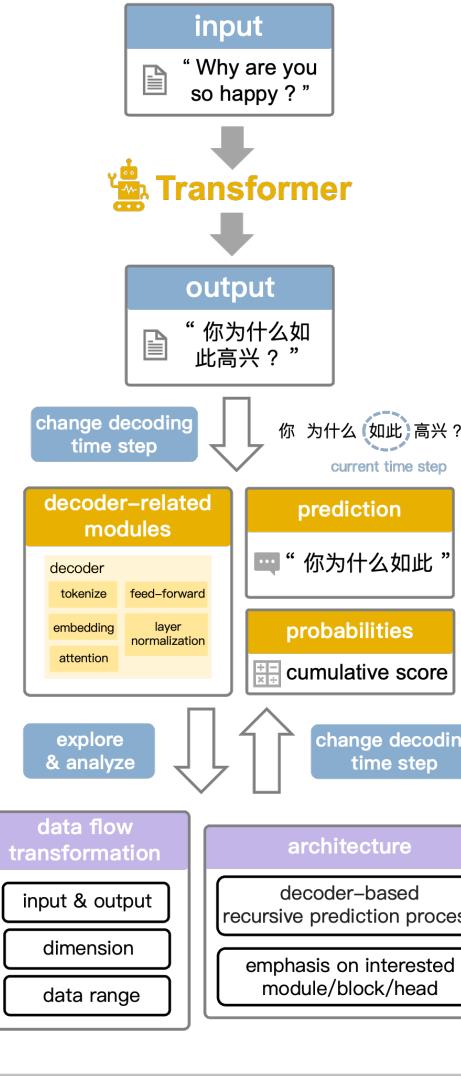


Task-driven Exploration



Task-driven Exploration

Task-driven Exploration



Explore data flow changes

- Input and output, data dimension, data range

Analyze structural features

- Decoding time step -> translation progress
- Focus on a specific module or head



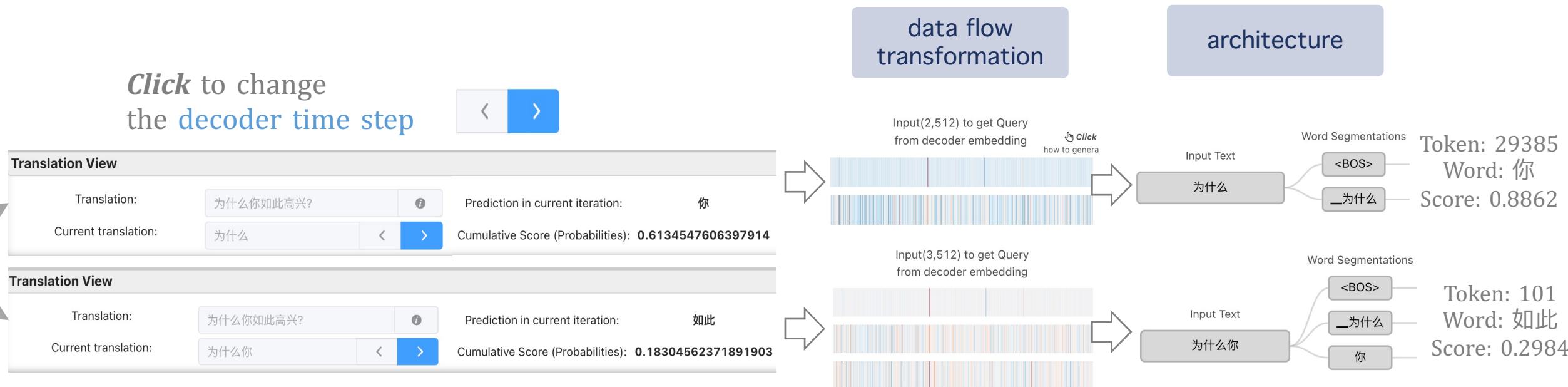
Task-driven Exploration

Explore data flow changes

- Input and output, data dimension, data range

Analyze structural features

- Decoding time step -> translation progress
- Focus on a specific module or head





Usage Scenario

Self-study guidance for a beginner

- utilize Transformer to extract features from sequence data
- the concept and generation process of the Q, K, and V matrices
- the use of decoders for prediction

Teaching aid for lectures

- better summarize and present the teaching points
- increases the practicality and vividness of the entire teaching process

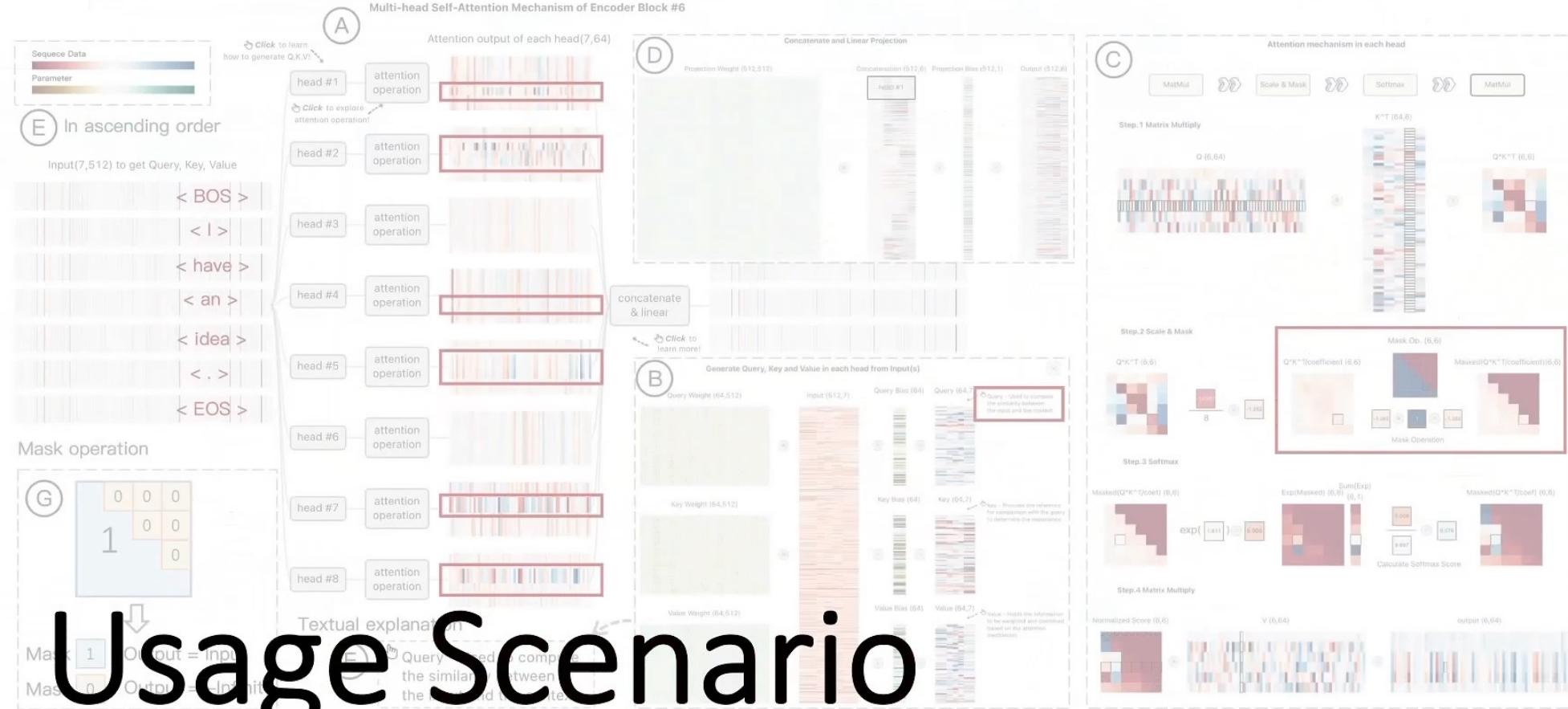


Usage Scenario





Usage Scenario



Usage Scenario

Self-study guidance for beginners

Evaluation

User-controlled Experiment



R-1 visual summary

R-2 interactive interface

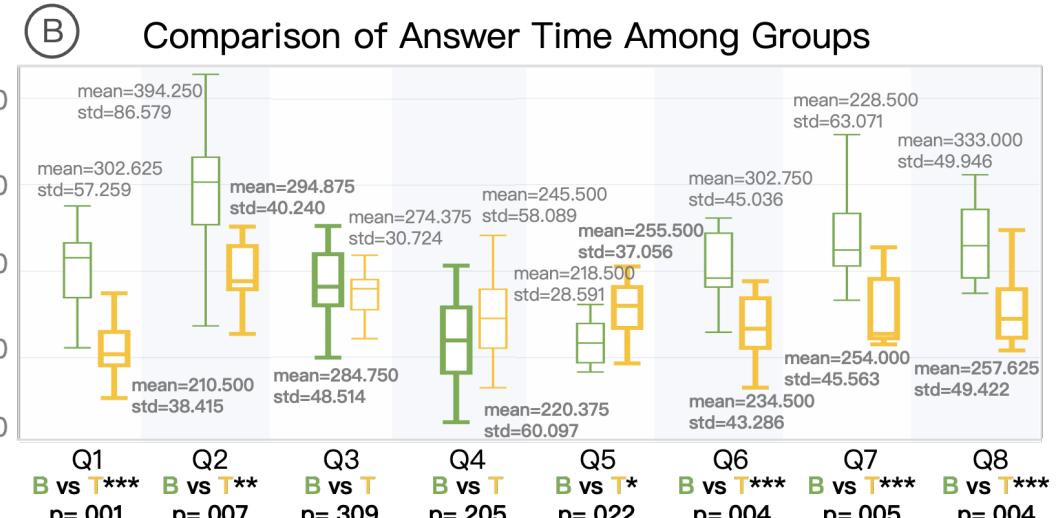
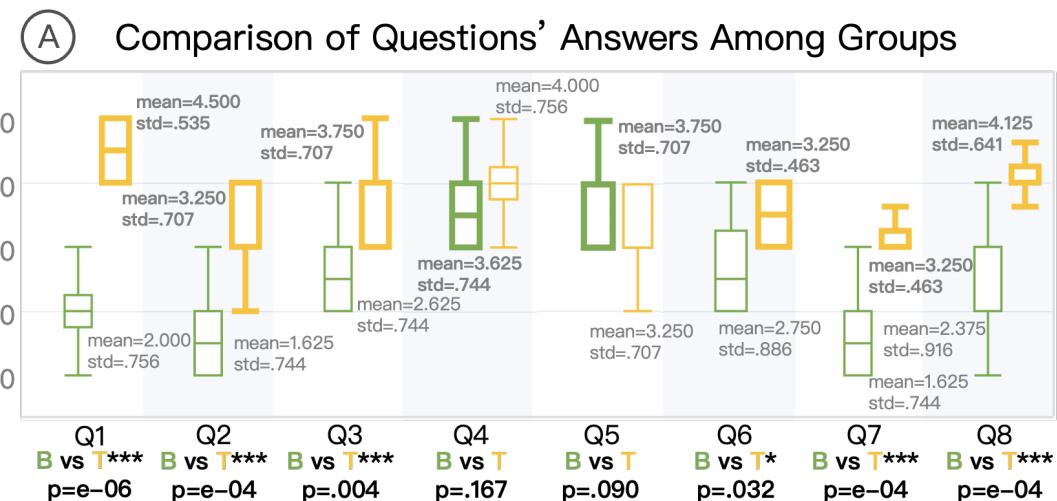
R-3 exploration mode

R-4 self-directed & immersive

Level	Goal	Question
easy	G1	Q1: Components and data flow of feed-forward network.
easy	G3	Q2: Identify key words from attention matrix.
easy	G3	Q3: Final output in translation task and its derivation.
medium	G1	Q4: Differences between cross- and self-attention.
medium	G2	Q5: Add & LN significance and implementation.
medium	G1	Q6: Parallelism in Transformer.
hard	G2	Q7: Reasons for scaling before softmax.
hard	G2	Q8: Process of calculating PE & variation with position.

Evaluation

- improve users' **understanding** of structures and tasks
- bring more **activity**, **autonomy** and **divergent thinking**
- enhancing users' **efficiency** in learning through a broader coverage and enhanced interaction



(C) Comparison of Learning Efficiency Index Among Groups

$E_{GroupX,i}$	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$	$i = 8$	Mean	Std
$X = B$	0.737	1.005	0.680	0.839	0.981	0.824	0.965	0.851	0.851	0.121
$X = T$	1.421	1.702	1.631	1.402	1.263	1.385	1.381	1.542	1.466	0.146



Evaluation

User interviews

Implication

- Usability and effectiveness.
- Validating the knowledge for experts.

Limitation

- Different appropriate learning resources for different needs.
- Need for more instructions, animations, and comparisons.



VIS 2023

Thanks for your listening !

TransforLearn: <https://trans-for-learn.github.io/>

Welcome to our homepage: <http://fduvis.net/>

Email: leenagao0430@gmail.com



TransforLearn

Interactive Visual Tutorial for the Transformer Model

Lin Gao¹, Zekai Shao¹, Ziqing Luo¹, Haibo Hu², Cagatay Turkay³, Siming Chen^{1,4}

¹School of Data Science, Fudan University

²School of Big Data & Software Engineering, Chongqing University

³Centre for Interdisciplinary Methodologies, University of Warwick

⁴Shanghai Key Laboratory of Data Science

Paper Code Demo

