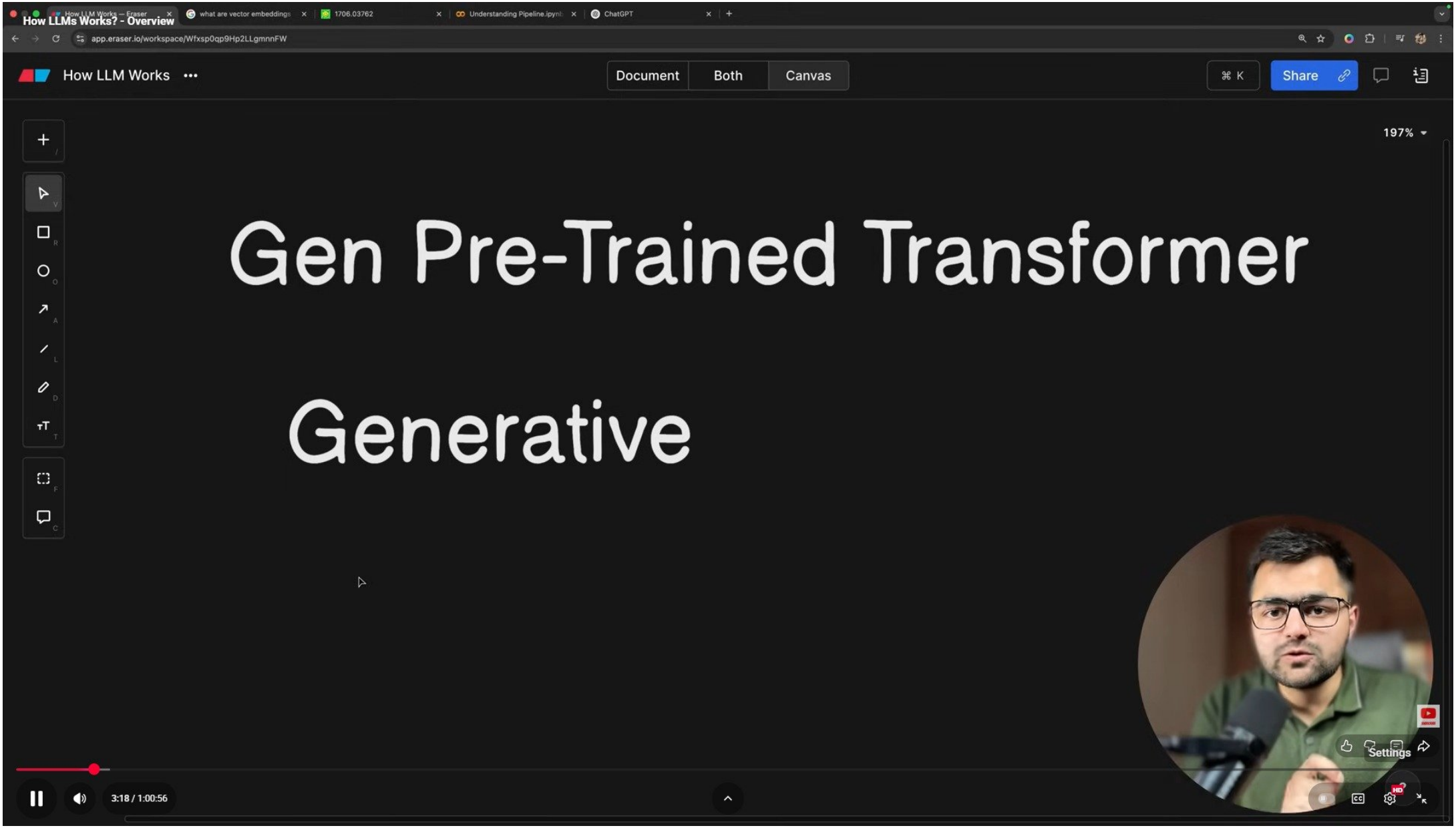

How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
⌘ K
Share
197%
Gen Pre-Trained Transformer
Generative
3:18 / 1:00:56
Settings

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

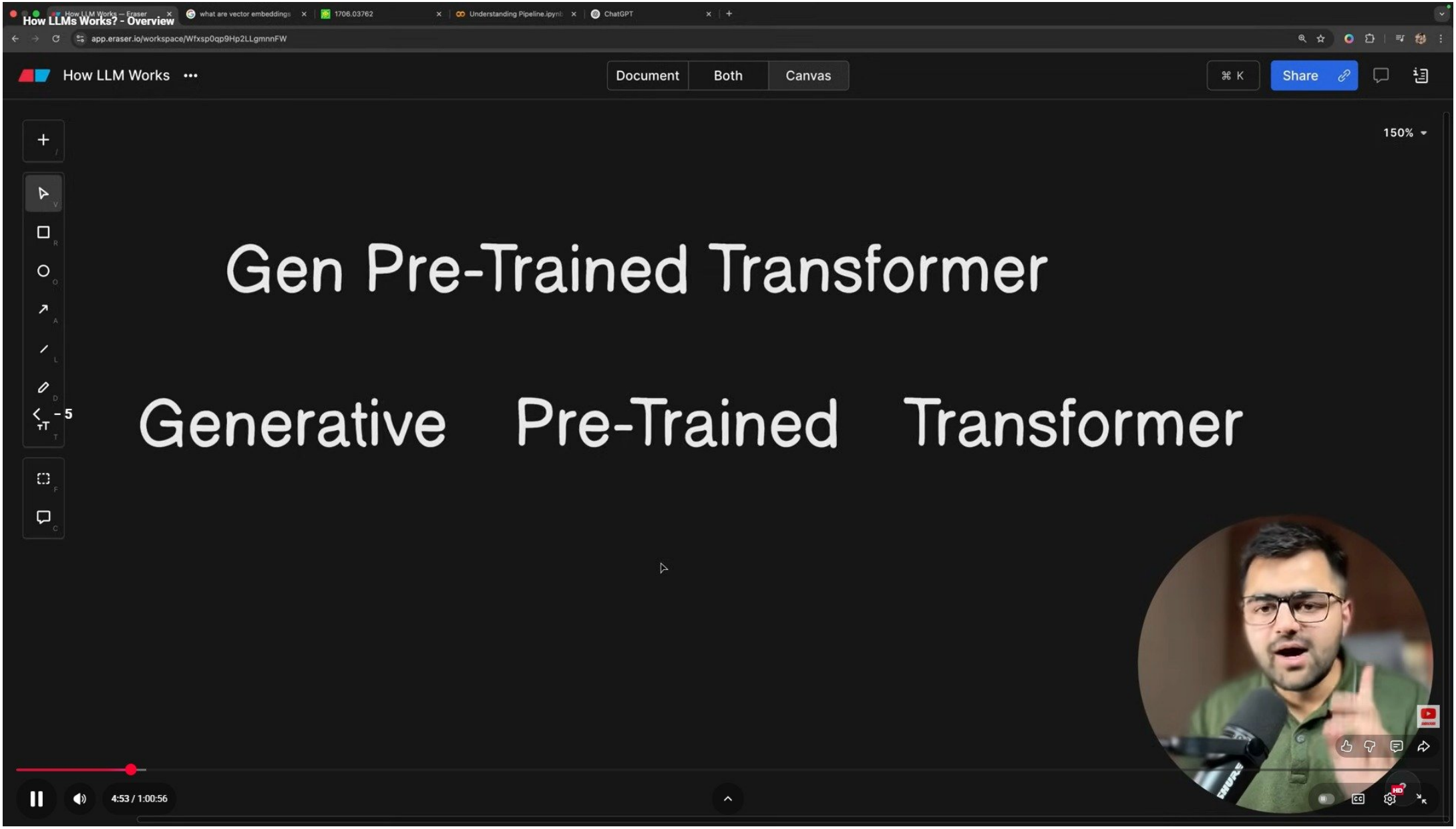
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

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


How LLMs Works? - Overview
app.eraser.io/workspace/WfxsP0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
Share
150%
Gen Pre-Trained Transformer
Generative Pre-Trained Transformer
4:53 / 1:00:56

How LLMs Works? - Overview

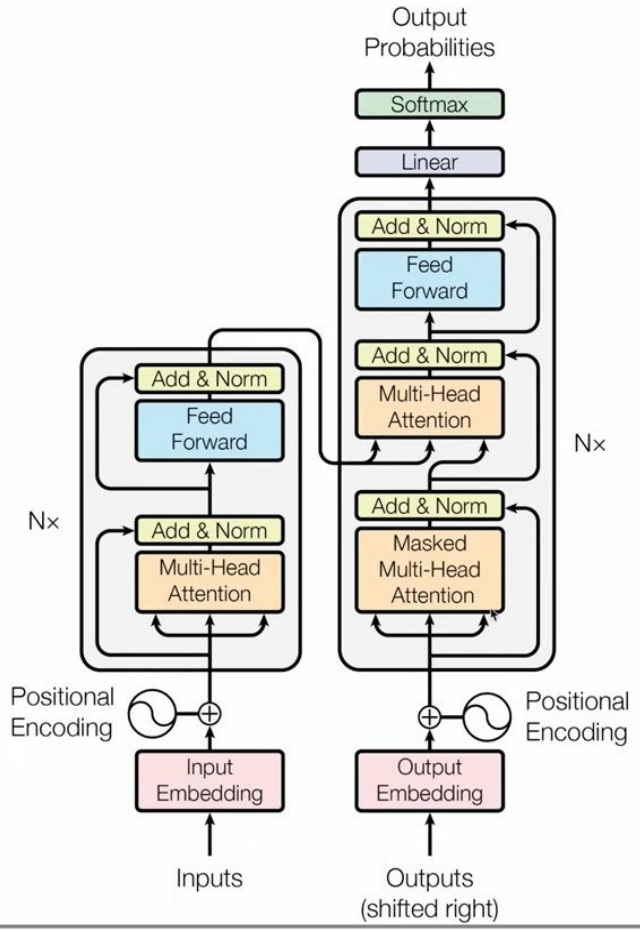
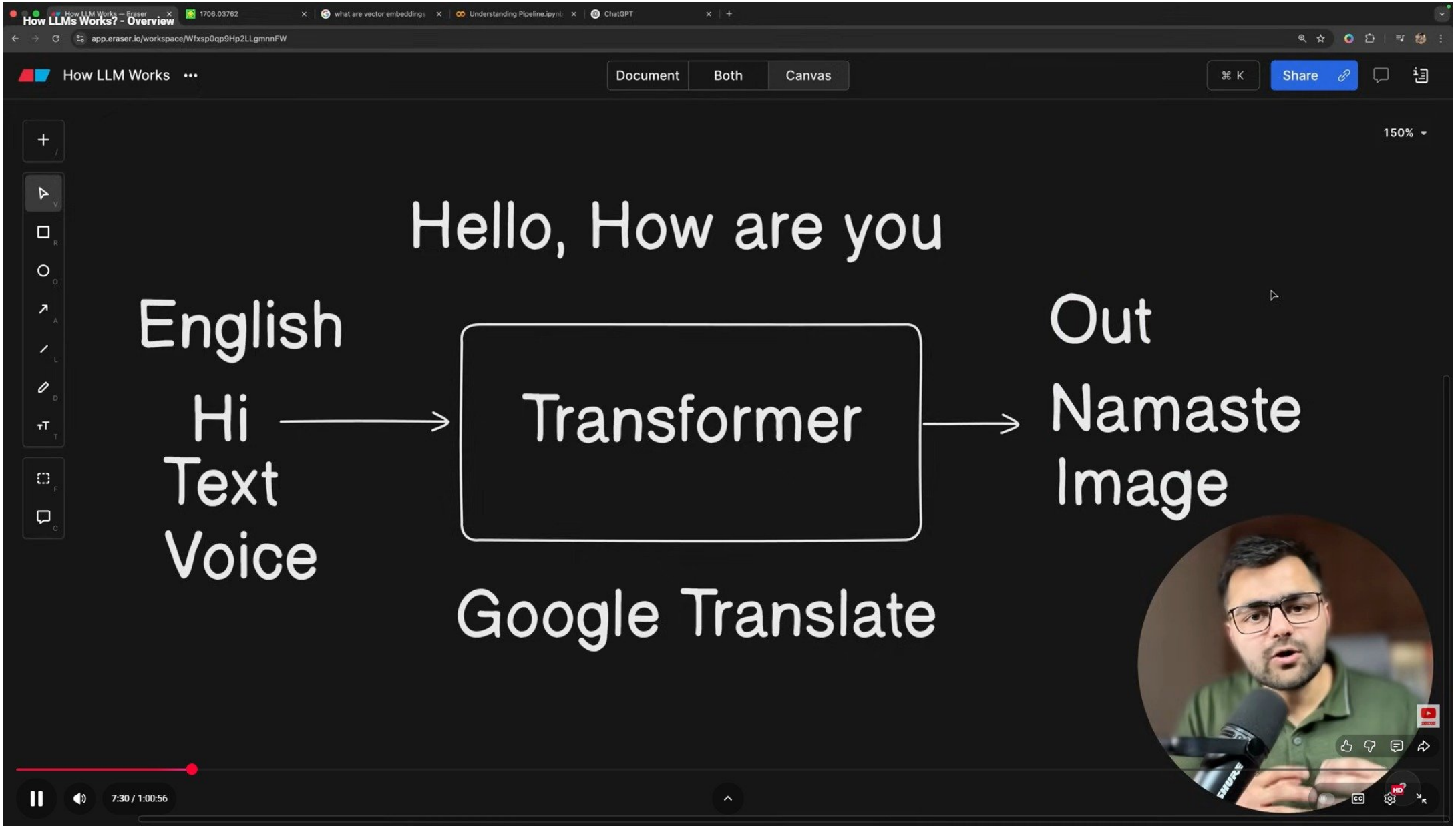
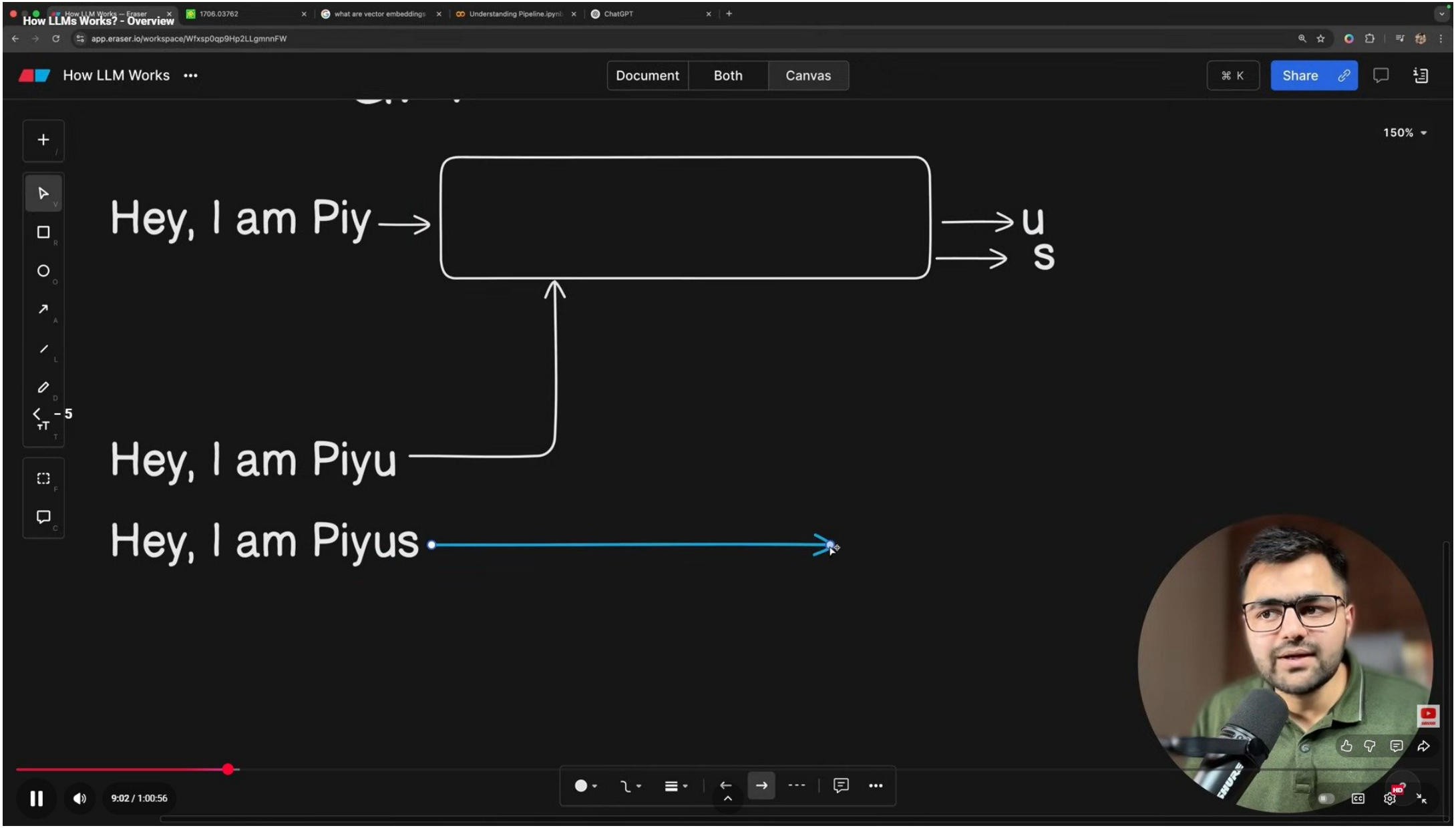
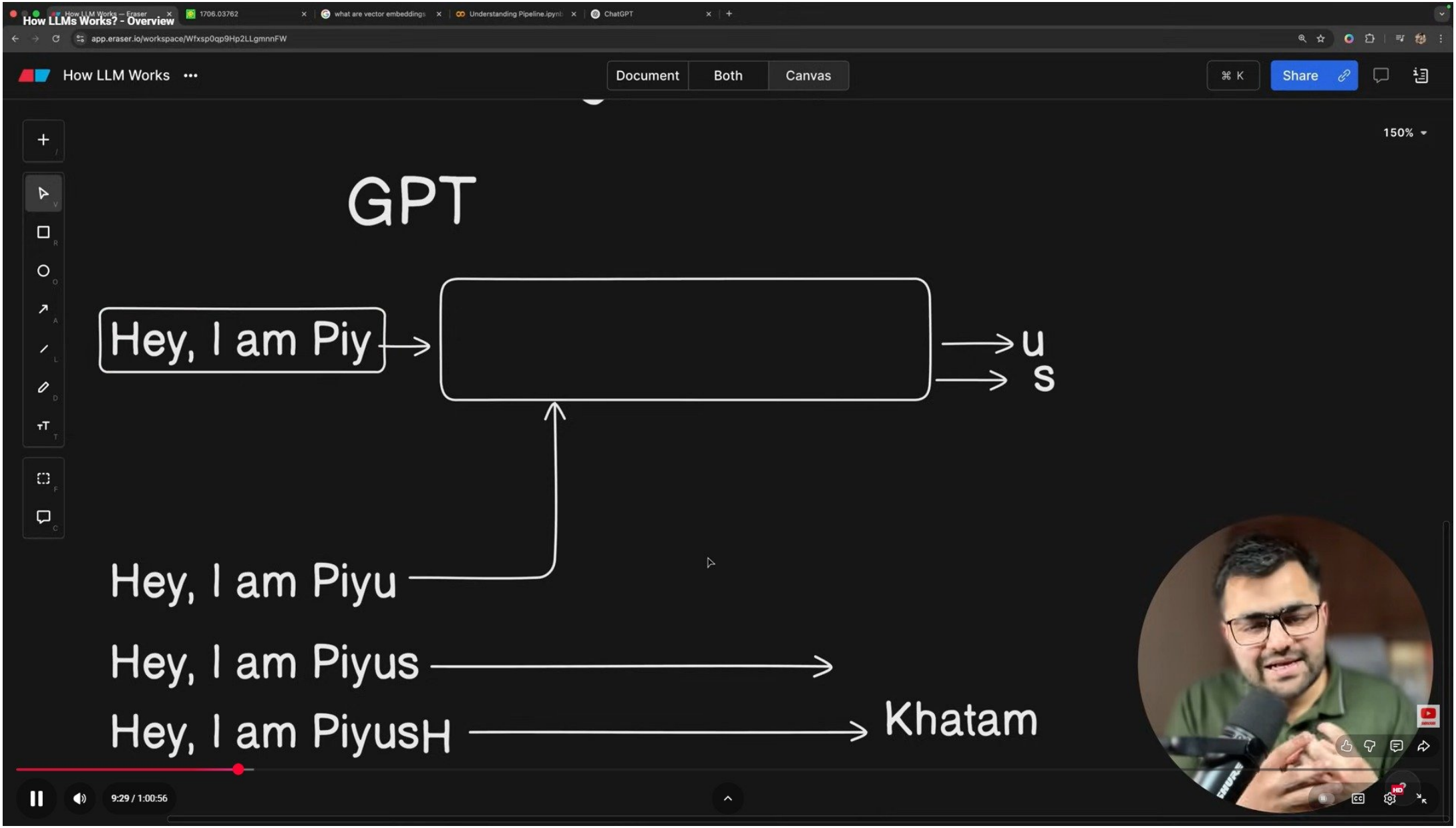


Figure 1: The Transformer - model architecture.


How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
Share
150%
Hello, How are you
English
Hi
Text
Voice
Transformer
Out
Namaste
Image
Google Translate
7:30 / 1:00:56


How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
Share
150%
Hey, I am Piy
u
s
Hey, I am Piyu
Hey, I am Piyus
9:02 / 1:00:56


How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
Share
150%
GPT
Hey, I am Piy
u
s
Hey, I am Piyu
Hey, I am Piyus
Hey, I am PiyusH
Khatam
9:29 / 1:00:56

How LLMs Works? - Overview

app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW

How LLM Works

Document | Both | Canvas

⌘ K | Share

149%

Hey, I am PiyusH → Khatam

a-1
b-2
c-3

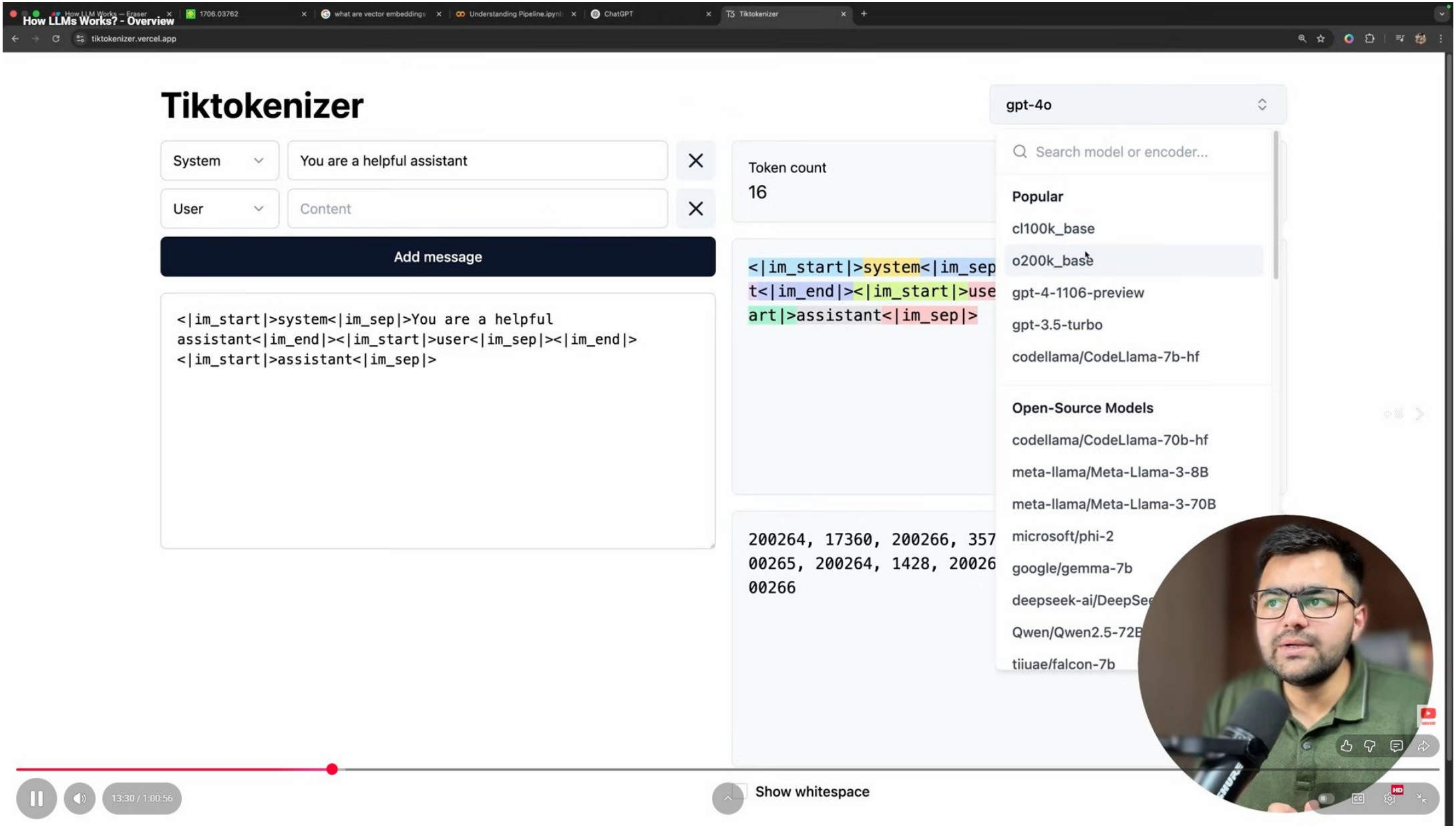
Dic
Hey - 10
There - 36

- 10

Step 1: Encoding → Hey There

Tokenize → 10 36

13:14 / 1:00:56


How LLMs Works? - Overview
tiktokenizer.vercel.app
Tiktokenizer
gpt-4o
System
You are a helpful assistant
User
Content
Add message
<|im_start|>system<|im_sep|>You are a helpful assistant<|im_end|><|im_start|>user<|im_sep|><|im_end|><|im_start|>assistant<|im_sep|>
Token count
16
<|im_start|>system<|im_sep
t<|im_end|><|im_start|>use
art|>assistant<|im_sep|>
200264, 17360, 200266, 357
00265, 200264, 1428, 20026
00266
Search model or encoder...
Popular
cl100k_base
o200k_base
gpt-4-1106-preview
gpt-3.5-turbo
codellama/CodeLlama-7b-hf
Open-Source Models
codellama/CodeLlama-70b-hf
meta-llama/Meta-Llama-3-8B
meta-llama/Meta-Llama-3-70B
microsoft/phi-2
google/gemma-7b
tiiuae/falcon-7b
13:30 / 1:00:56
Show whitespace

How LLMs Works? - Overview

tiktokenizer.vercel.app

Tiktokenizer

gpt-4o

User | Hey There

Add message

```
<|im_start|>user<|im_sep|>Hey There<|im_end|>
<|im_start|>assistant<|im_sep|>
```

Token count
9

```
<|im_start|>user<|im_sep|>Hey There<|im_end|><|im_star
t|>assistant<|im_sep|>
```

```
200264, 1428, 200266, 25216, 3274, 200265, 20026
781, 200266
```

14:04 / 1:00:56

Show whitespace

How LLMs Works? - Overview

colab.research.google.com/drive/1nY5N9G3iWTKZrhjOt5soXl2LD3FZ9GyY#scrollTo=1MOHKyPxPuHL

Understanding Pipeline.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text

RAM Disk

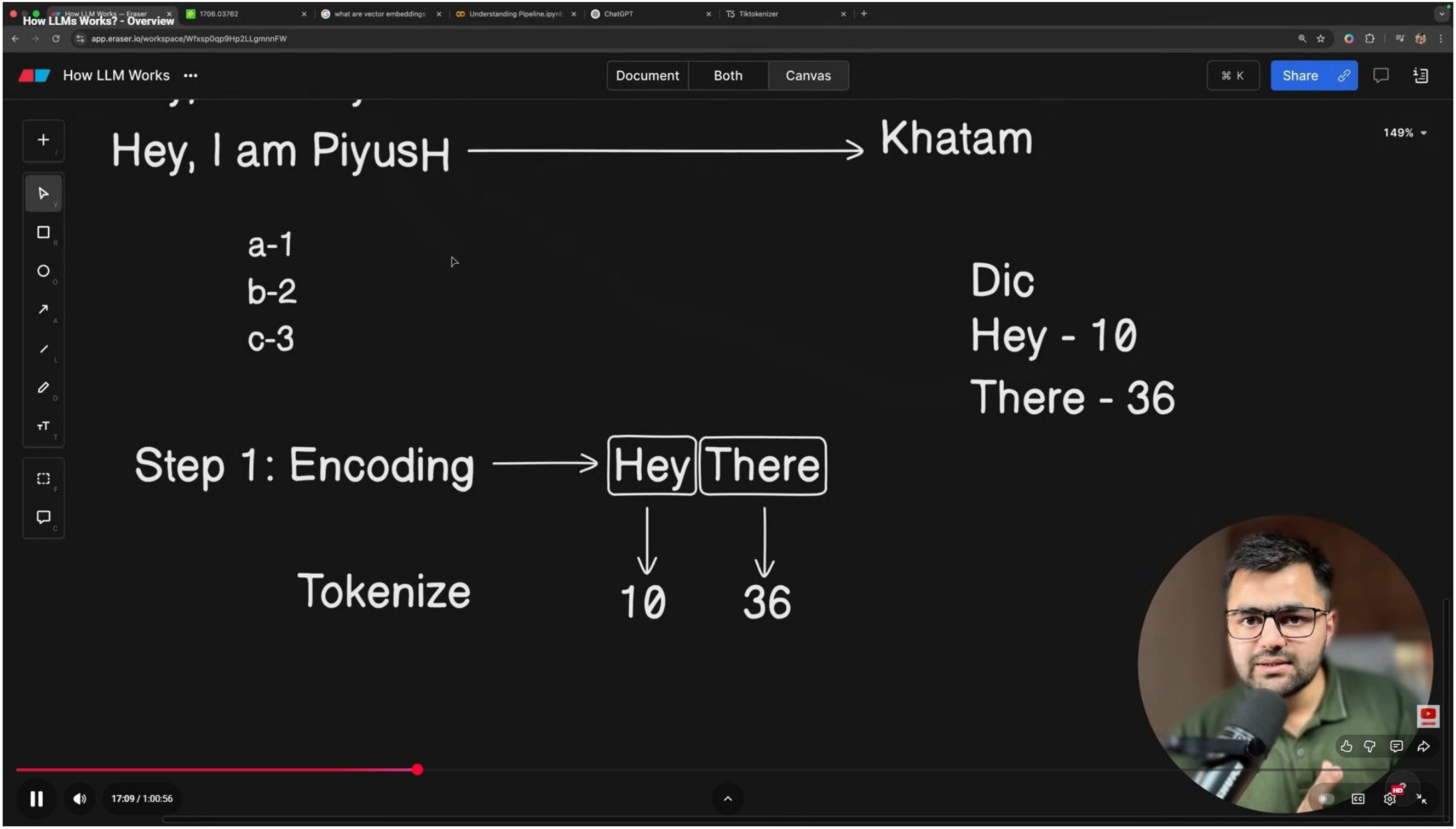
```
import os

os.environ["HF_TOKEN"] = "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"
```

```
[4] from transformers import AutoTokenizer

    tokenizer = AutoTokenizer.from_pretrained("google/gemma-3-1b-it")
    tokenizer("Hey There!")
```

```
tokenizer_config.json: 100% 1.16M/1.16M [00:00<00:00, 6.19MB/s]
tokenizer.model: 100% 4.69M/4.69M [00:00<00:00, 14.7MB/s]
tokenizer.json: 100% 33.4M/33.4M [00:00<00:00, 80.9MB/s]
added_tokens.json: 100% 35.0/35.0 [00:00<00:00, 1.60kB/s]
special_tokens_map.json: 100% 662/662 [00:00<00:00, 16.9kB/s]
{'input_ids': [2, 17531, 2085, 236888], 'attention_mask': [1, 1, 1, 1]}
```

16:24 / 1:00:56

5s completed at 10:23


How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
Share
Hey, I am PiyusH
Khatam
149%
a-1
b-2
c-3
Dic
Hey - 10
There - 36
Step 1: Encoding
Hey
There
Tokenize
10
36
17:09 / 1:00:56

How LLMs Works? - Overview

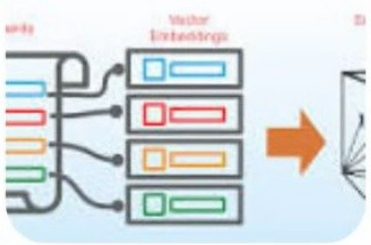
what are vector embeddings

All | Images | Videos | Short videos | Forums | Shopping | Web | More

Search Labs | AI Overview

हिं | En | Listen

Vector embeddings are numerical representations of data, like words, images, or audio, that capture semantic relationships and meaning, allowing machine learning models to process and compare them efficiently.



Here's a more detailed explanation:

What they are:

Vector embeddings are essentially "fingerprints" or numerical representations of data

What are Vector

Vector embeddings

and other data into

Elastic

18:57 / 1:00:56

Show more

How LLMs Works? - Overview

app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW

How LLM Works

Document | Both | Canvas

⌘ K | Share

There - 36

149%

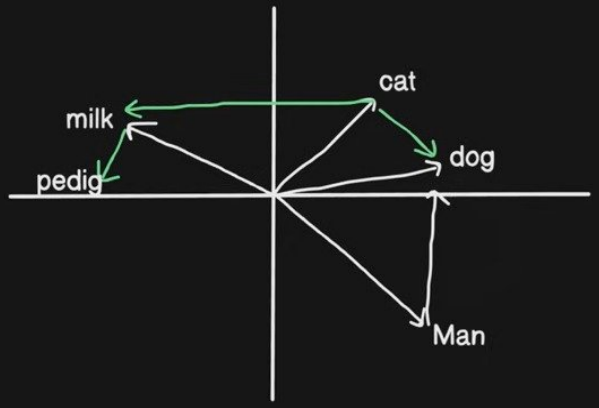
Step 1: Encoding → Hey There

Tokenize → 10 36

CAT - MILK
DOG - Pedig

+ 80

Vector Embeddings



CAT = 20
DOG = 80

21:51 / 1:00:56

How LLMs Works? - Overview

How LLM Works — Eraser | 1706.03762 | what are vector embeddings | Understanding Pipeline.ipynb | ChatGPT | Tiktokenizer

app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW

How LLM Works ••• Document Both Canvas ⌘ K Share

149%

There - 36

Step 1: Encoding → Hey There

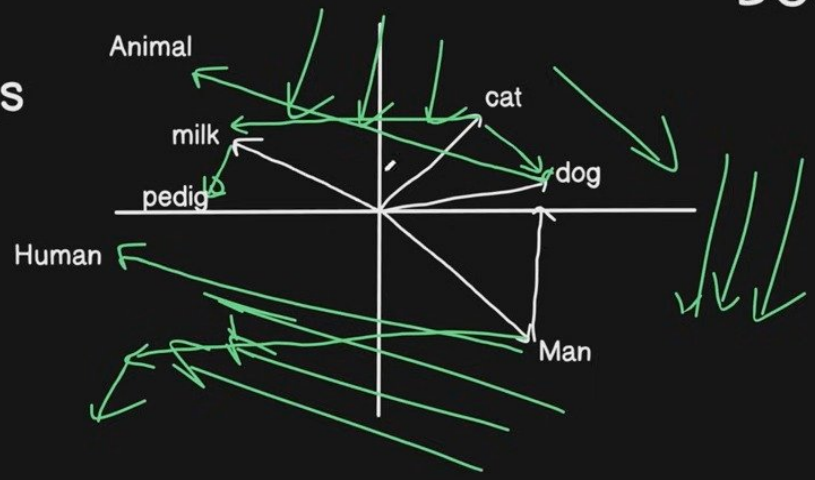
Tokenize → 10 36

CAT - MILK
DOG - Pedig

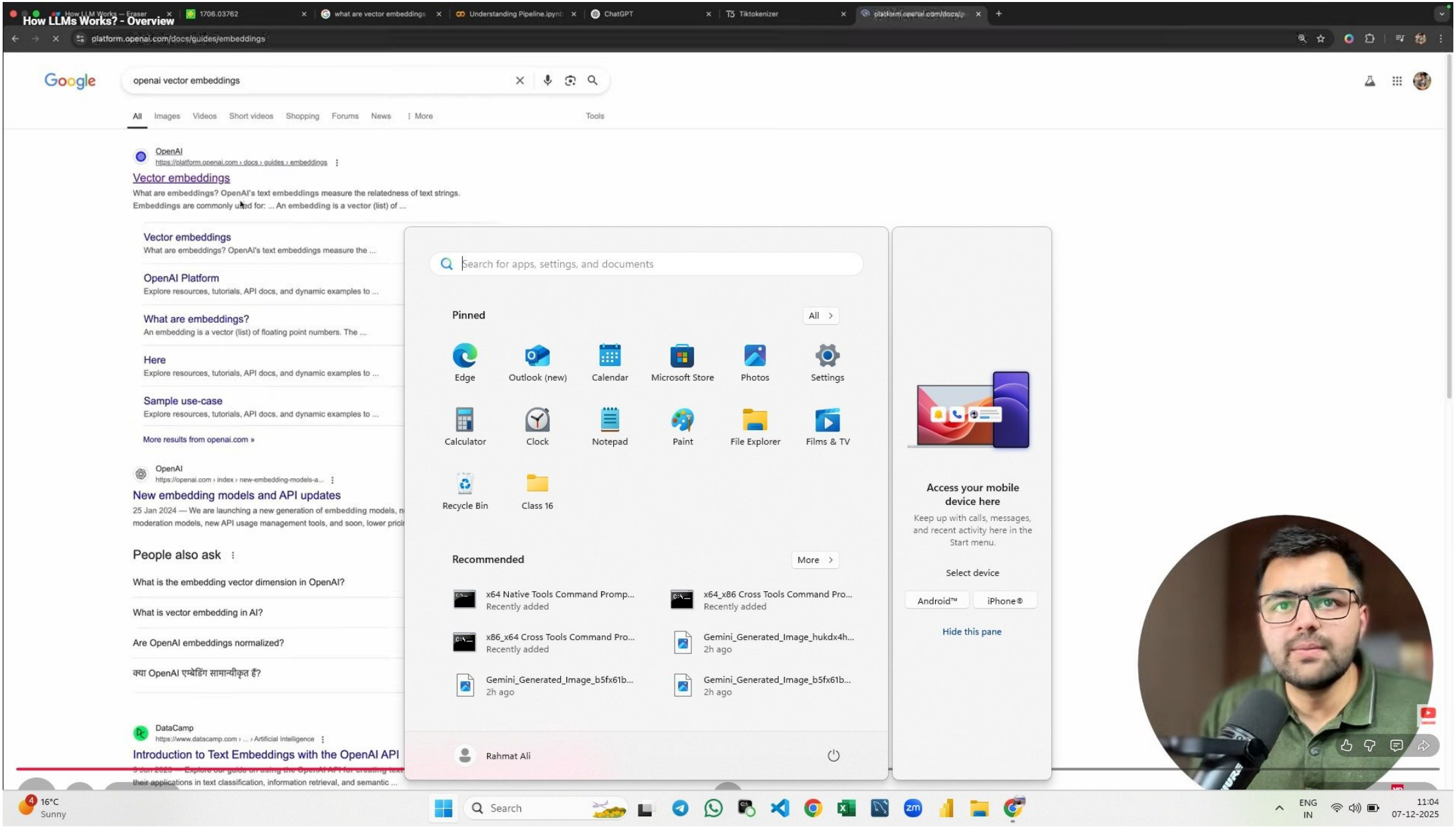
+ 30

Vector Embeddings

CAT = 20
DOG = 80



22:55 / 1:00:56


How LLMs Works? - Overview
platform.openai.com/docs/guides/embeddings
openai vector embeddings
All Images Videos Short videos Shopping Forums News More Tools
OpenAI
https://platform.openai.com › docs › guides › embeddings
Vector embeddings
What are embeddings? OpenAI's text embeddings measure the relatedness of text strings.
Embeddings are commonly used for: ... An embedding is a vector (list) of ...
Vector embeddings
What are embeddings? OpenAI's text embeddings measure the ...
OpenAI Platform
Explore resources, tutorials, API docs, and dynamic examples to ...
What are embeddings?
An embedding is a vector (list) of floating point numbers. The ...
Here
Explore resources, tutorials, API docs, and dynamic examples to ...
Sample use-case
Explore resources, tutorials, API docs, and dynamic examples to ...
More results from openai.com »
OpenAI
https://openai.com › index › new-embedding-models-a...
New embedding models and API updates
25 Jan 2024 — We are launching a new generation of embedding models, n
moderation models, new API usage management tools, and soon, lower prici
People also ask
What is the embedding vector dimension in OpenAI?
What is vector embedding in AI?
Are OpenAI embeddings normalized?
क्या OpenAI एम्बेडिंग सामान्यीकृत हैं?
DataCamp
https://www.datacamp.com › ... › Artificial Intelligence
Introduction to Text Embeddings with the OpenAI API
their applications in text classification, information retrieval, and semantic ...
Search for apps, settings, and documents
Pinned
All
Edge
Outlook (new)
Calendar
Microsoft Store
Photos
Settings
Calculator
Clock
Notepad
Paint
File Explorer
Films & TV
Recycle Bin
Class 16
Recommended
More
x64 Native Tools Command Promp...
Recently added
x64_x86 Cross Tools Command Pro...
Recently added
x86_x64 Cross Tools Command Pro...
Recently added
Gemini_Generated_Image_hukdx4h...
2h ago
Gemini_Generated_Image_b5fx61b...
2h ago
Gemini_Generated_Image_b5fx61b...
2h ago
Rahmat Ali
Access your mobile device here
Keep up with calls, messages, and recent activity here in the Start menu.
Select device
Android™
iPhone®
Hide this pane
16°C
Sunny
Search
ENG
IN
11:04
07-12-2025

- **Diversity measurement** (where similarity distributions are analyzed)
- **Classification** (where text strings are classified by their most similar label)

An embedding is a vector (list) of floating point numbers. The distance between two vectors measures their relatedness. Small distances suggest high relatedness and large distances suggest low relatedness.

Visit our pricing page to learn about embeddings pricing. Requests are billed based on the number of tokens in the input.

How to get embeddings

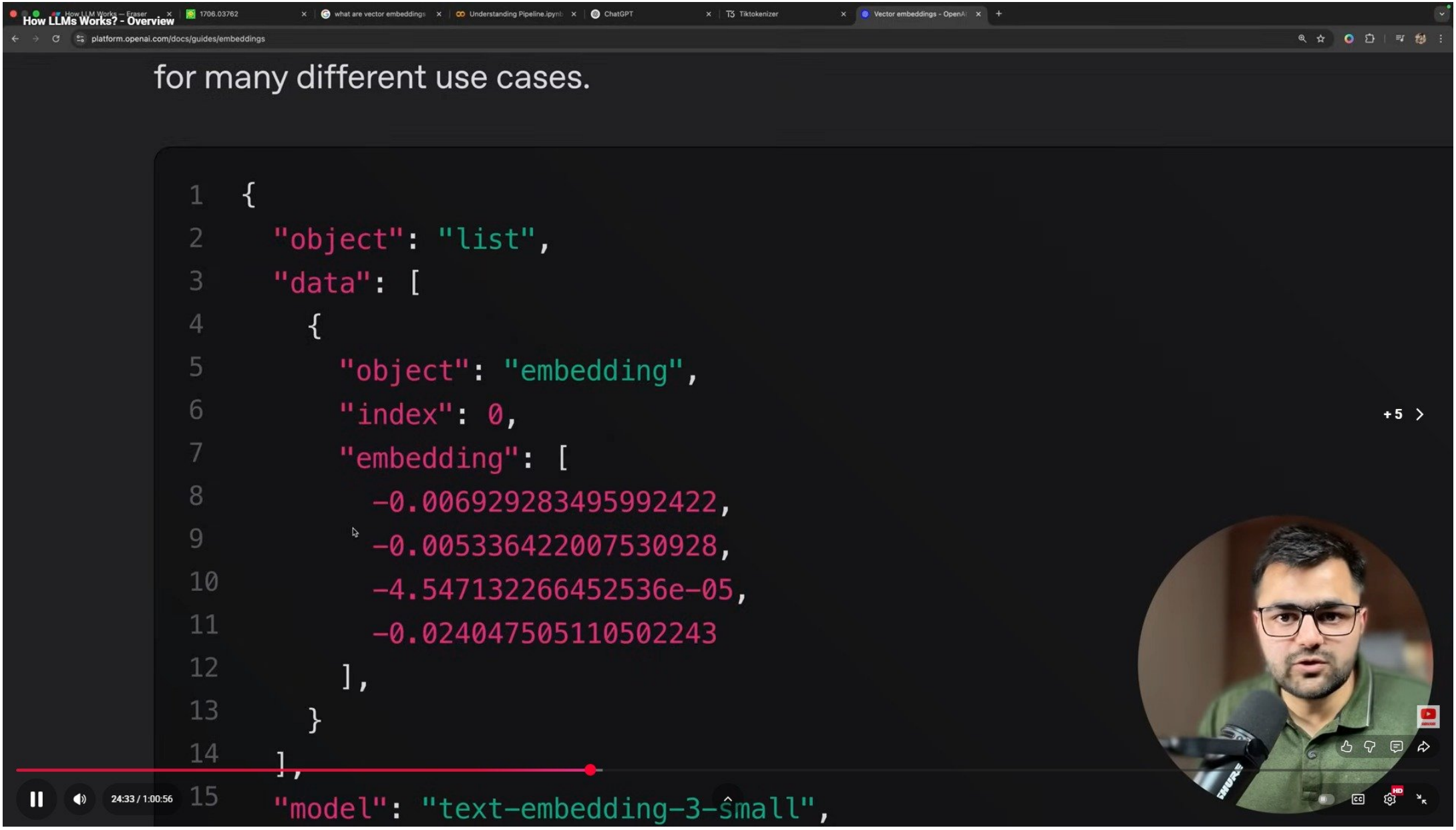
To get an embedding, send your text string to the embeddings API endpoint along with the embedding model name (e.g., `text-embedding-3-small`):

Example: Getting embeddings — python

```python
from openai import OpenAI
client = OpenAI()

response = client.embeddings.create(
    input="Your text string goes here",
    model="text-embedding-3-small"
)

print(response.data[0].embedding)
```

The response contains the embedding vector (list of floating point numbers) along with some additional metadata. You can extract the embedding vector, save it in a vector database, and use for many different use cases.

```
{
```

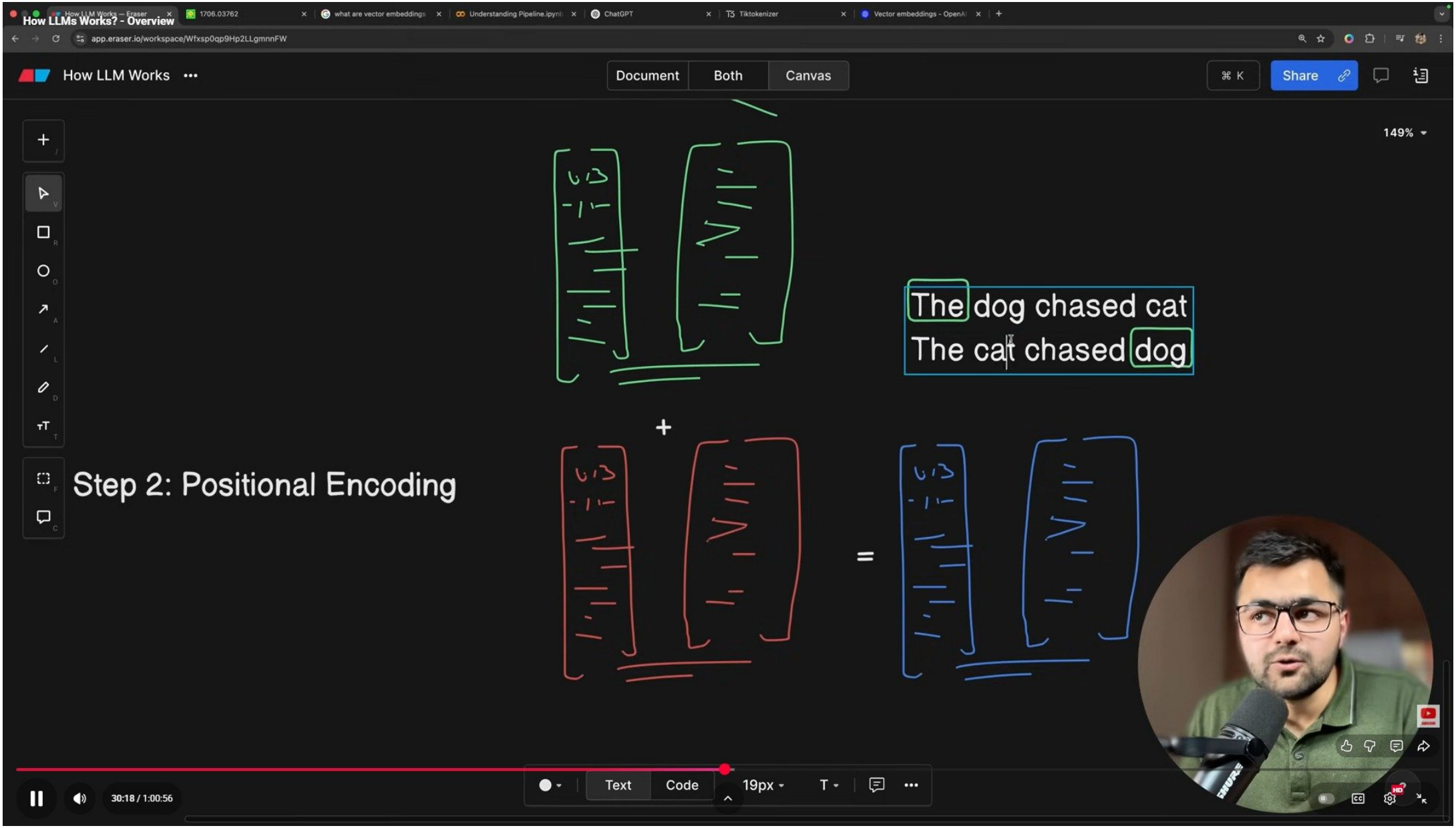

How LLMs Works? - Overview
platform.openai.com/docs/guides/embeddings
for many different use cases.
{
"object": "list",
"data": [
{
"object": "embedding",
"index": 0,
"embedding": [
-0.006929283495992422,
-0.005336422007530928,
-4.547132266452536e-05,
-0.024047505110502243
],
}
],
"model": "text-embedding-3-small",
+5
24:33 / 1:00:56

How LLMs Works? - Overview

```
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.11/dist-packages (from httpcore=
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from py
Requirement already satisfied: pydantic-core==2.27.2 in /usr/local/lib/python3.11/dist-packages (from pyd
```

```python
from openai import OpenAI
import os

client = OpenAI()

os.environ["OPENAI_API_KEY"] = "sk-proj-JJSk5dBf12JGy_4Giniq3UvbV7Xmbb7ezXRKhE_BDjqWeUY-2EHVGtbMzOOCeluMl

response = client.embeddings.create(
    input="Cat loves milk",
    model="text-embedding-3-small"
)

print(response.data[0].embedding)
```

26:20 / 1:00:56


How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
⌘ K
Share
149%
The dog chased cat
The cat chased dog
Step 2: Positional Encoding
Text
Code
19px
30:18 / 1:00:56

		Operations	
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

3.5 Positional Encoding

Since our model contains no recurrence and no convolution, in order for the model to make use of the order of the sequence, we must inject some information about the relative or absolute position of the tokens in the sequence. To this end, we add "positional encodings" to the input embeddings at the bottoms of the encoder and decoder stacks. The positional encodings have the same dimension d_{model} as the embeddings, so that the two can be summed. There are many choices of positional encodings, learned and fixed [9].

In this work, we use sine and cosine functions of different frequencies:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

where pos is the position and i is the dimension. That is, each dimension of the positional en corresponds to a sinusoid. The wavelengths form a geometric progression from 2π to $10000 \cdot 2$ chose this function because we hypothesized it would allow the model to easily learn to atten relative positions, since for any fixed offset k, PE_{pos+k} can be represented as a linear function PE_{pos}.

How LLMs Works? - Overview

app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW

How LLM Works

Document | Both | Canvas

⌘ K | Share

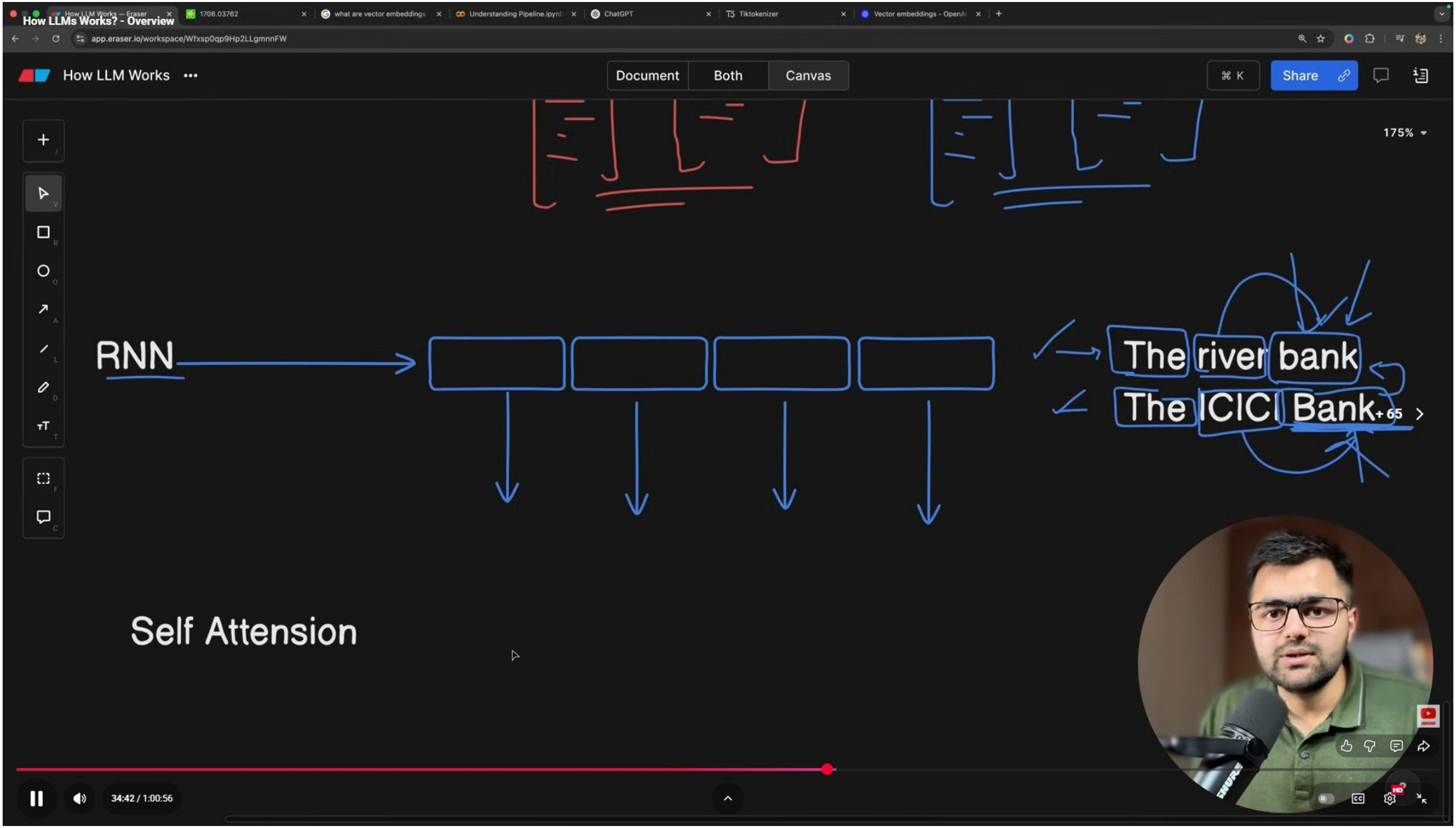
Step 2: Positional Encoding

149%

Self Attension

Text | Code | 19px | T

32:25 / 1:00:56


How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
Share
175%
RNN
The river bank
The ICICI Bank
+65
Self Attension
34:42 / 1:00:56

How LLMs Works? - Overview

app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW

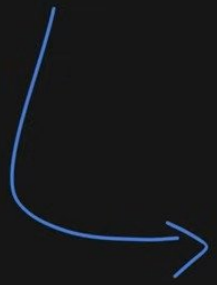
How LLM Works

Document | Both | Canvas

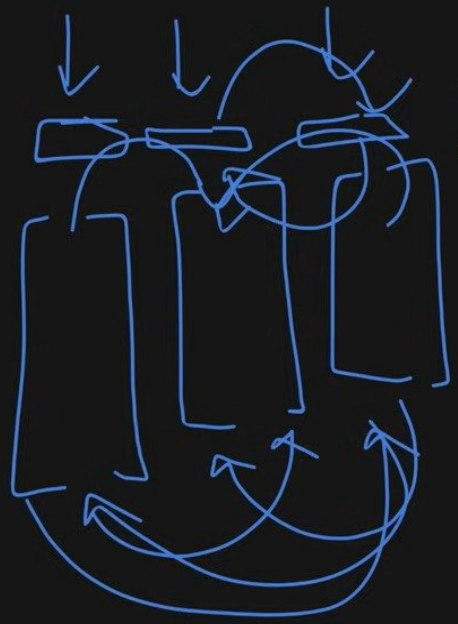
⌘ K | Share

170%

Self Attension

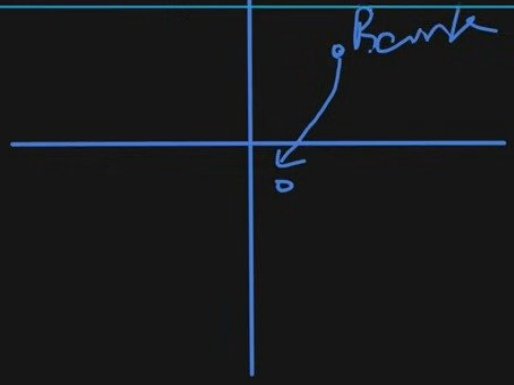


The river bank



Contex of the word is maintained

Relates words / token to each other



Text | Code | 15px | T

37:12 / 1:00:56

How LLMs Works? - Overview

app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW

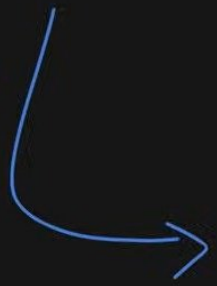
How LLM Works

Document | Both | Canvas

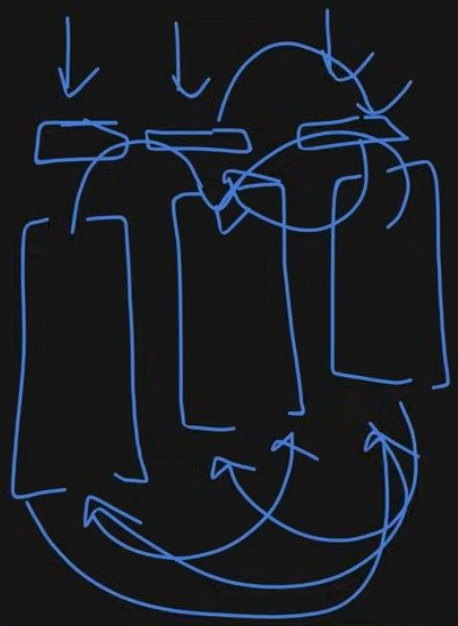
⌘ K | Share

170%

Self Attension



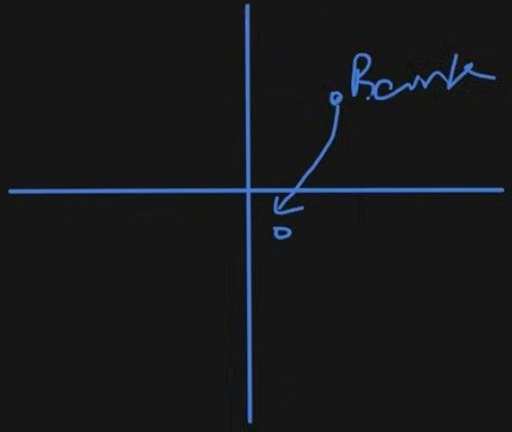
The river bank



Single Head Self Attension

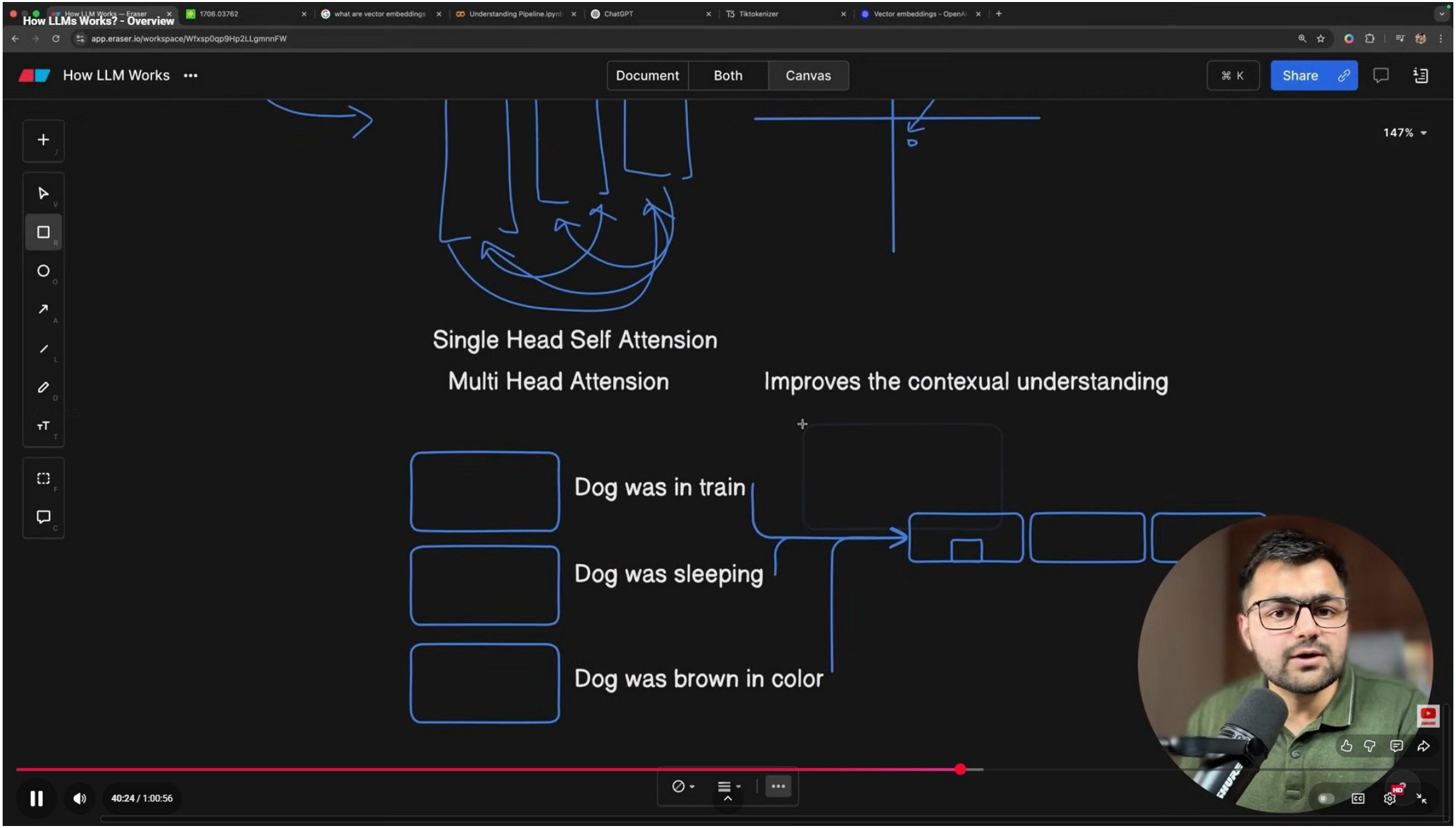
Contex of the word is maintained

Relates words / token to each other



37:36 / 1:00:56

Text | Code | 15px | T


How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
Share
147%
Single Head Self Attension
Multi Head Attension
Improves the contexual understanding
Dog was in train
Dog was sleeping
Dog was brown in color
40:24 / 1:00:56

How LLMs Works? - Overview

app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW

How LLM Works

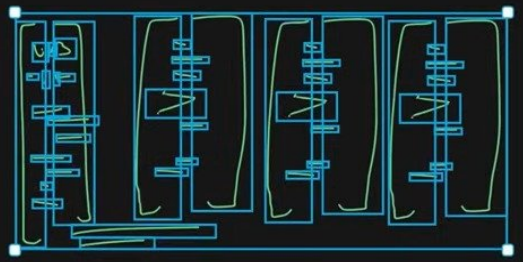
Document | Both | Canvas

⌘ K | Share

Dog was brown in color

94%

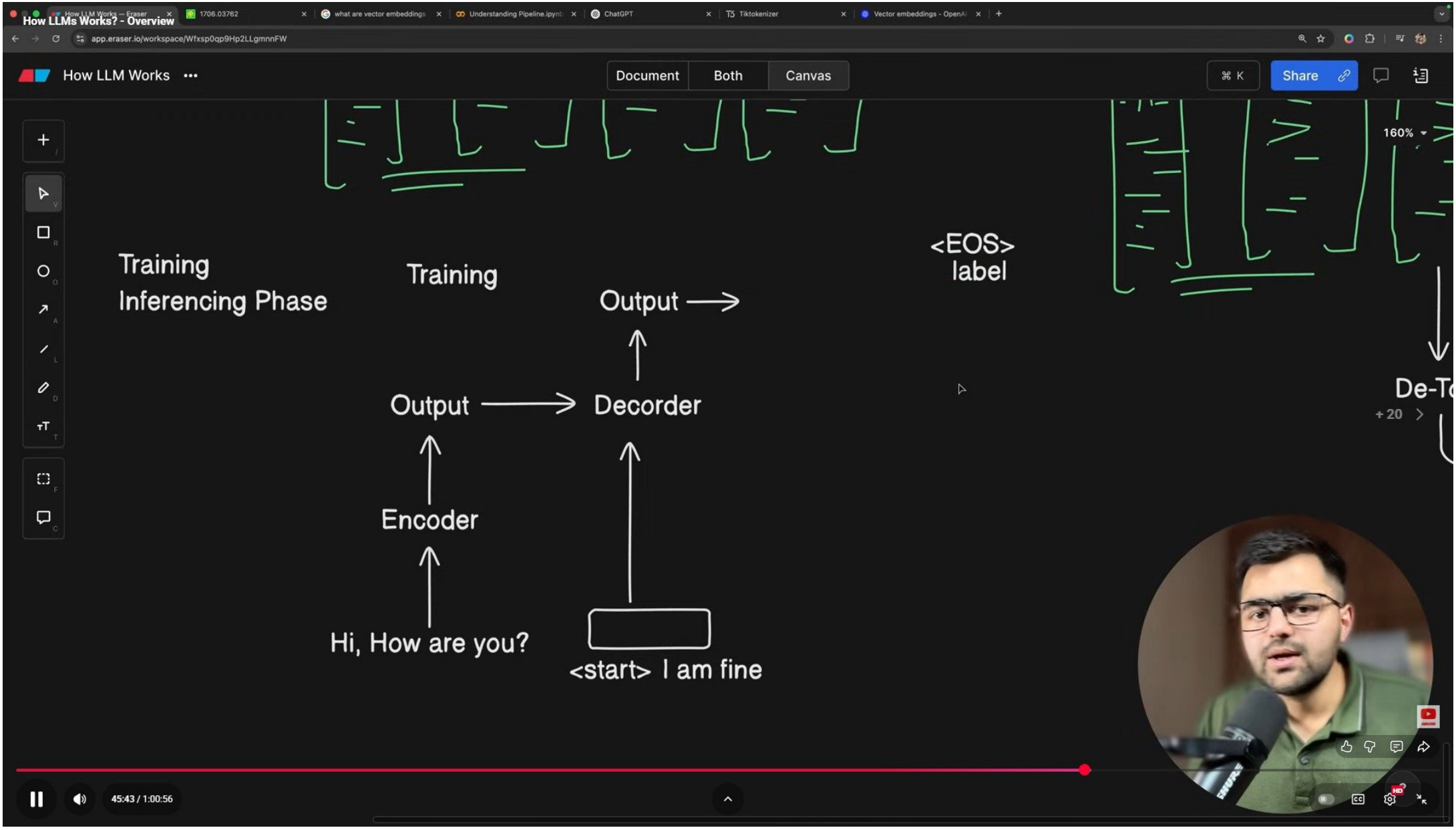
+ 60

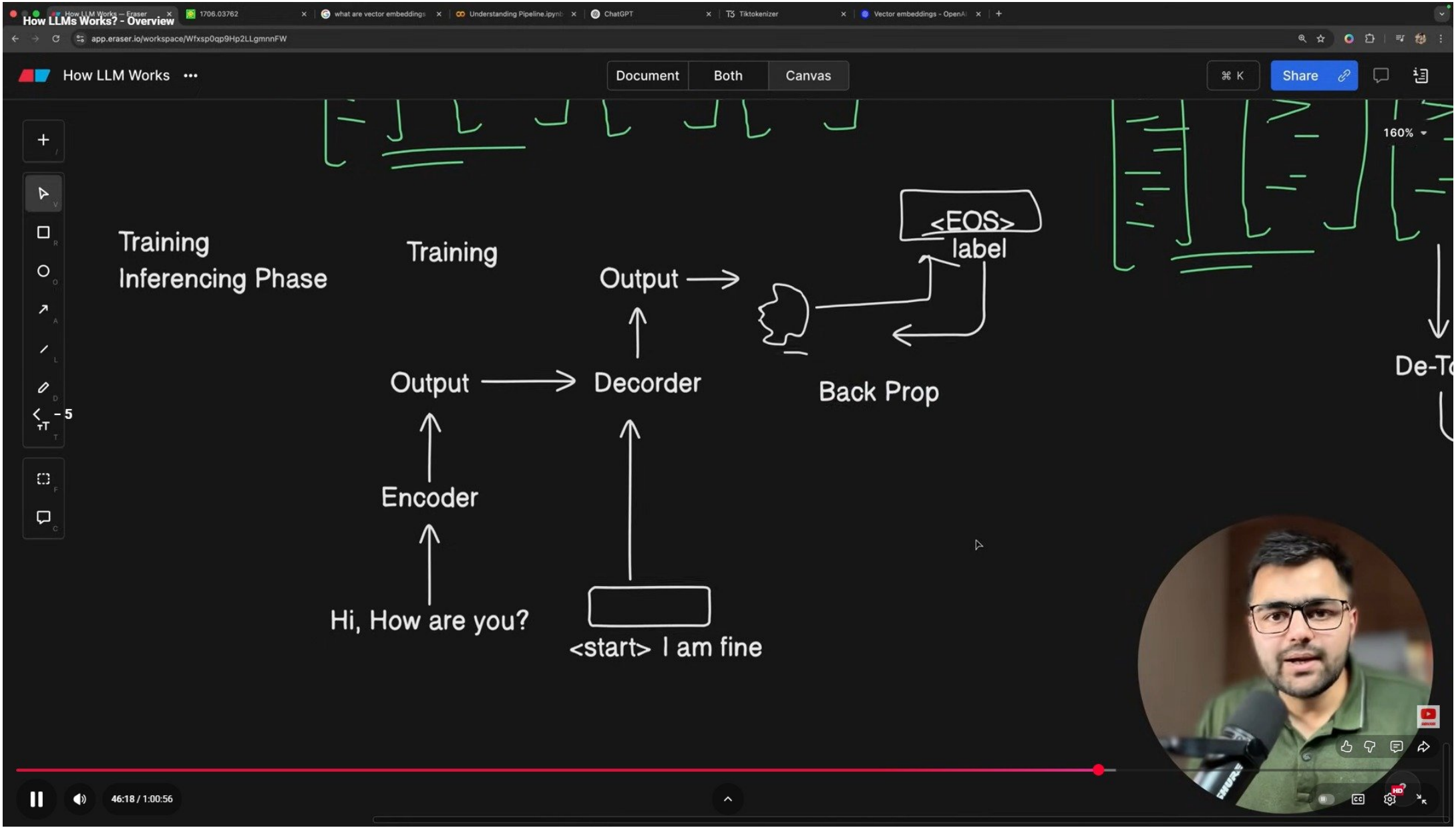


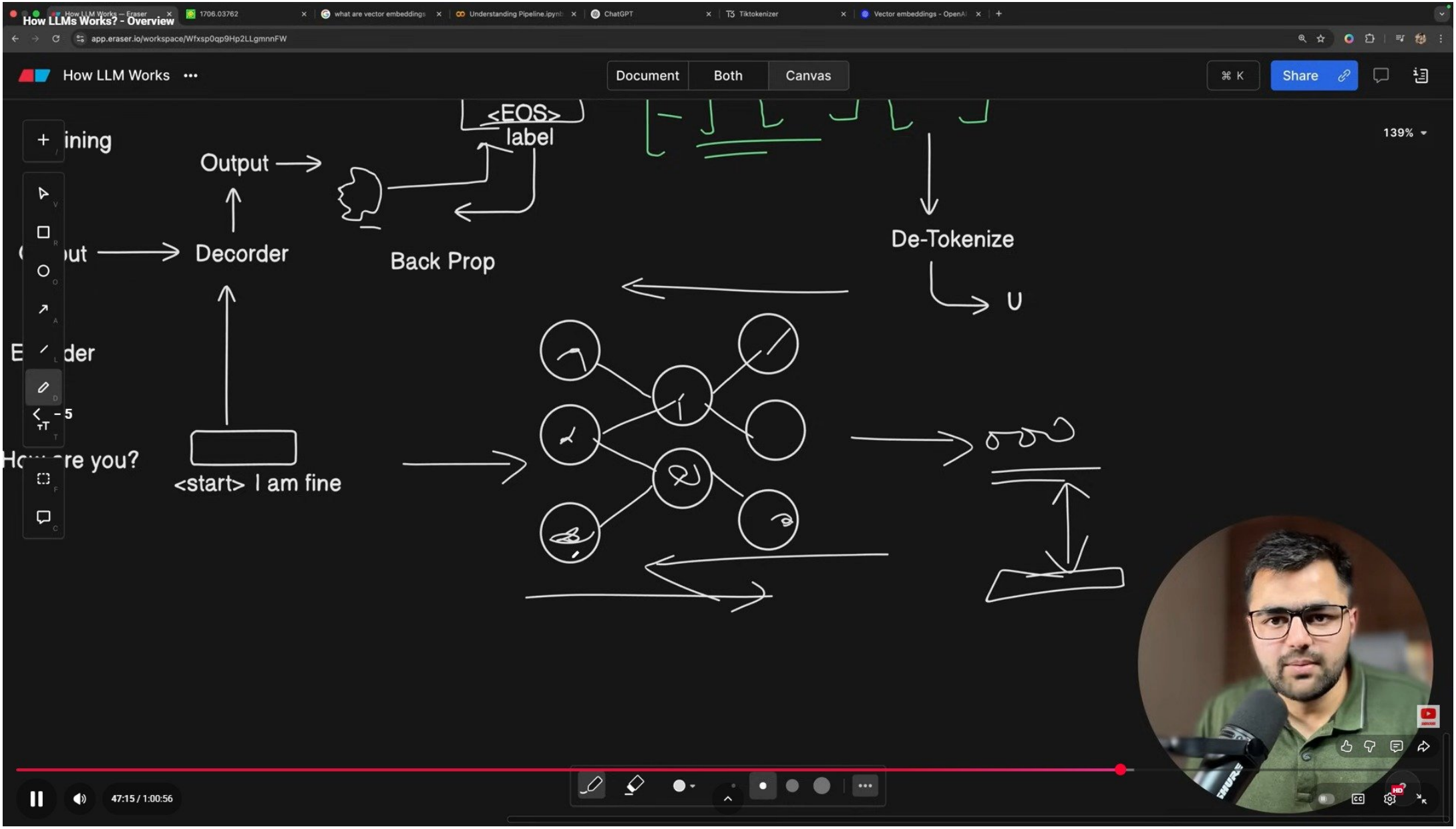
De-Tokenize

Create Figure ⇧ F

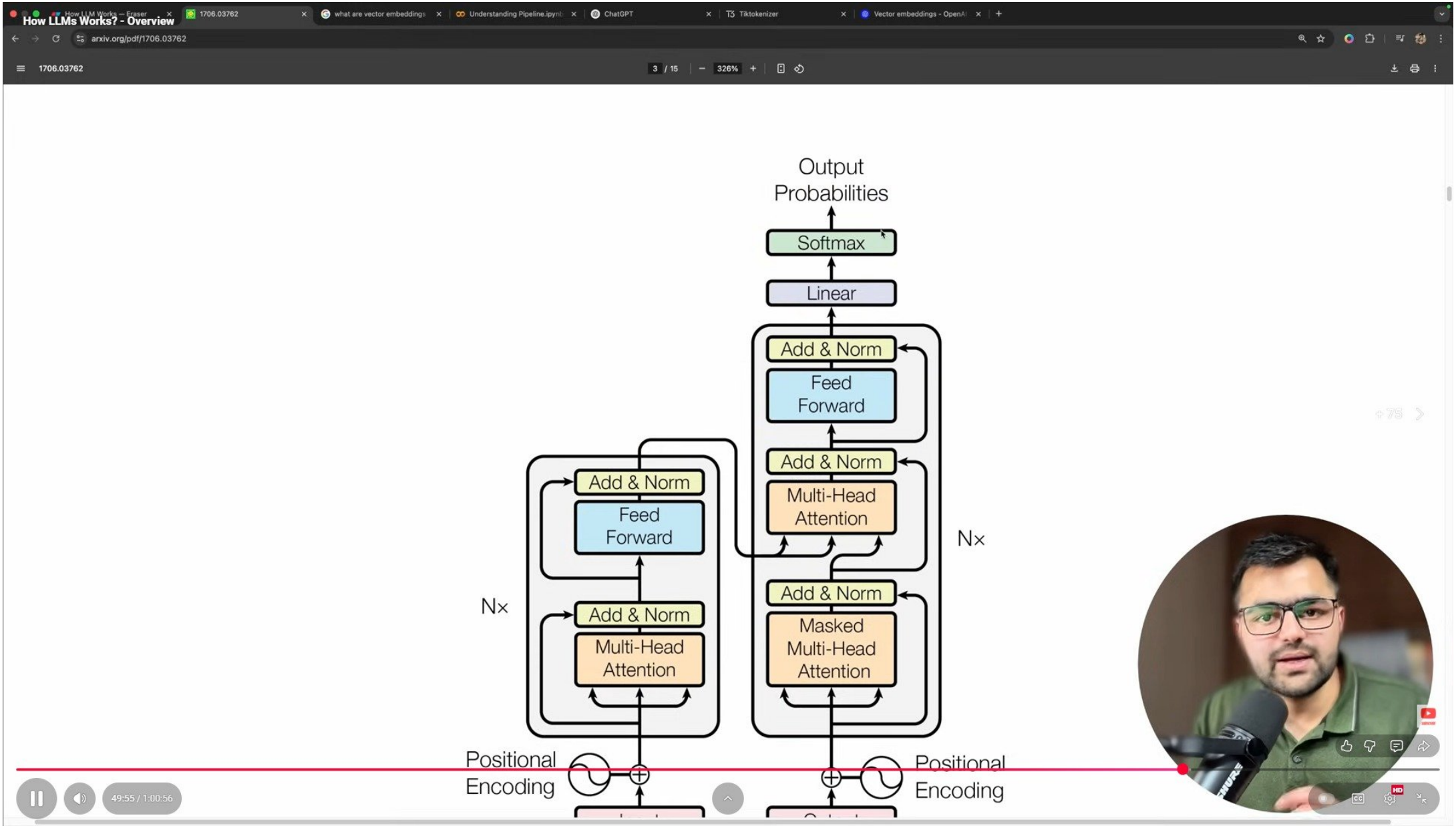
42:31 / 1:00:56

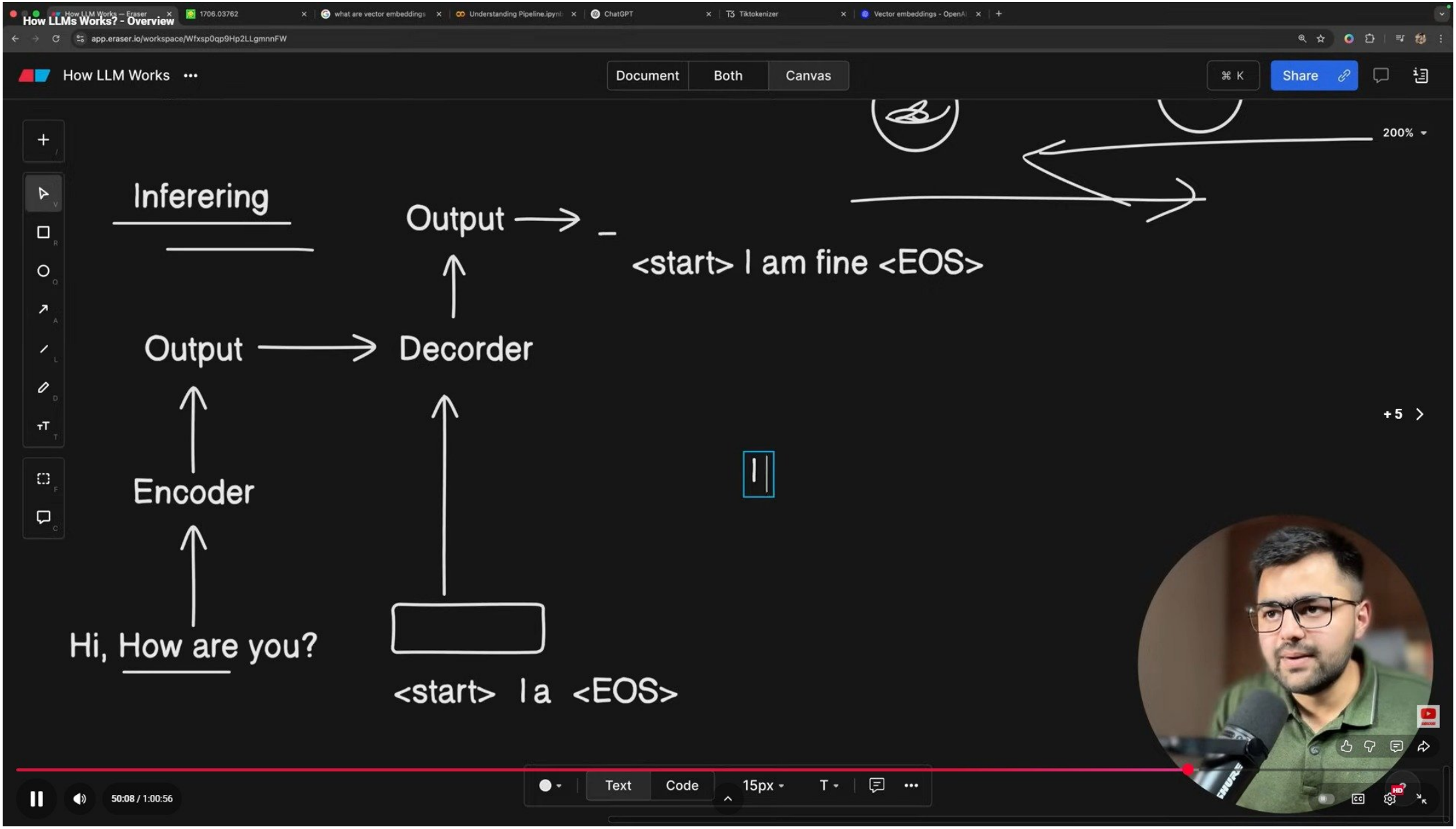

How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
⌘ K
Share
160%
Training
Inferencing Phase
Training
Output
Output
Decorder
Encoder
Hi, How are you?
<start> I am fine
<EOS>
label
De-T
+ 20
45:43 / 1:00:56

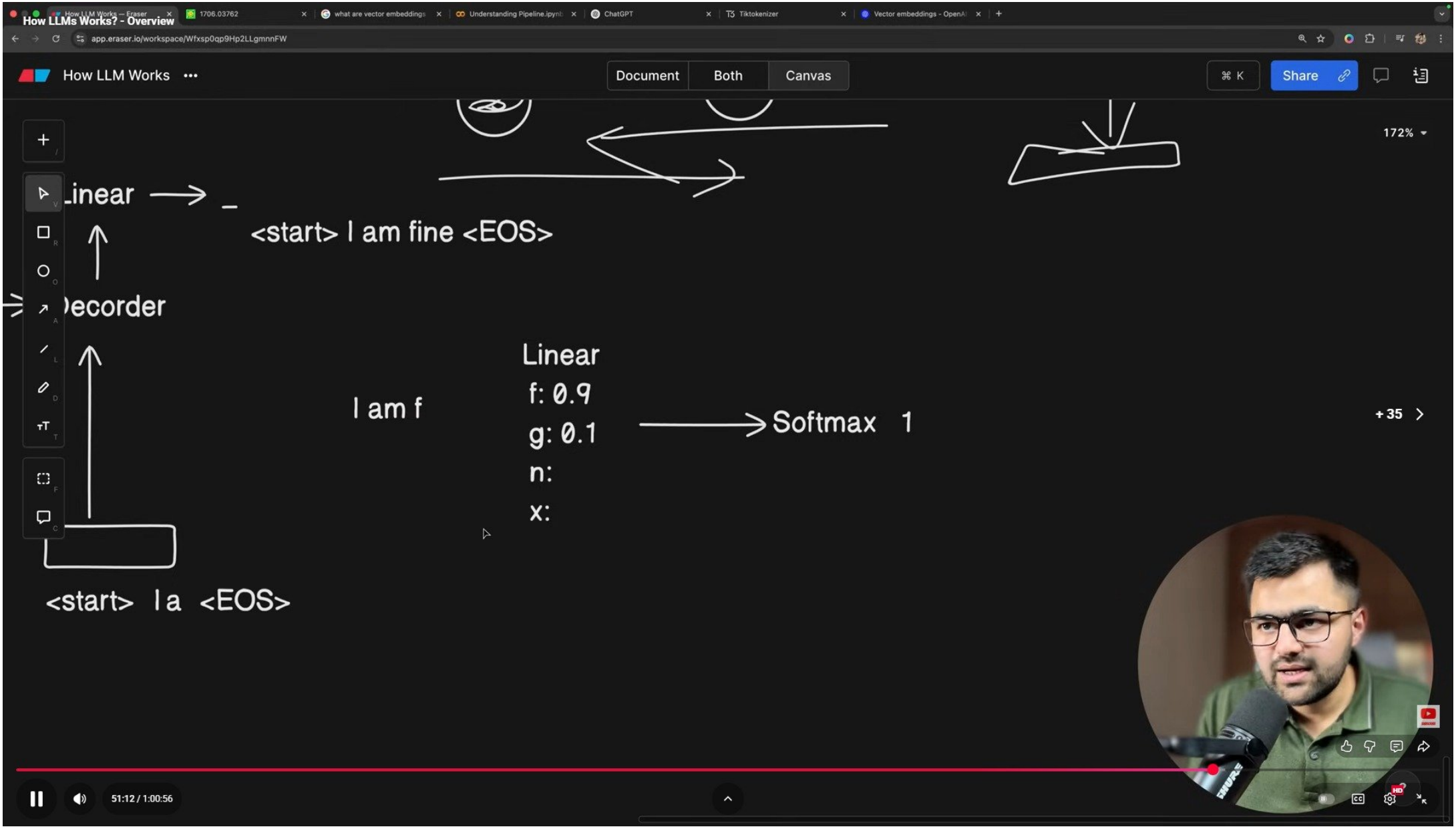

How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
Share
160%
Training
Inferencing Phase
Training
Output
<EOS>
label
Output
Decorder
Back Prop
Encoder
Hi, How are you?
<start> I am fine
46:18 / 1:00:56


How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
Share
<EOS>
label
Output
Decorder
Back Prop
De-Tokenize
U
<start> I am fine
139%
47:15 / 1:00:56

How LLMs Works? - Overview




How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
⌘ K
Share
200%
Inferering
Output
<start> I am fine <EOS>
Output
Decorder
Encoder
Hi, How are you?
<start> Ia <EOS>
+5
Text
Code
15px
50:08 / 1:00:56


How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
⌘ K
Share
172%
Linear
<start> I am fine <EOS>
Decorder
I am f
Linear
f: 0.9
g: 0.1
n:
x:
Softmax 1
+35
<start> I a <EOS>
51:12 / 1:00:56

How LLMs Works? - Overview

colab.research.google.com/drive/1nY5N9G3iWTKZrhjOt5soXl2LD3FZ9GyY#scrollTo=9W_UO7THaQ8t

```
CausalLMOutputWithPast(loss=None, logits=tensor([[[-14.1875,  -1.1250,   4.6562,  ..., -14.6250, -14.6875, -14.6250],
         [-17.6250,  -0.6992,  -3.2500,  ..., -18.0000, -18.0000, -17.8750],
         [-12.1875,  -2.9844,  -4.9688,  ..., -13.4375, -13.4375, -13.3125],
         ...,
         [-14.8750,  -3.8750,  -2.4688,  ..., -16.6250, -16.6250, -16.5000],
         [-15.9375,  -3.1250,   1.3047,  ..., -18.0000, -18.0000, -17.8750],
         [-17.0000,  -2.3125,  -0.4629,  ..., -19.5000, -19.5000, -19.2500]]],
       dtype=torch.bfloat16, grad_fn=<UnsafeViewBackward0>), past_key_values=<transformers.cache_utils.HybridCache object
hidden_states=None, attentions=None)
```

13s

```
[18] gen_out = model.generate(input_ids=input_tokens["input_ids"])
     gen_out
```

```
tensor([[     2,   6974,    496,  23181,   3393,    573,   8009,   1156,   4945,
           107,   2717,   6719,    107,   2063,   1138, 236779,  34488, 236769,
        236781, 236764,    570,   1473,    107,    138,  12234, 174054,   1156,
          4945,   3075]])
```

+155

Start coding or generate with AI.

56:36 / 1:00:56

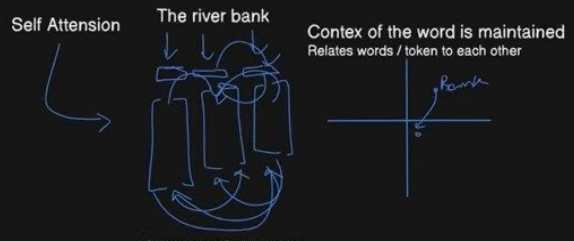
How LLMs Works? - Overview

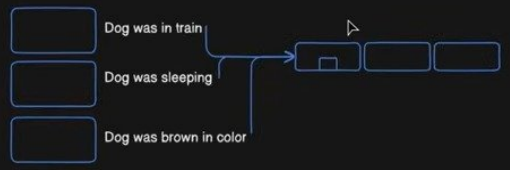
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW

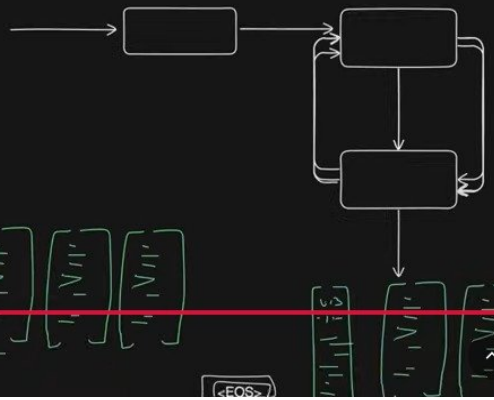
How LLM Works

Document | Both | Canvas

⌘ K | Share







56%

+ 40

58:01 / 1:00:56

How LLMs Works? - Overview

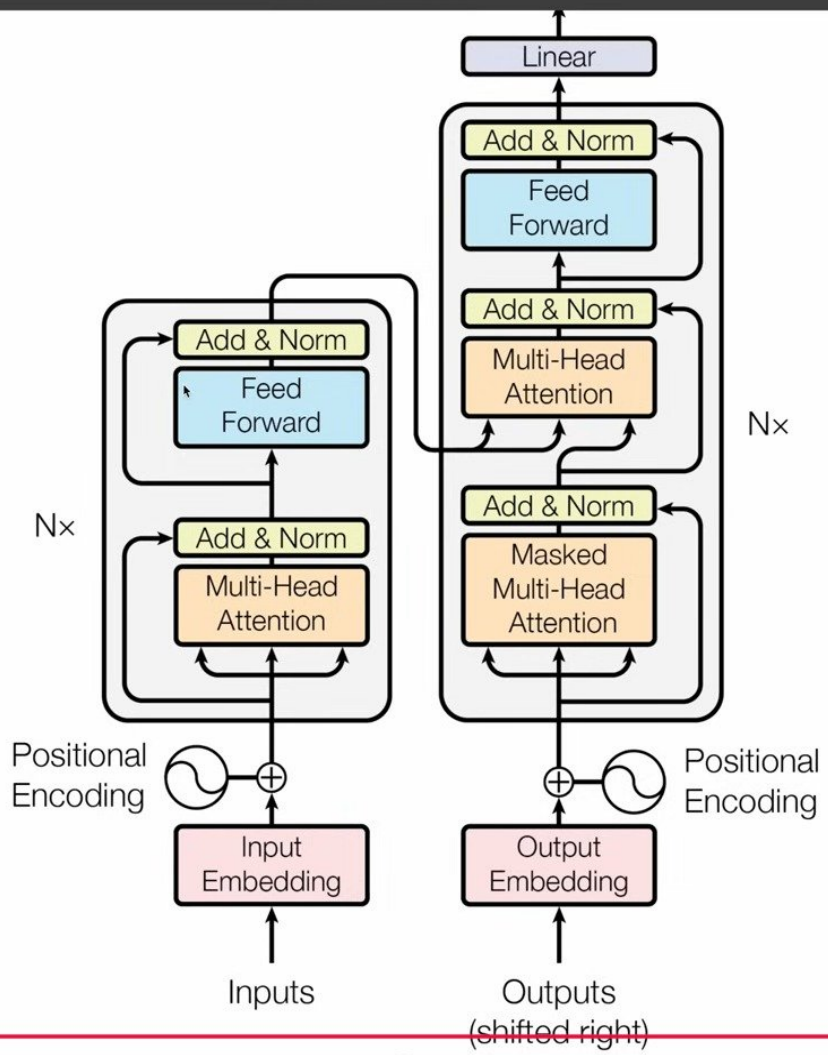
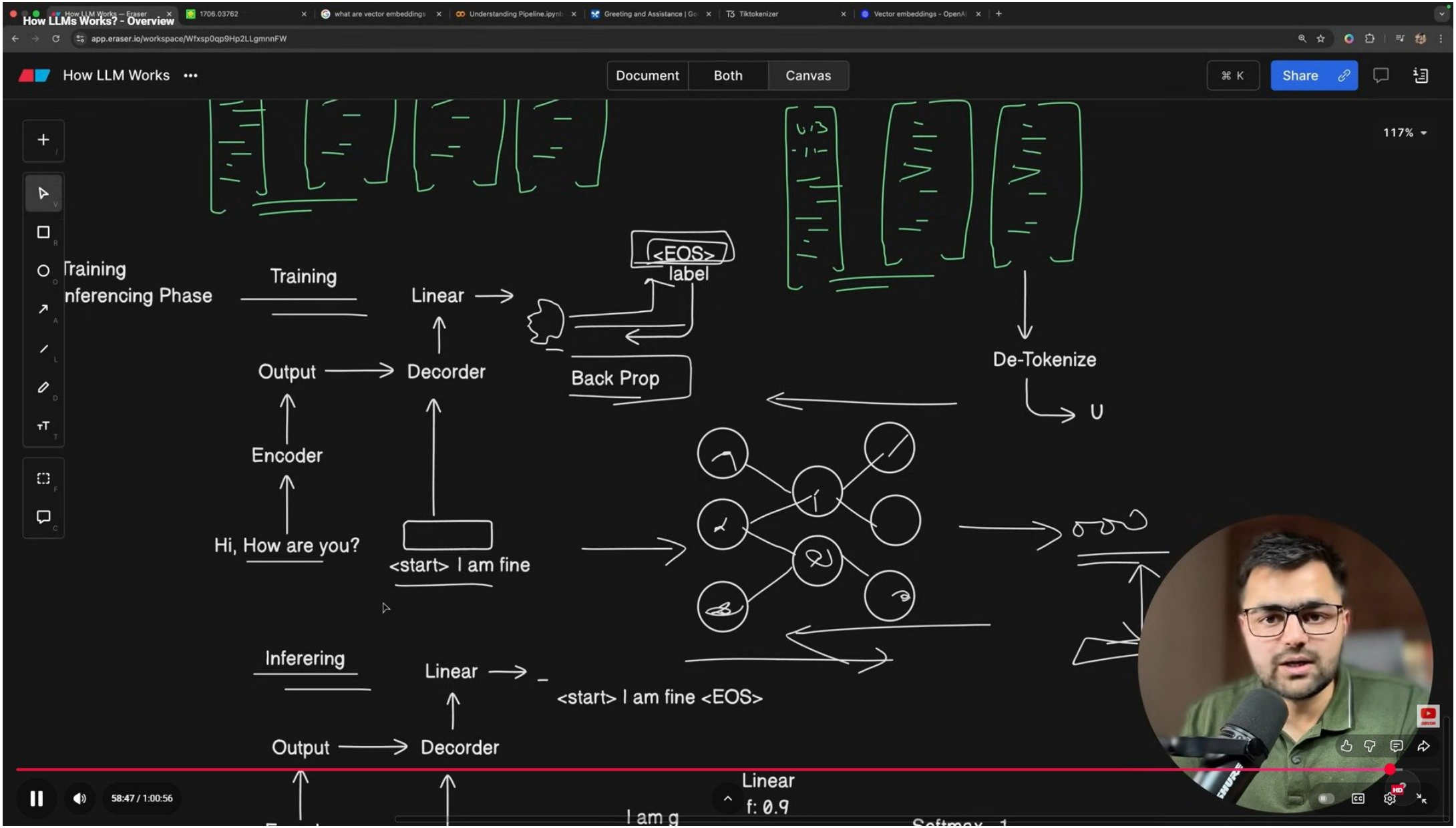


Figure 1: The Transformer - model architecture.


How LLMs Works? - Overview
app.eraser.io/workspace/Wfxsp0qp9Hp2LLgmnnFW
How LLM Works
Document
Both
Canvas
Share
117%
Training
nferencing Phase
Training
Linear
<EOS>
label
Output
Decorder
Back Prop
De-Tokenize
U
Encoder
Hi, How are you?
<start> I am fine
Inferering
Linear
<start> I am fine <EOS>
Output
Decorder
Linear
f: 0.9
I am g
58:47 / 1:00:56