# Refreshers Modern Text & Image AI Model Architectures

1] Modern day LLMs (text) ← curved ↙ — entire pipeline (fast) (slow)

2] Modern day vision models ( Vision Transformers ) ↑ Attention

3] Multimodality (refresher)

↑ class intro to CLSP }

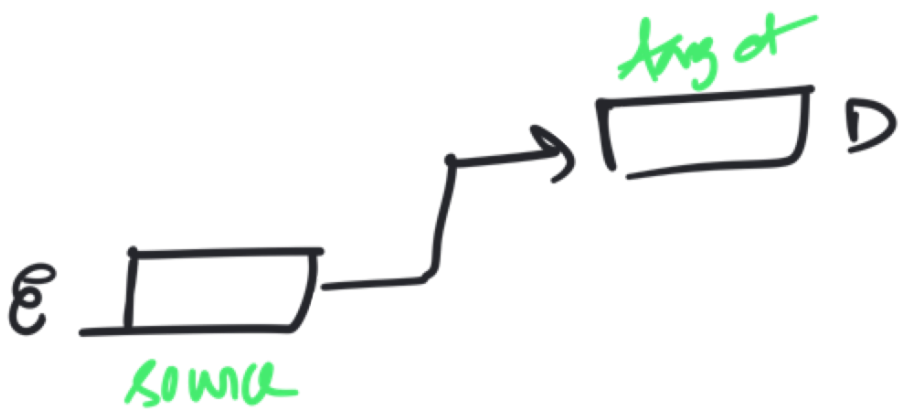
*Diffusion ( Multimodal )

1] Encoder only (BERT)

Chatting [X]

Sentiment Analysis [✓]

2] Encoder - Decoder (T5)




~~Translation~~ — a data dist

source

target

3] Decoder only GPT-2

No need of separate encoder

# Training Dataset

Use a special token

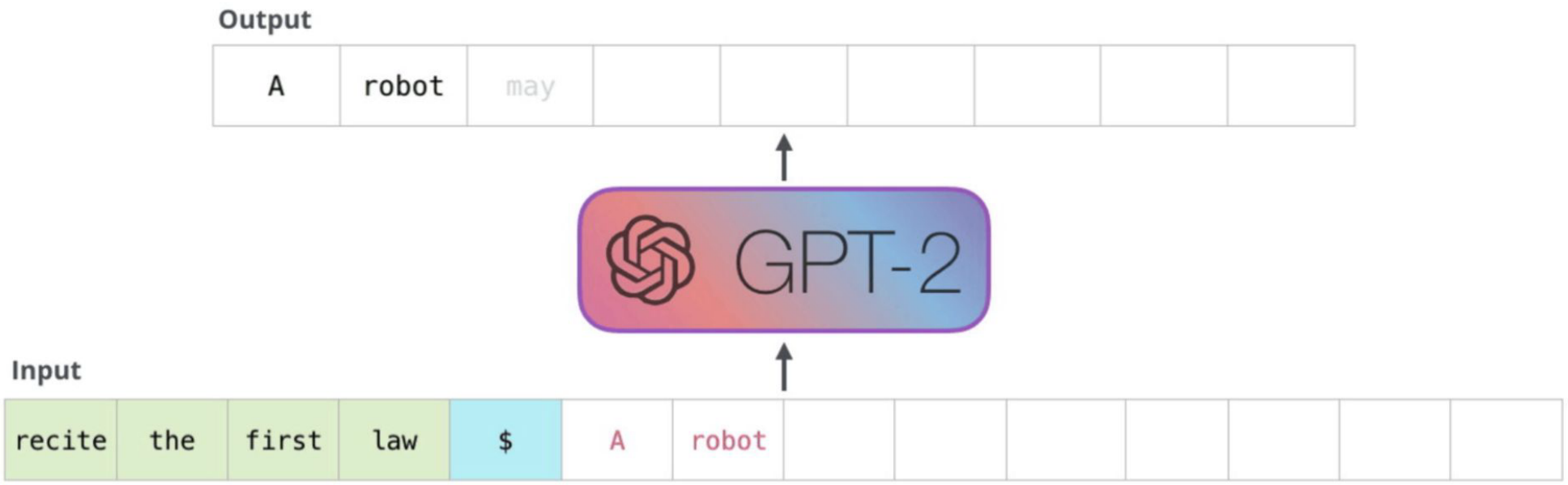
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| I | am | a | student | <to-fr> | je | suis | étudiant |
| let | them | eat | cake | <to-fr> | Qu'ils | mangent | de |
| good | morning | <to-fr> | Bonjour | | | | |

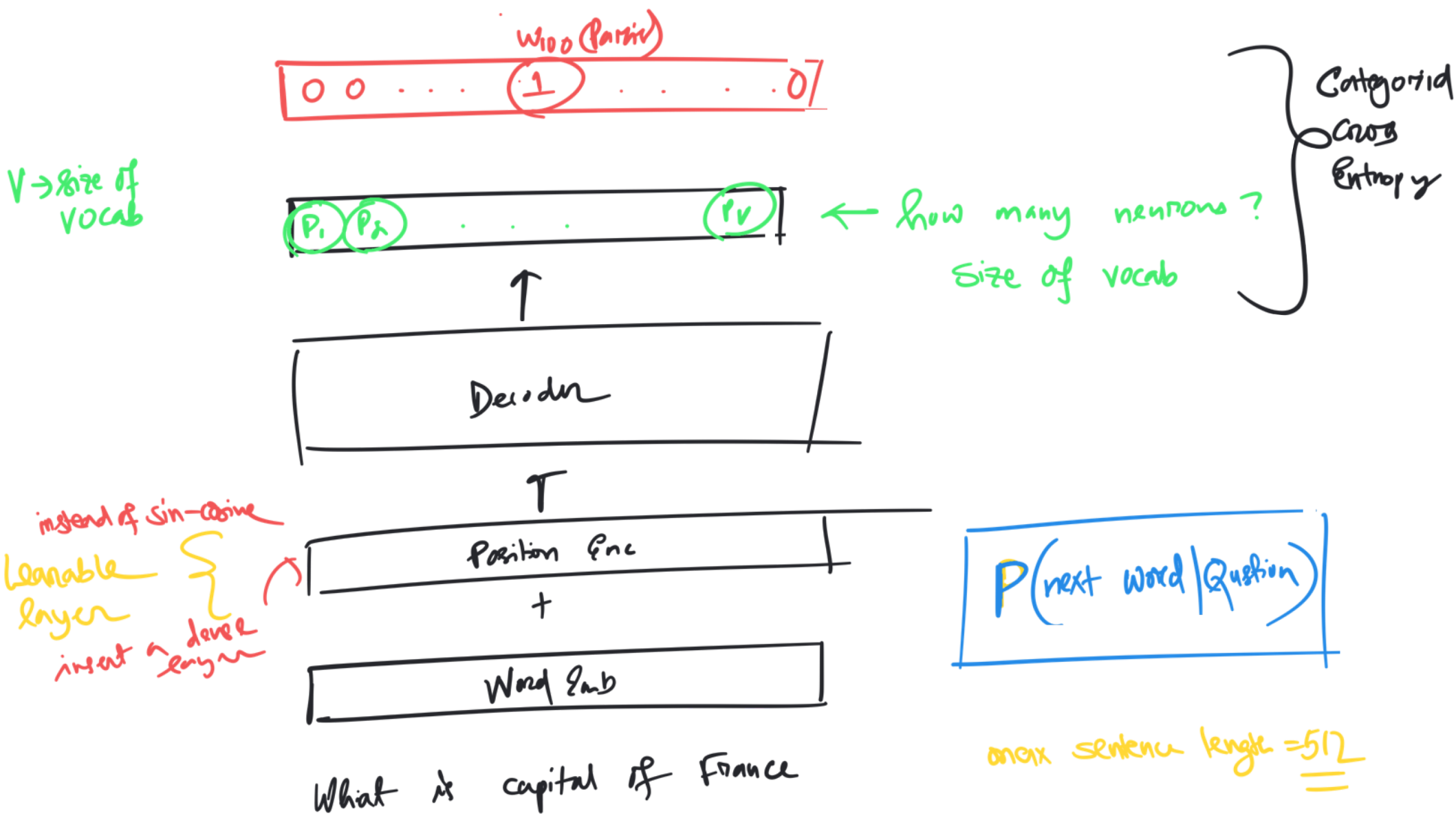
# Training Dataset

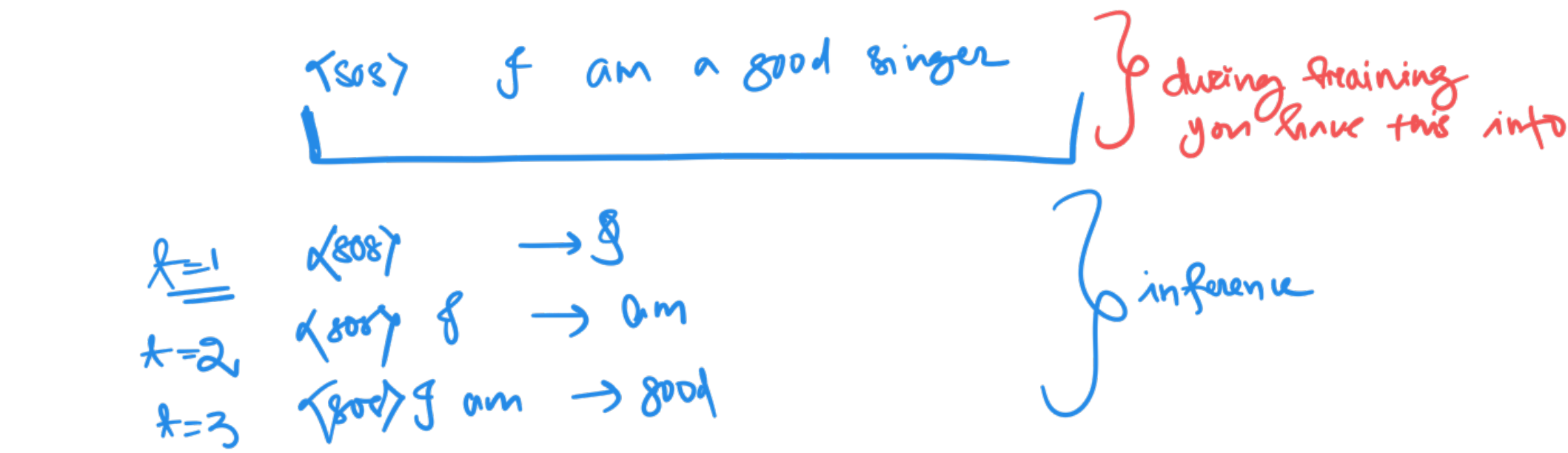
special token for summarization

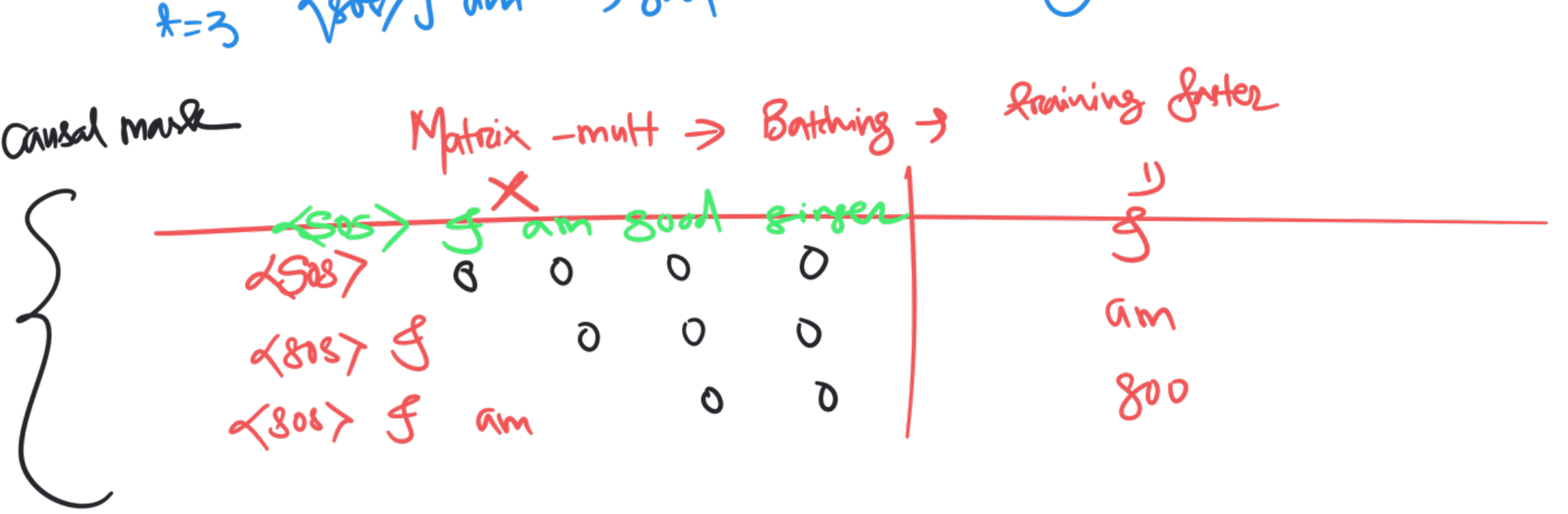
| | | | |
|---|---|---|---|
| Article #1 tokens | **<summarize>** | Article #1 Summary | |
| Article #2 tokens | **<summarize>** | Article #2 Summary | padding |
| Article #3 tokens | | **<summarize>** | Article #3 Summary |

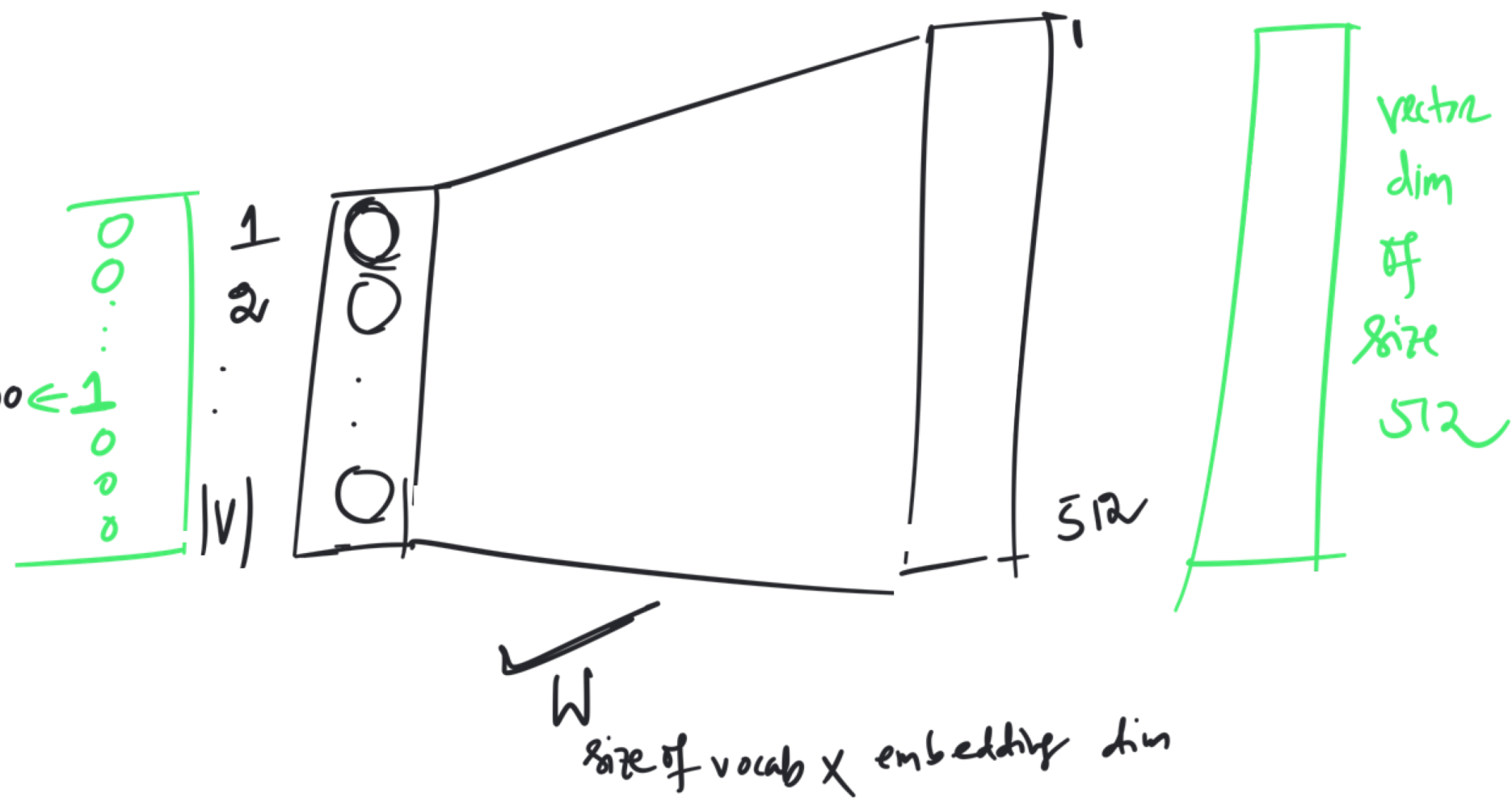
Next token prediction models

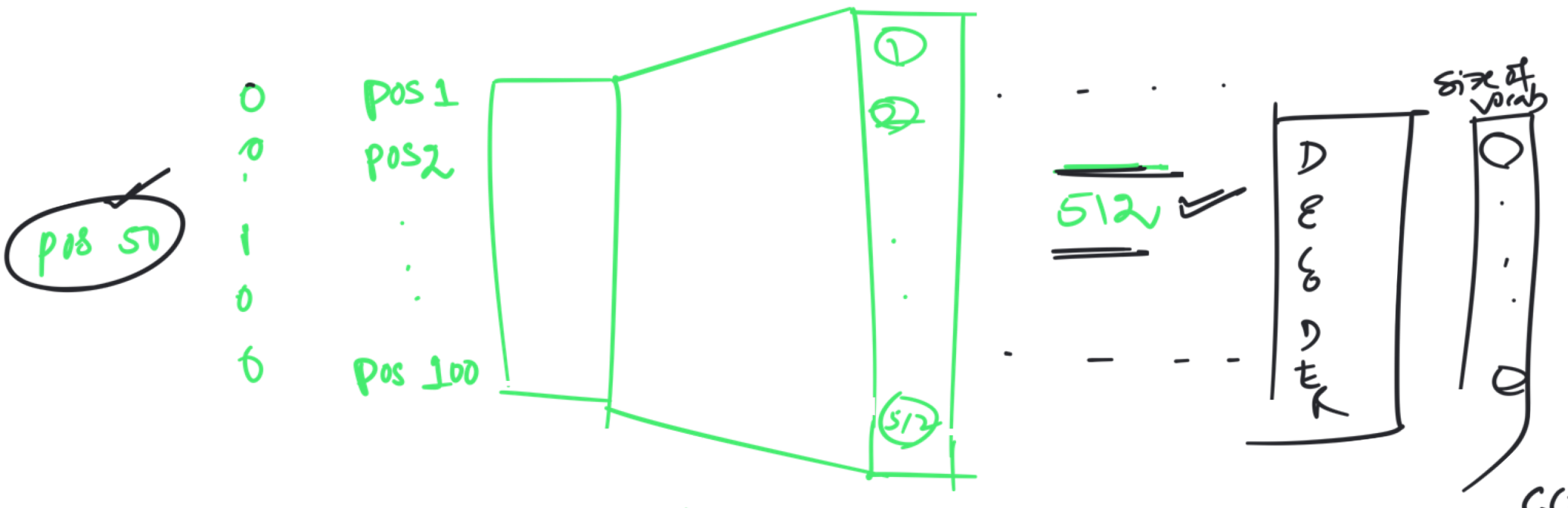



Word (Token)
0 0 . . . 1 . . . . 0
V → Size of vocab
P1 P2 . . . PV
← how many neurons ?
Size of vocab
Categorical Cross Entropy
Decoder
T
instead of sin-cosine
Learnable layer
insert a dense layer
Position Enc
+
Word Emb
What is capital of France
P(next word | Question)
max sentence length = 512


<SOS> I am a good singer
during training you have this info
k=1
<SOS> → I
k=2
<SOS> I → am
k=3
<SOS> I am → good
inference


causal mask
Matrix -mult → Batching → training faster
<SOS> I am good singer
<SOS> 0 0 0 0
<SOS> I 0 0 0
<SOS> I am 0 0
I
am
good


8101
100
1
0
0
0
0
|V|
1
2
Q
Q
Q
512
vector dim 的 size 512
W
size of vocab x embedding dim




W
no. of pos x embedding dim

↖ before training
random

↗ After training
learns to represent
each pos - 512 dim

Stand in

Softmax

# of words = vocab } Step 5

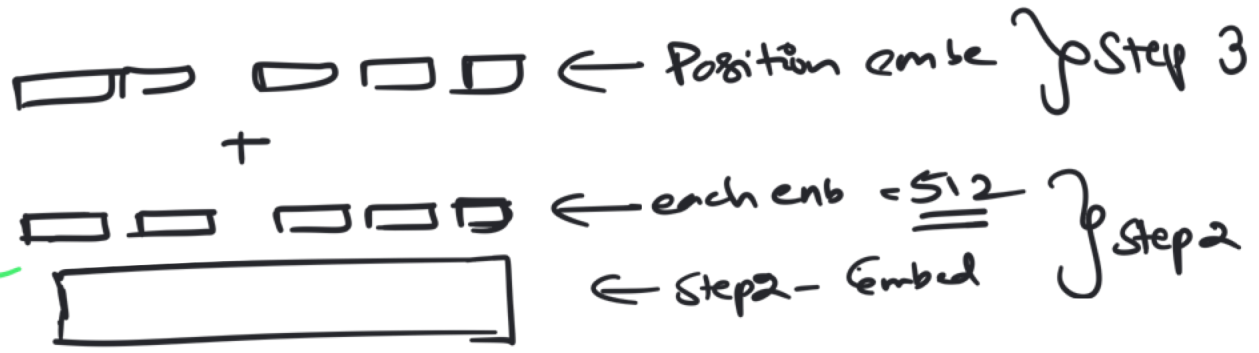

MHSA
Layer Norm
FF
Residual connection

pos
0
1
2
3
4

vector = 512

Step 4

Position emb
Token emb



← Position embe } Step 3

+

← each emb = 512 } Step 2

← Step2 - Embed

91 92 93 94 95 ← Step1 : Tokenize

What is Capital of France

pos = 0 1 2 3 4

sin-cos

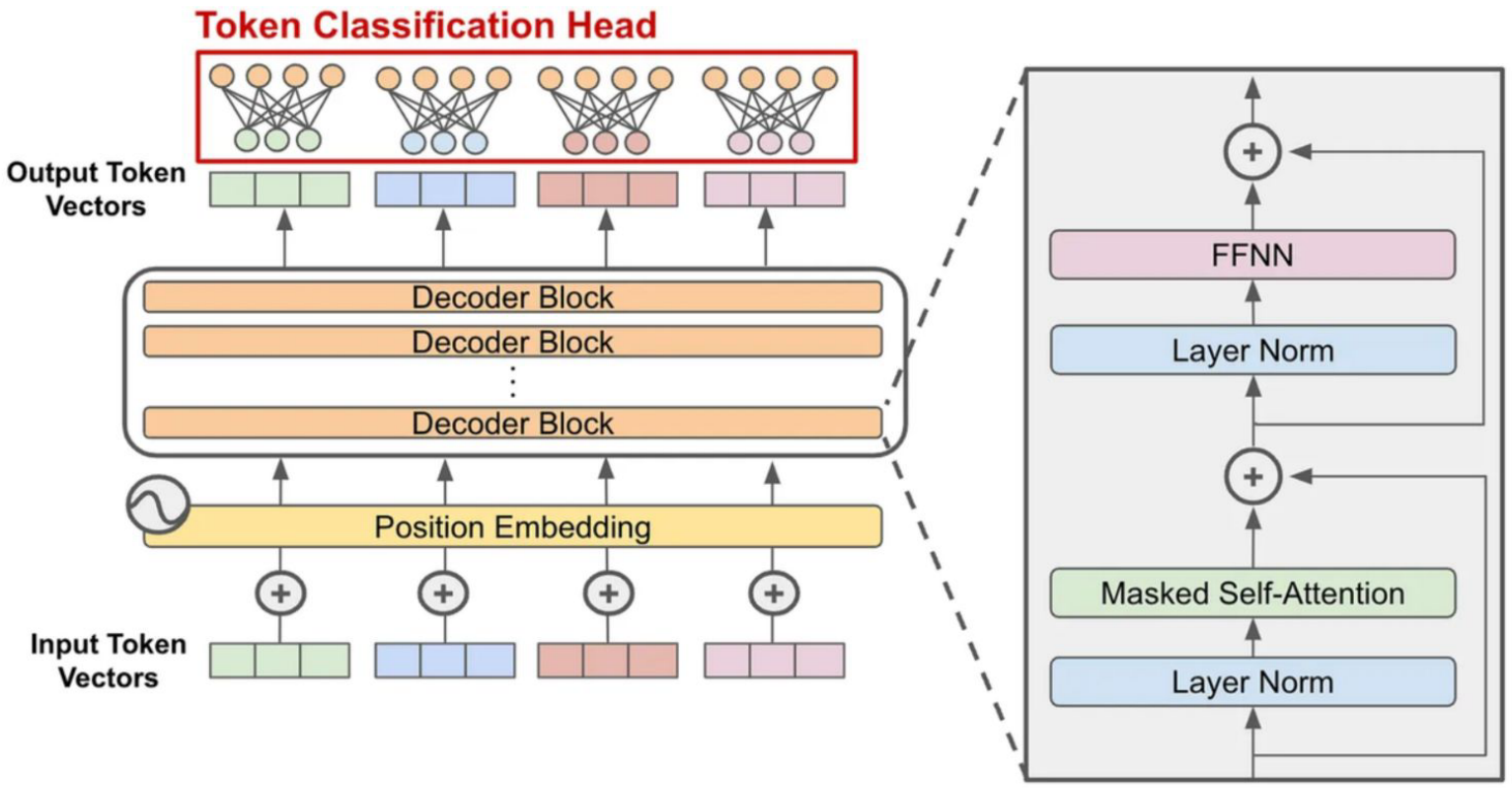
Base GPT 

Base Llama 

BA Pass


They understand that language

Expected output




input

Train  Max positions = 512

inference = 1024  > 512

Absolute position encodings — context length

---

Absolute

0 1 2 3 4

The dog chased another dog

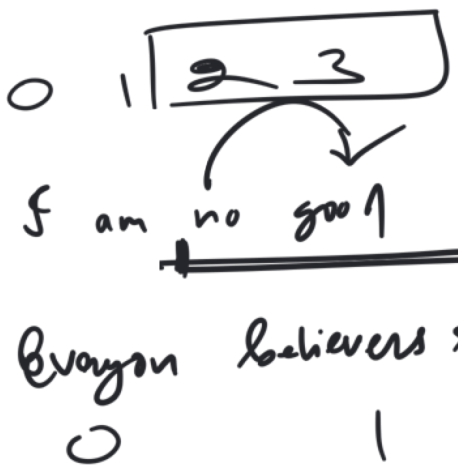
Relative

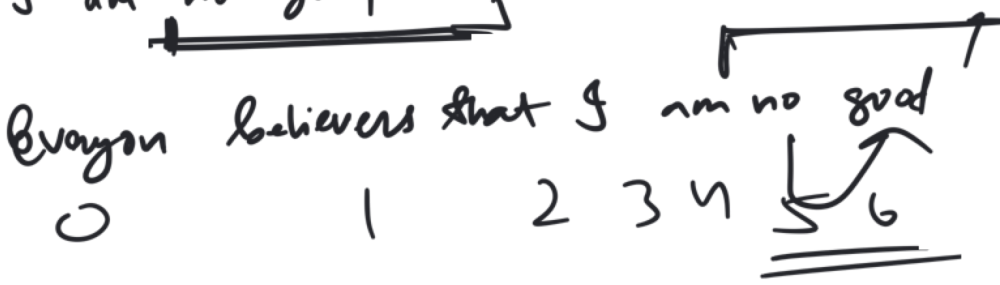
-2 -1 0 +1 +2

The dog chased another dog


phrase has same meaning

| | | |
|---|---|---|
| pos token no. | 10 | 11 |
| pos token | 110 | 111 |



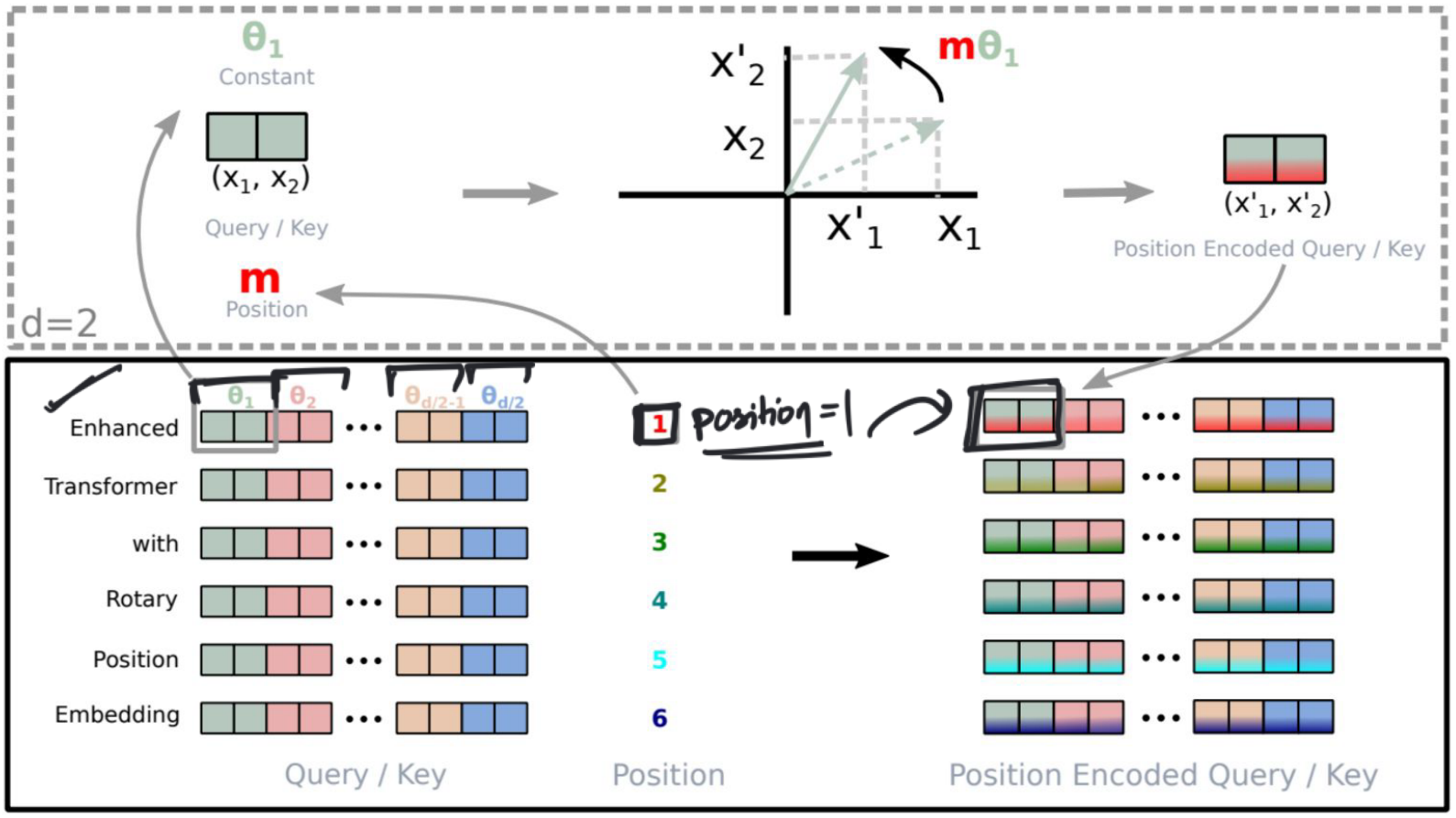
RoPE → encode the position as a rotation in vector space

instead of adding the pos info, rotate the Q, K vectors

↑

proportional to the position

rotate by pairs of the vector embedding



$Q_p \cdot K_q$

1024

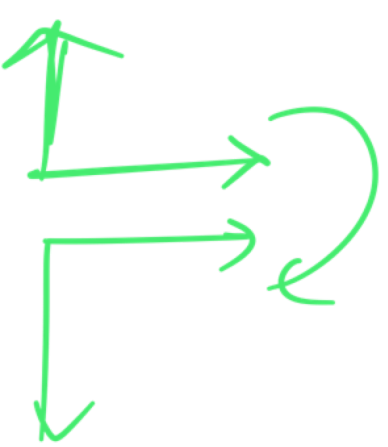
2028

(Q rotated by p) . K rotated by q

info is not $\not\propto$ p and q } no dependency on absolute pos

info $\propto$ (p - q) } RoPE dependence on diff of two positions

context length = 1M




original emb

Pair 1 | Pair 2

$x = [1, 0, 0, 1]$

L

rotated

$[0, 1, -0.7, 0.7]$

Pair 1 — 90°

Pair 2 = 45°

$(x, y) = 1, 0$

$$(x', y') = (x\cos\theta - y\sin\theta,\ x\sin\theta + y\cos\theta)$$

$x, y = 0, 1$

$(1, 0) \xrightarrow{90^\circ} (0, 1)$

$(0, 1) \xrightarrow{45^\circ} (-0.70, 0.70)$

$word_1$ emb

$word_2$ emb

$Rot(word_1, \theta_{251})$ . $Rot(word_2, \theta_{252})$

Q K

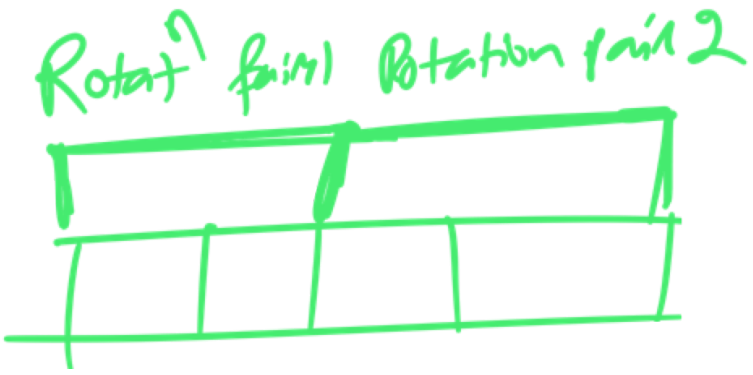
R

$Q.K = word1.word2\{(\theta_{252} - \theta_{251})\}$ vector

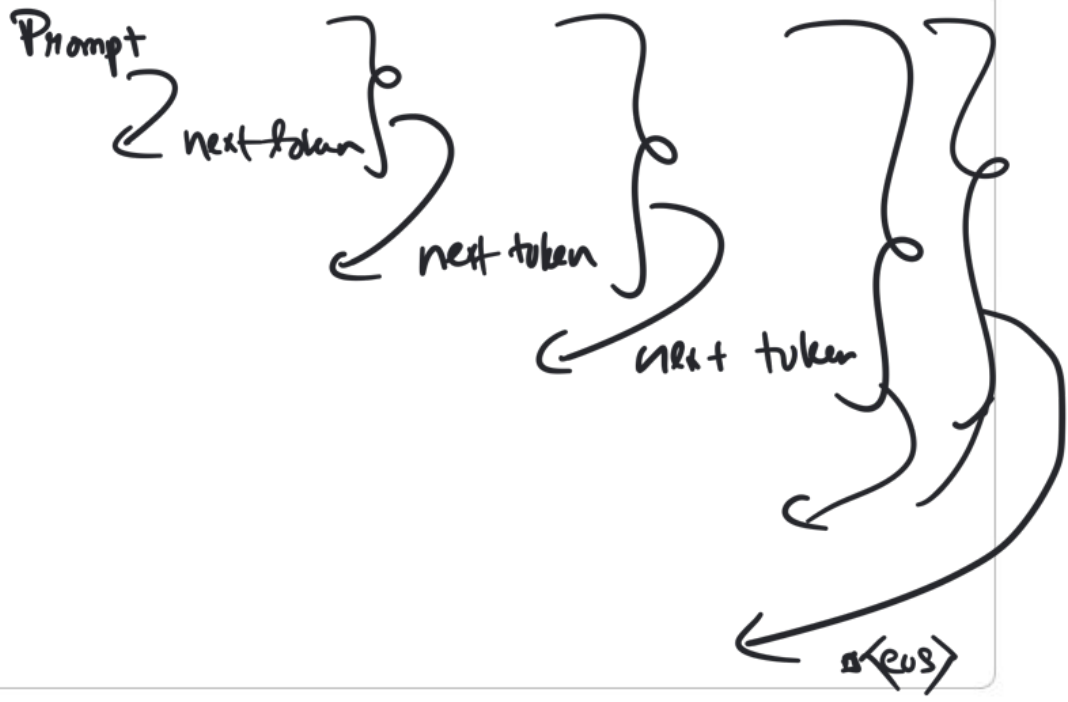
word1 word2

Deep dives RoPE

$\theta_k$ = position of token, rotation pair index



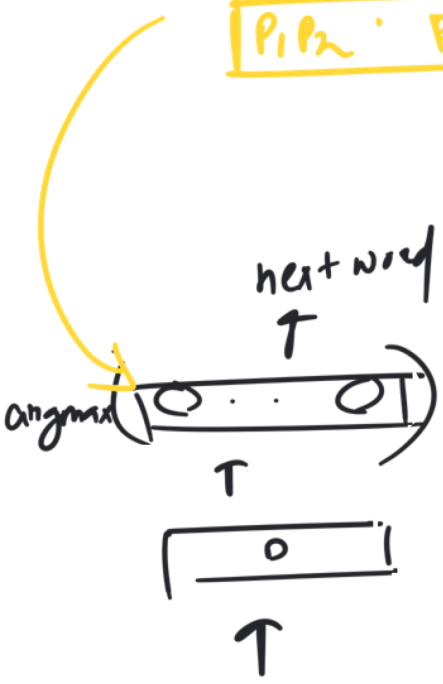
```
I have a dream (100%)
        |
    of (17.00%)
        |
  being (9.68%)
        |
     a (8.92%)
        |
 doctor (2.86%)
        |
     . (5.45%)
```



Decoding stops

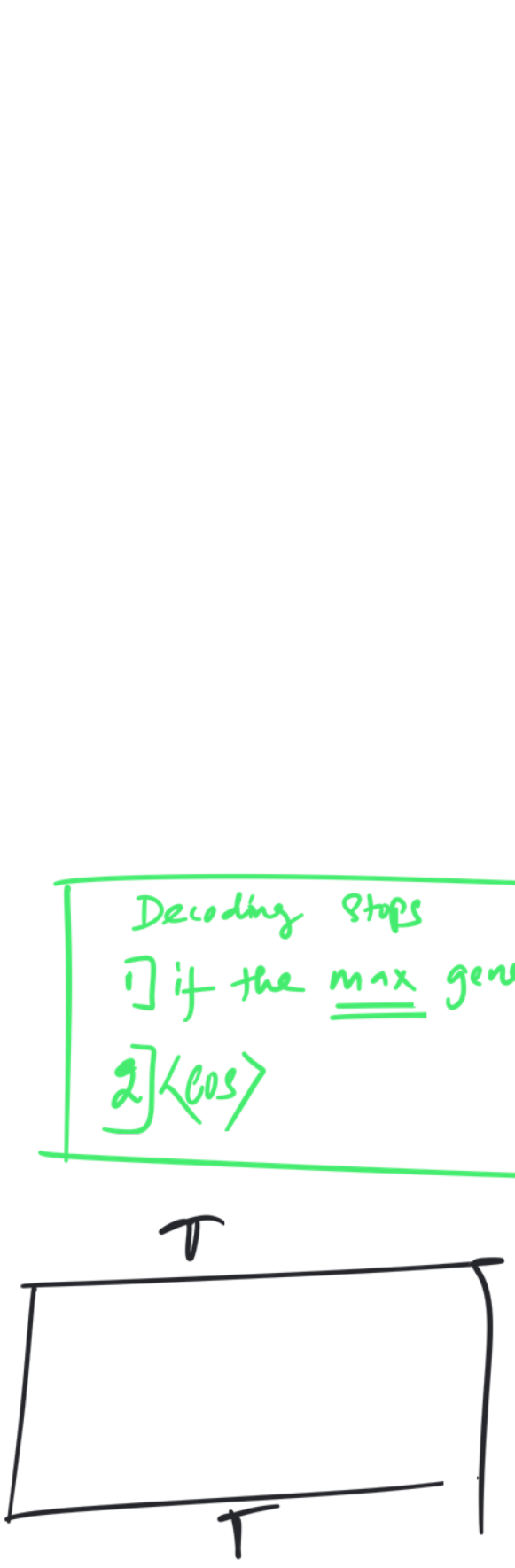
1] if the max generated token

2] <eos>




input


input + next word




input + next word + next word2

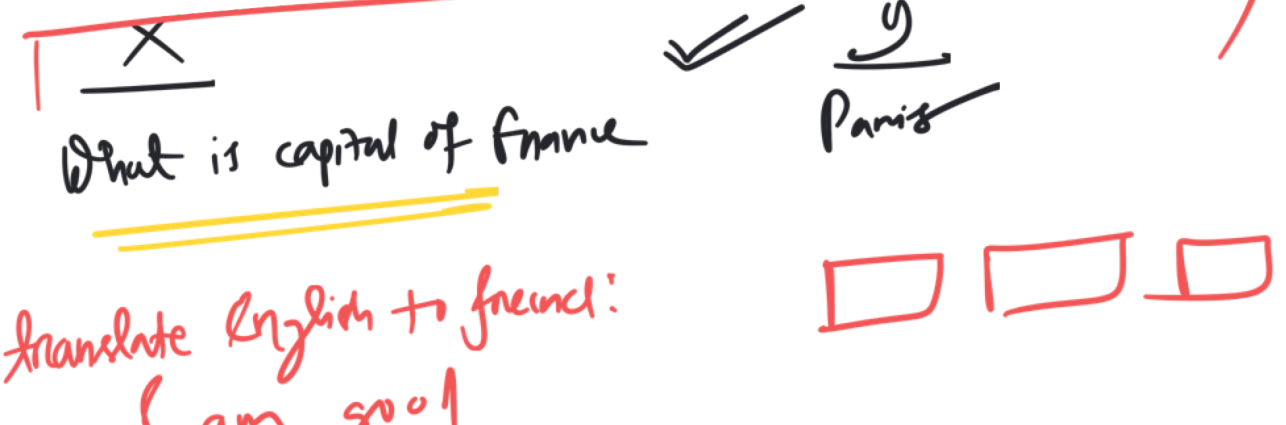
Preformed a Model

Base Model

The capital of France i.e Paris is a beautiful city.

Instruct Model

take the base




translate English to French:
I am good

* Instruction finetuning: Meta-learning

train data X | y

- translate english to french
  English sentence
  | french sentence

train 100 instructions

int

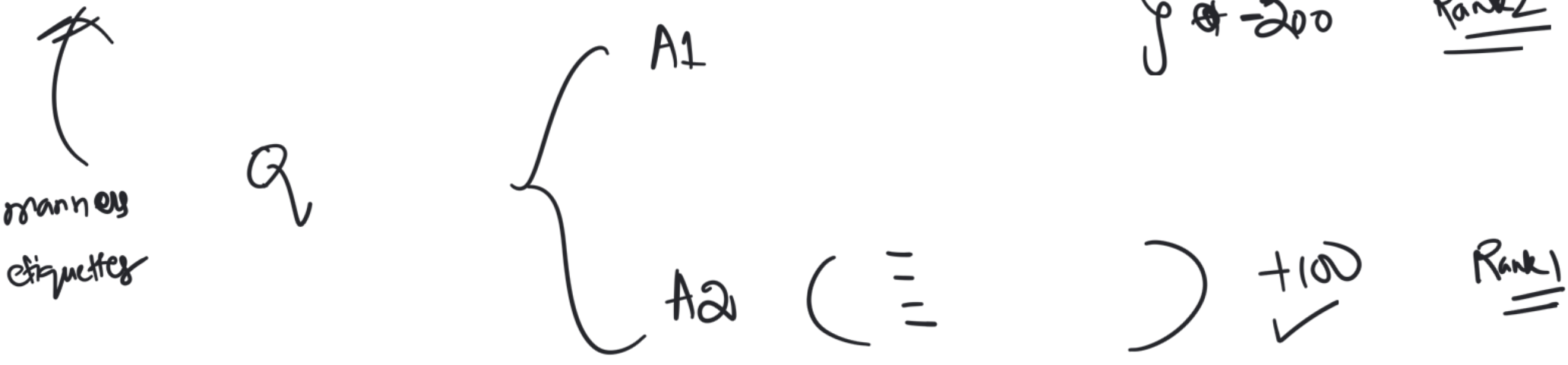
translate english: into german

another instruction

RLHF

Model - large chunks of Wiki/Reddit

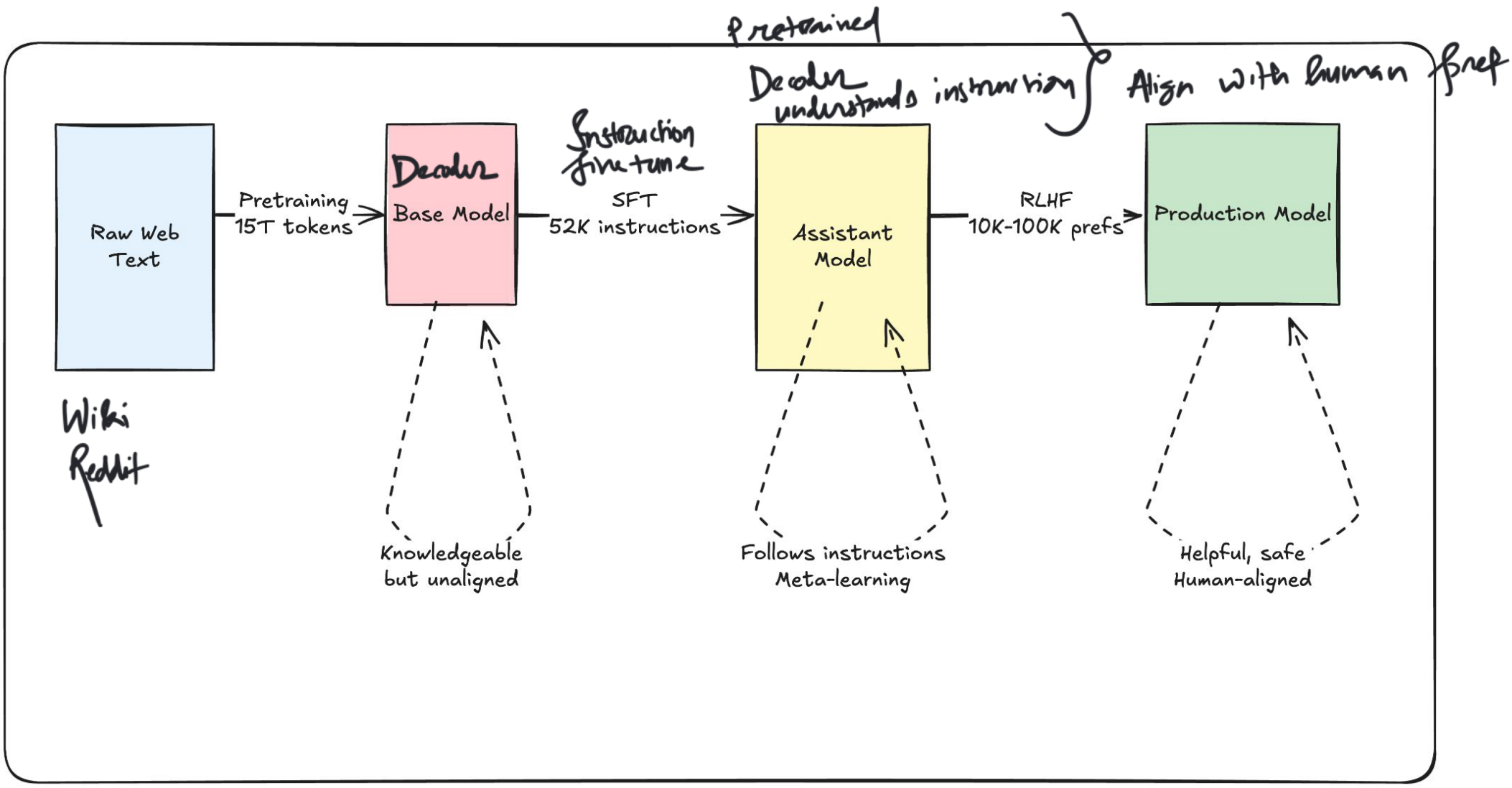
align it with human preferences

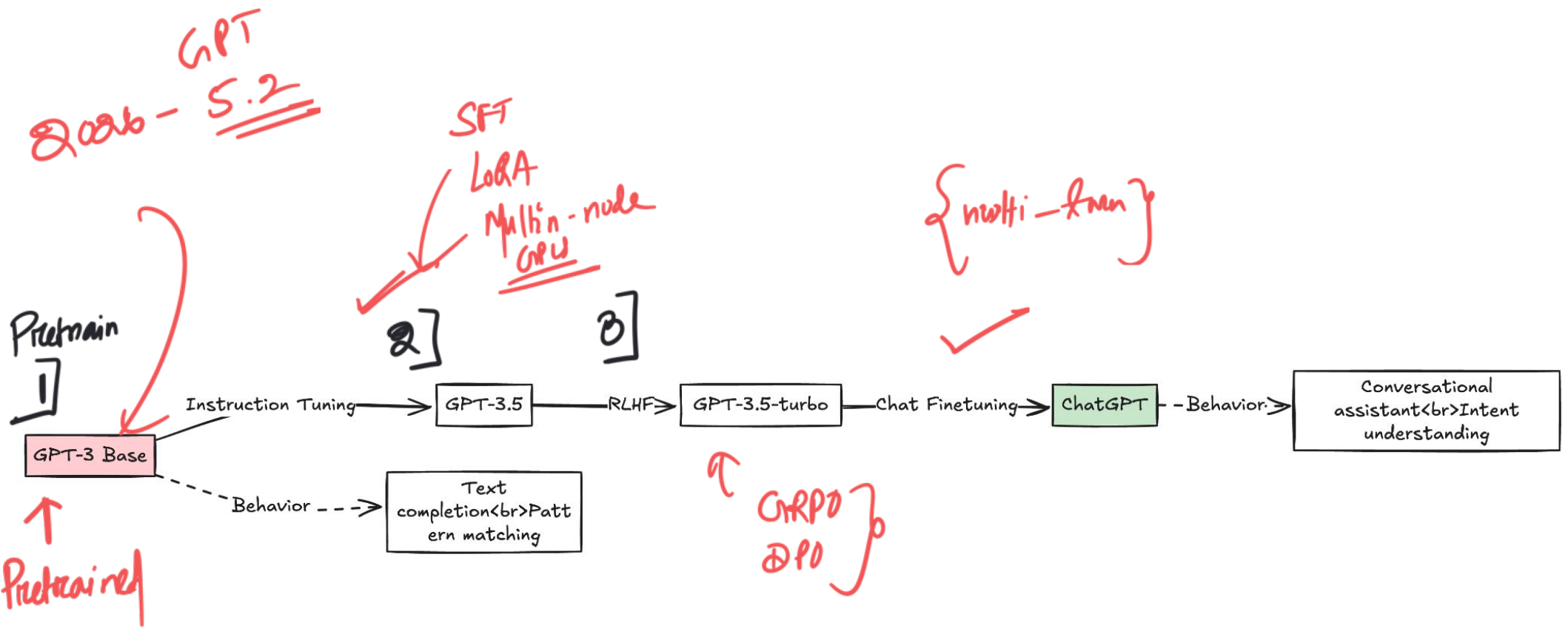


```
{
  "instruction": "Explain photosynthesis to a 10-year-old",
  "response": "Photosynthesis is how plants make food from sunlight! The green
  leaves act like tiny solar panels. They take in sunlight, water from the
  ground, and carbon dioxide from the air. Then, like a magical kitchen, they
  mix it all together to make sugar (food for the plant) and release oxygen
  that we breathe. That's why plants are so important—they make their own food
  and give us fresh air!"
}

{
  "instruction": "Translate to French: Hello, how are you?",
  "response": "Bonjour, comment allez-vous?"
}

{
  "instruction": "Write a Python function to reverse a string",
  "response": "Here's a simple function:\n\n```python\ndef reverse_string(s):\n
  return s[::-1]\n\n# Example usage\nprint(reverse_string('hello'))
  # Output: 'olleh'\n```"
}
```


Pretrained
Decoder understands instructions
Align with human pref
Raw Web Text
Pretraining 15T tokens
Decoder
Base Model
Instruction fine tune
SFT 52K instructions
Assistant Model
RLHF 10K-100K prefs
Production Model
Wiki
Reddit
Knowledgeable but unaligned
Follows instructions Meta-learning
Helpful, safe Human-aligned


GPT
2022 - 5.2
SFT
LoRA
Multi-node GPU
{multi-turn}
Pretrain
1]
2]
3]
Instruction Tuning
GPT-3.5
RLHF
GPT-3.5-turbo
Chat Finetuning
ChatGPT
Behavior
Conversational assistant<br>Intent understanding
GPT-3 Base
Behavior
Text completion<br>Pattern matching
Pretrained
GRPO
DPO

1] Pretraining → Decoder

2] Instruction FT

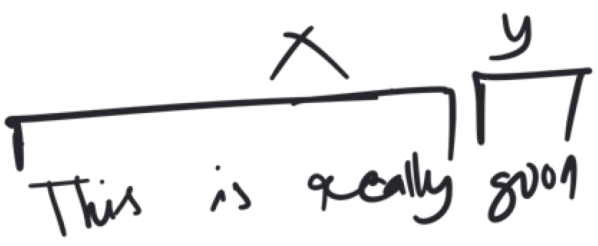
3] RL

Supervised ✓ Human annotation

Self-supervised ✓ Human preference

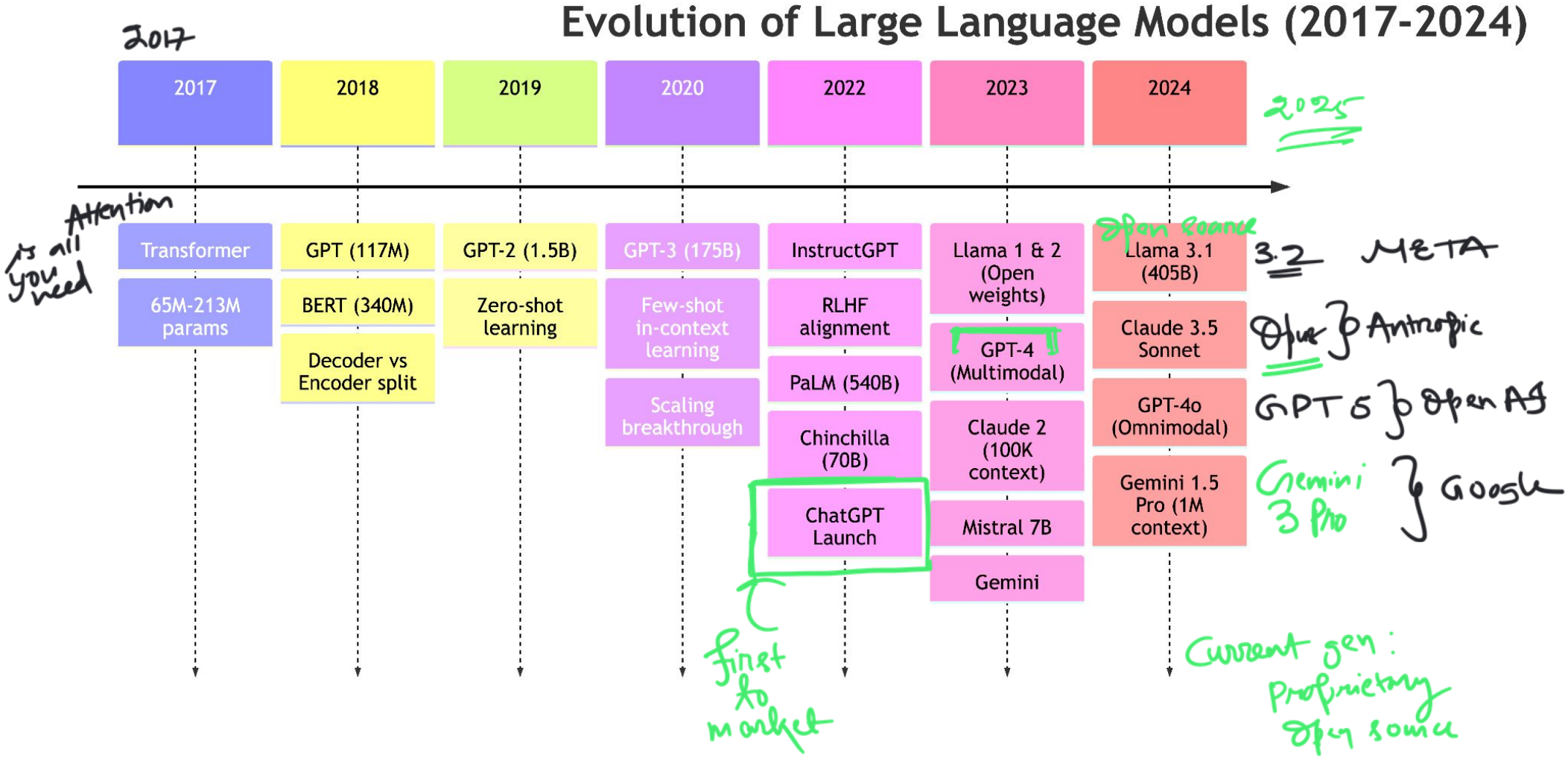
Unsupervised

1) Decoder only
2) Modern LLM (long context)
3) Only pretraining — SFT
4) Align — RL



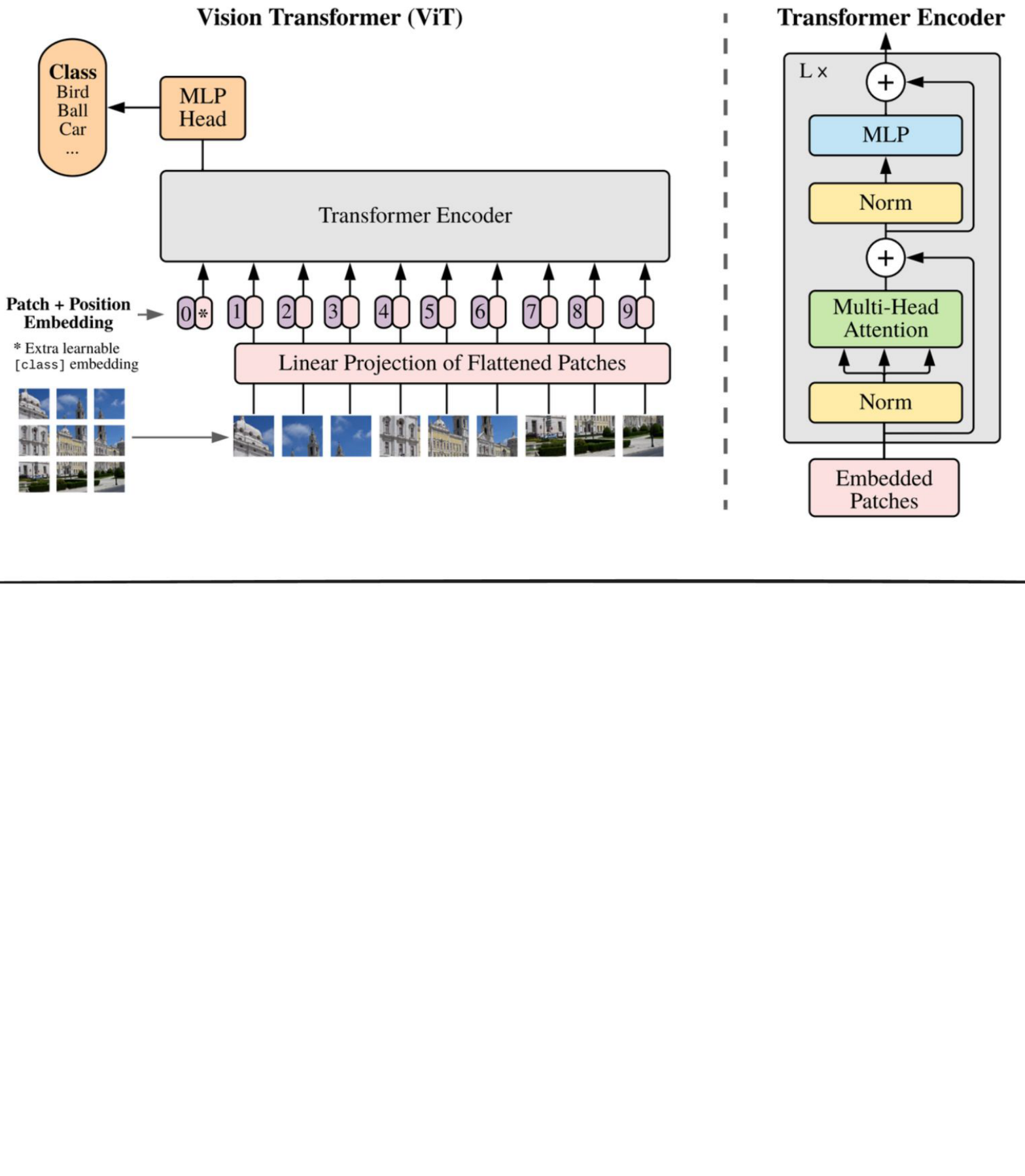
Reinforcement Learning (LLMs)
GRPO
Good Planning + Reasoning =
Deepseek (self correct)

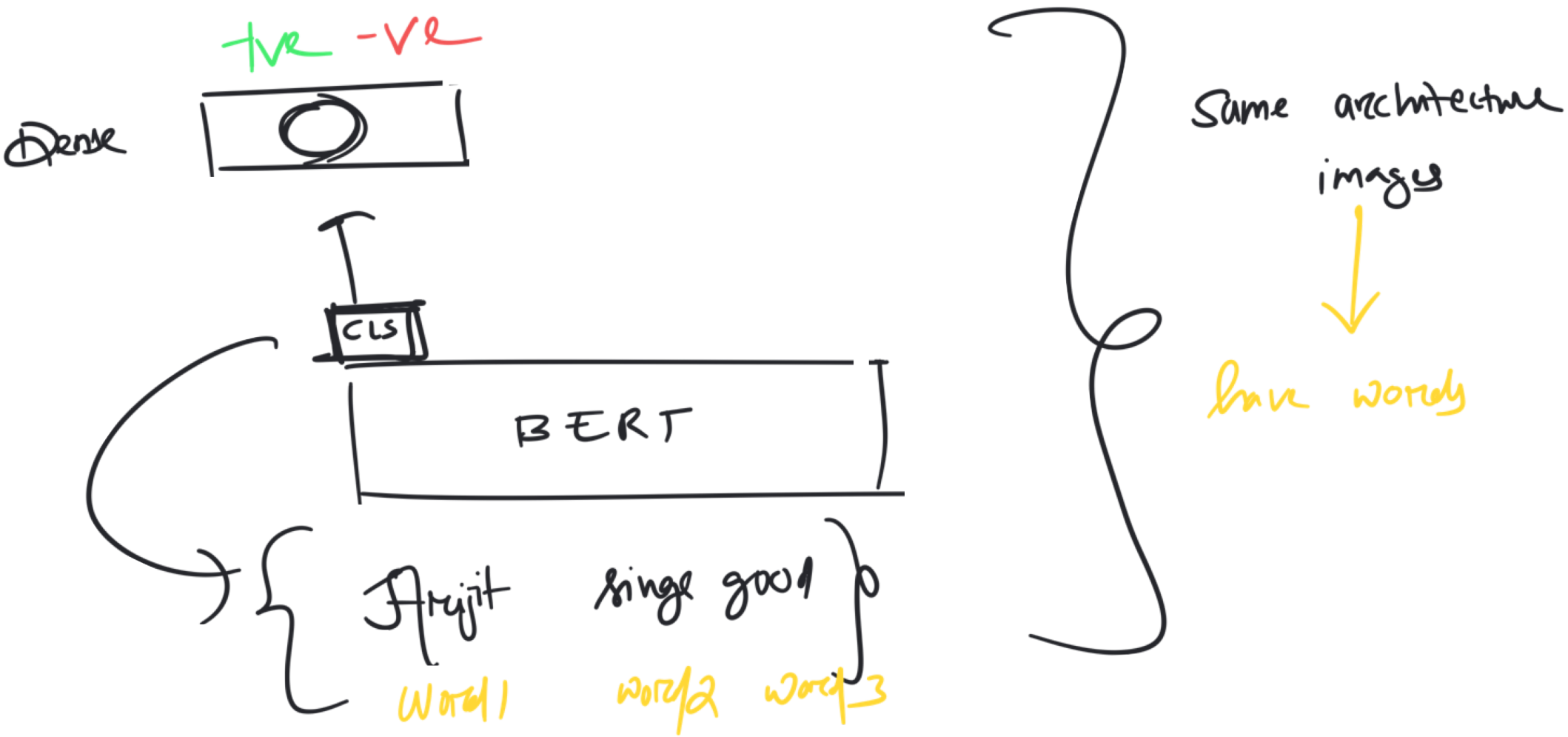
# Evolution of Large Language Models (2017-2024)

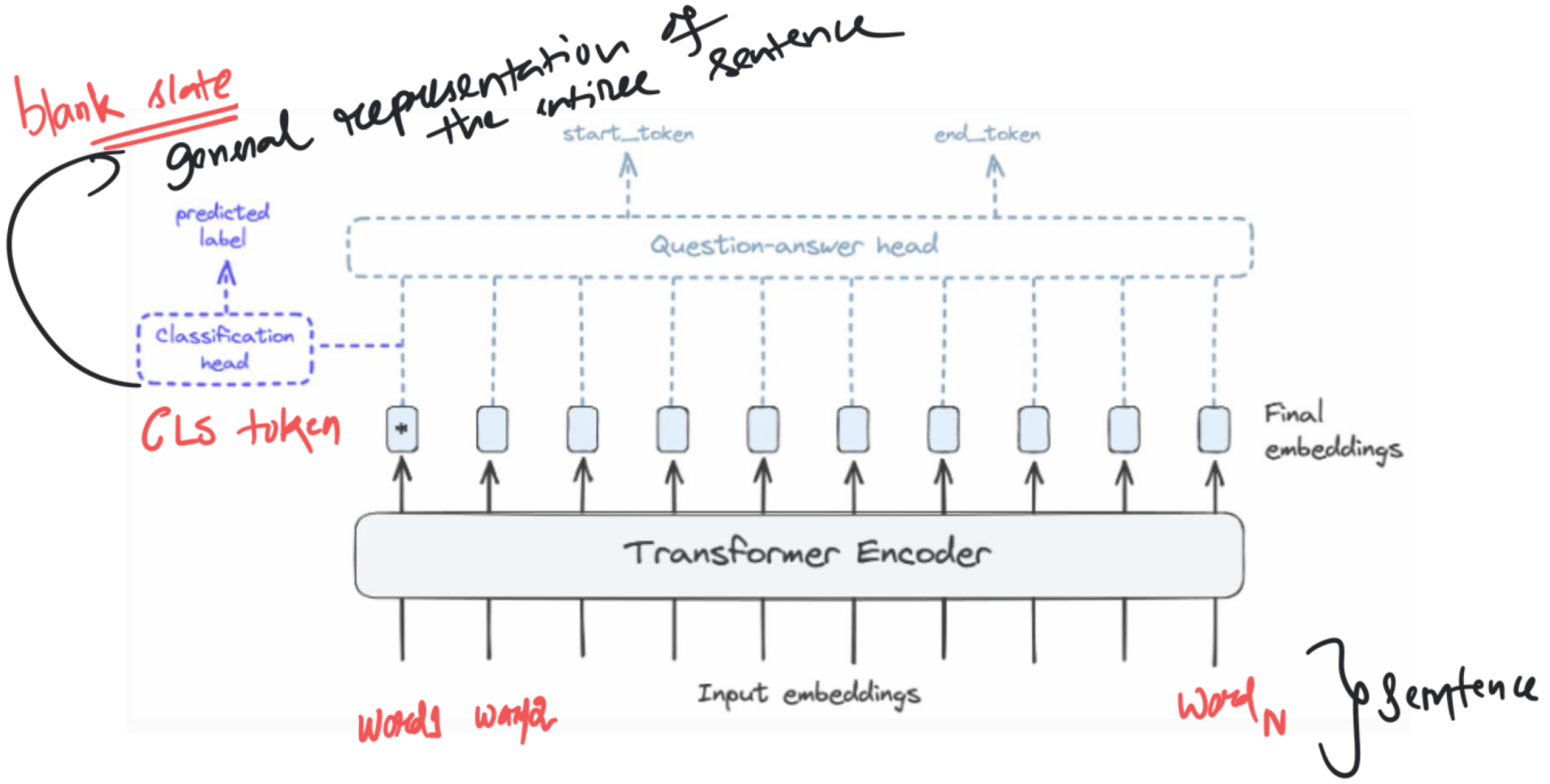


BERT

Vision Transformers

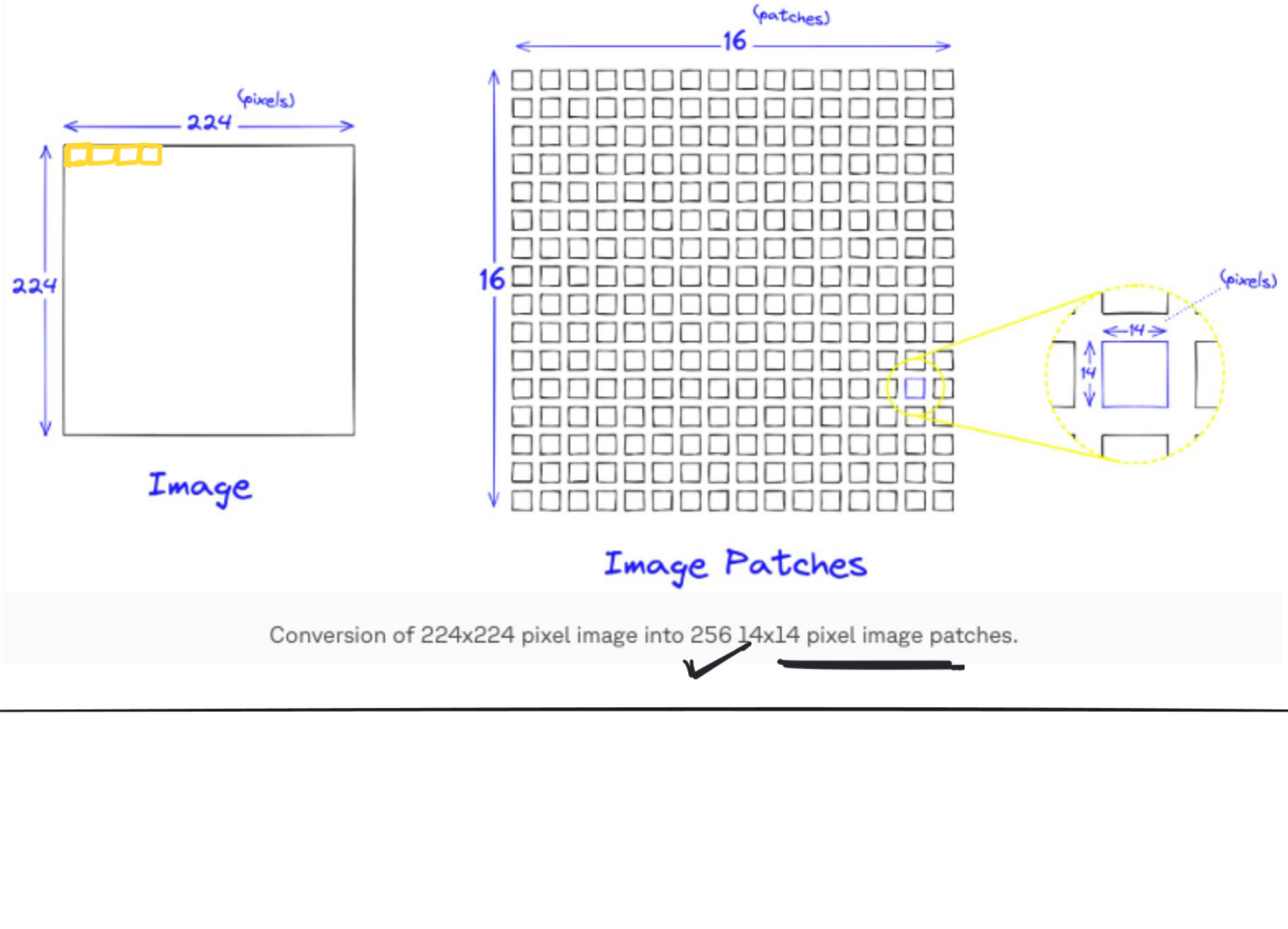

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP
Head
Transformer Encoder
Patch + Position
Embedding
* Extra learnable
[class] embedding
0*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head
Attention
Norm
Embedded
Patches


+VE -VE
Dense
CLS
BERT
Arijit
Singe good
Word1
word2
word3
Same architecture
image
have words

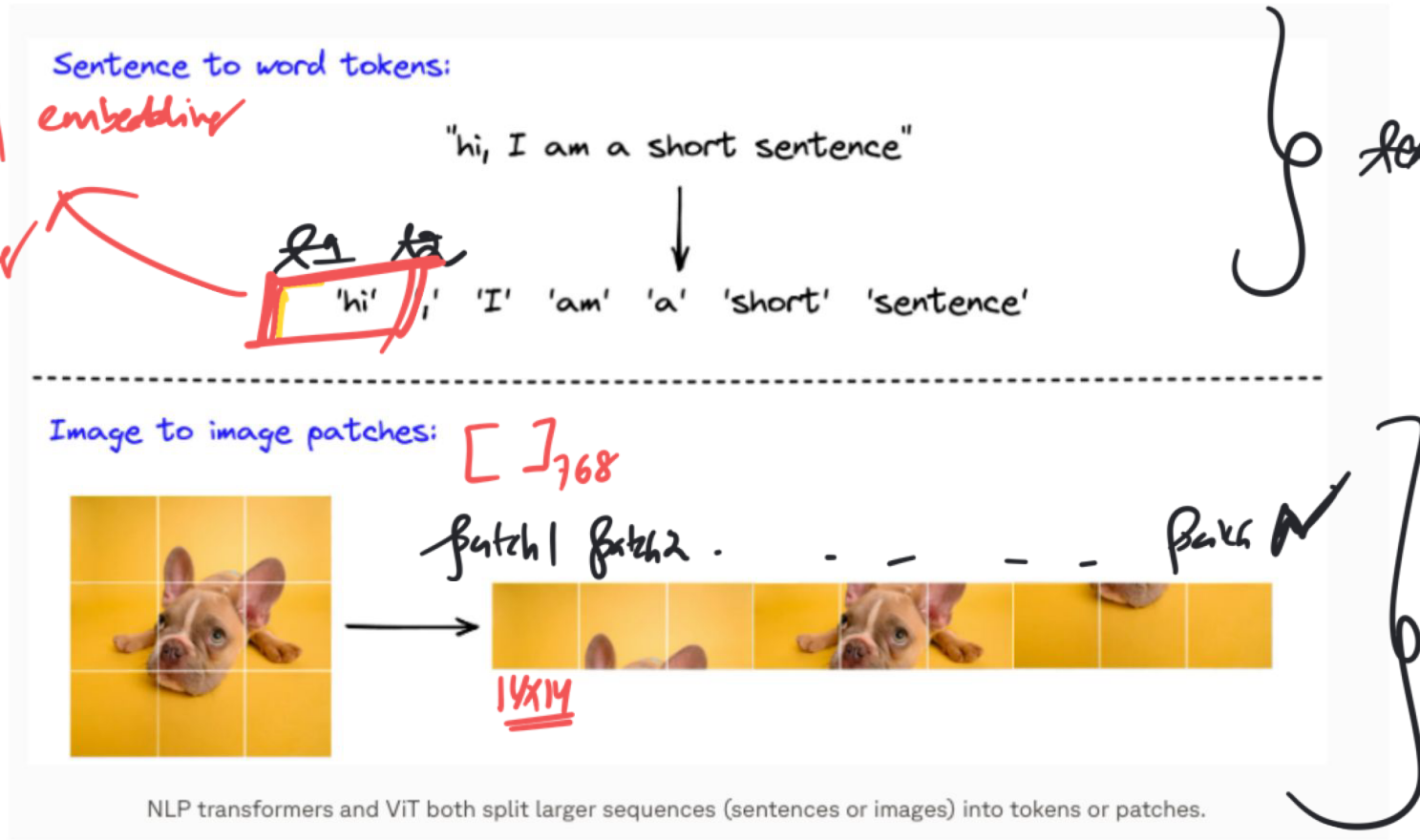

blank slate
general representation of the entire sentence
start_token
end_token
predicted label
Question-answer head
Classification head
CLS token
Final embeddings
Transformer Encoder
Input embeddings
Word1
Word2
WordN
sentence

individual pixel values = words

224x224 ✓




Conversion of 224x224 pixel image into 256 14x14 pixel image patches.




NLP transformers and ViT both split larger sequences (sentences or images) into tokens or patches.

i] 14x14 } → [ ]512

i] flatten

ii] CNN + FF }

iii] ff

Feature extractor

Dense layer

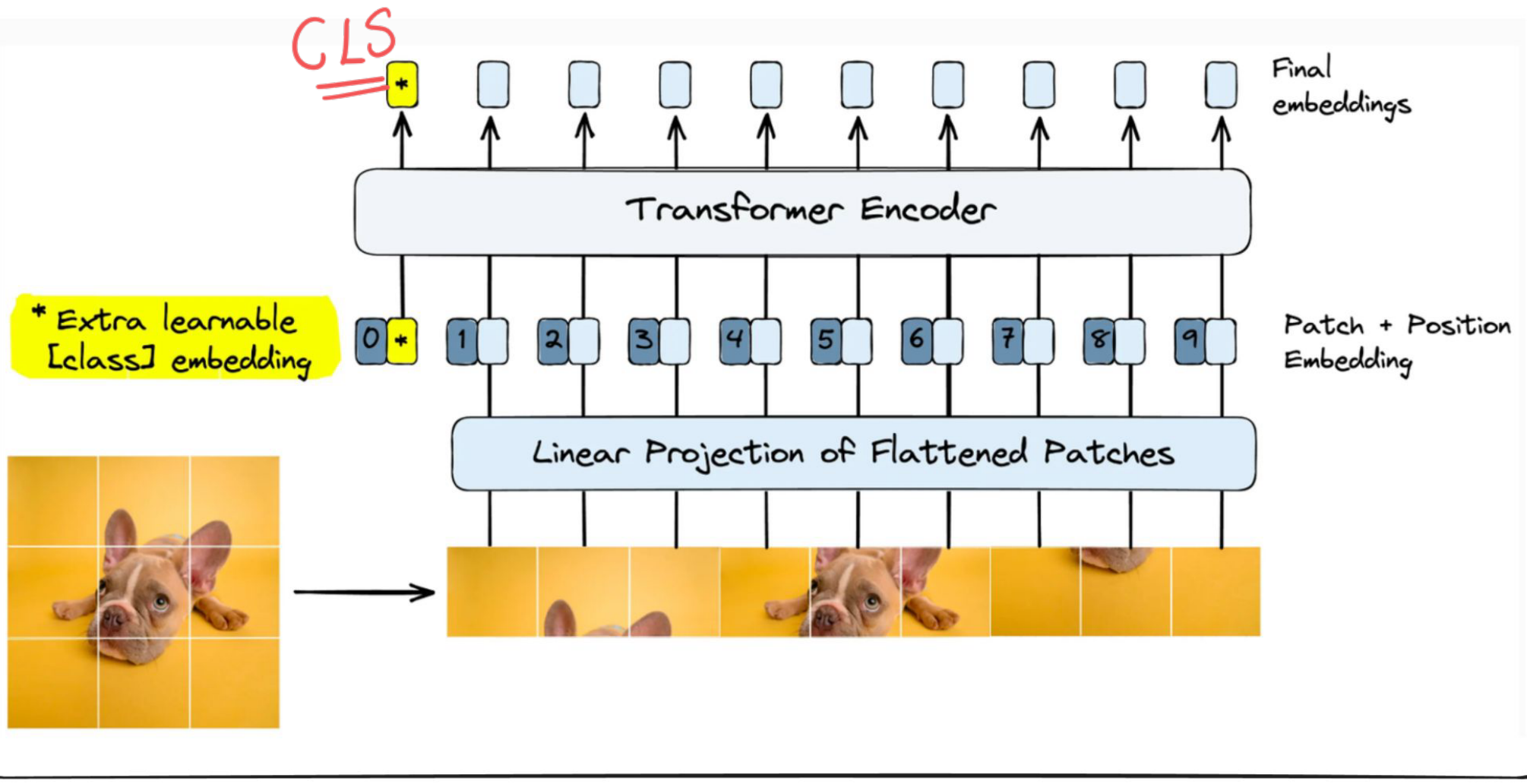
Flatten { 14x14 } grow pixels

Feature Extractor

Linear projection

768

PATCH EMBEDDINGS

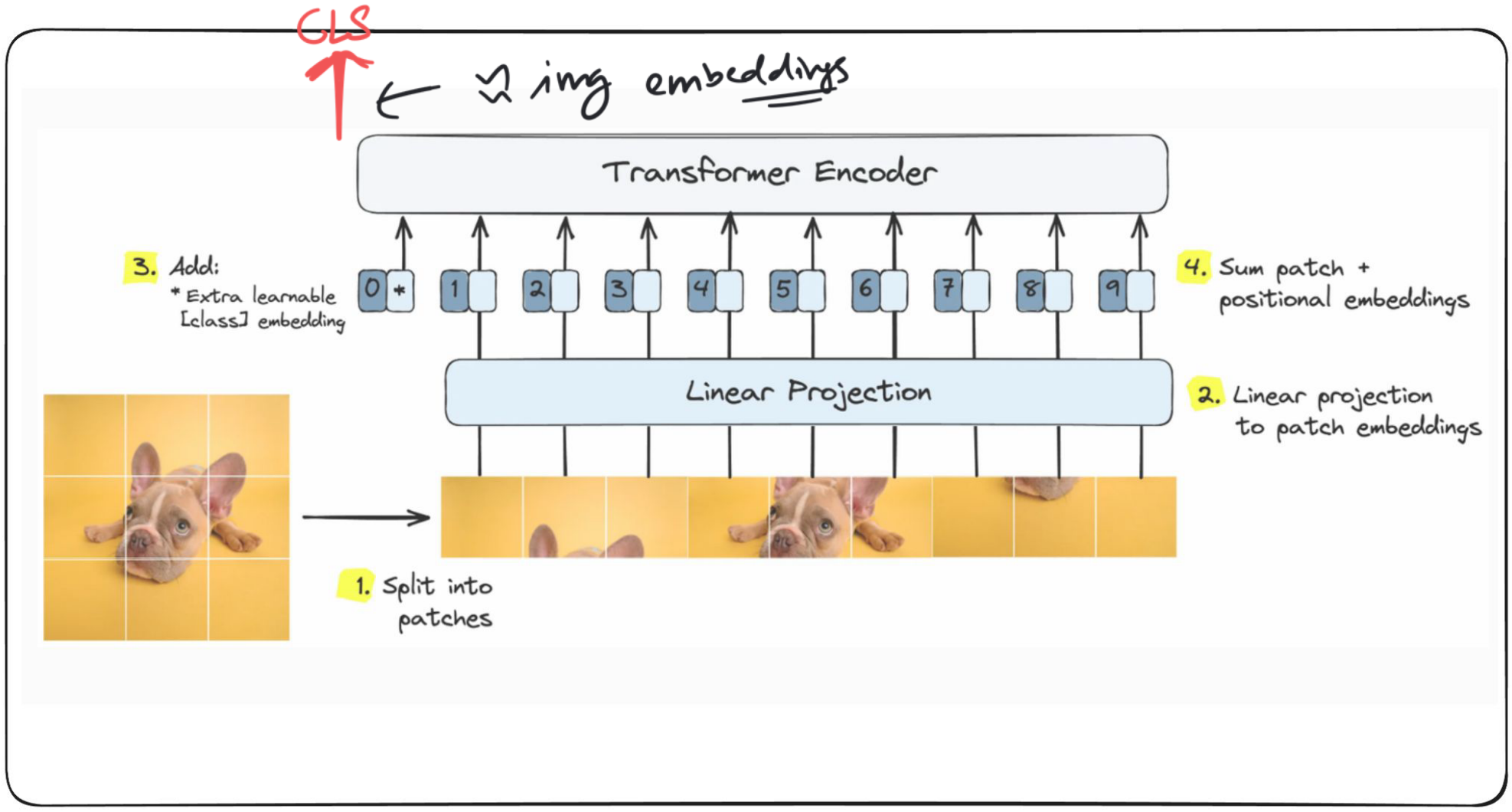

CLS
*
Final embeddings
Transformer Encoder
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Patch + Position Embedding
Linear Projection of Flattened Patches

softmax

no. of op classes

CLS

img embeddings



Step0 · 224 x 224 x 3

Step1 Patches — 196 Patch

Step2 Conv/Dense — 196 x 16 x 16 x 3 ⟶ 196 x 768

Step3 Position encoding — 196 x 768 } position encode

Step4 CLS (196 + 1) x 768

CLS

Step5 Transformers

Step6 CLS emb

Step7 — Dense (768, no. of class)

P1 P2

Image Input

Text Input

An Image of a Dog

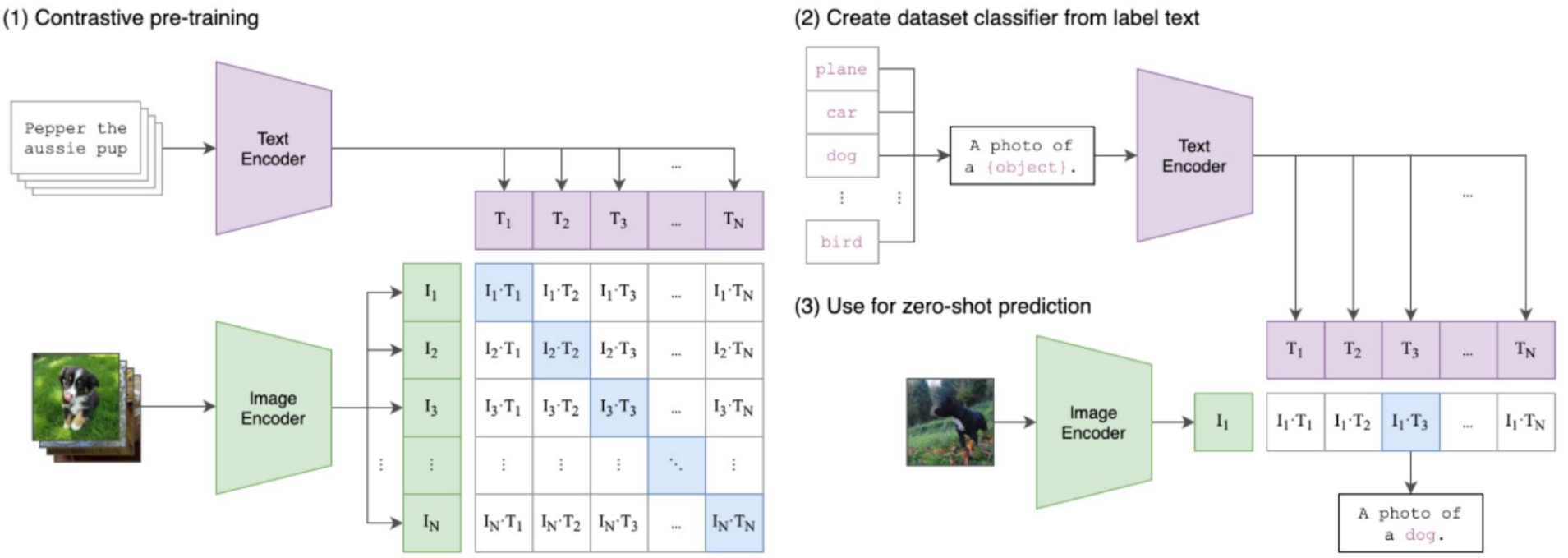
Vision Encoder
ViT or ResNet

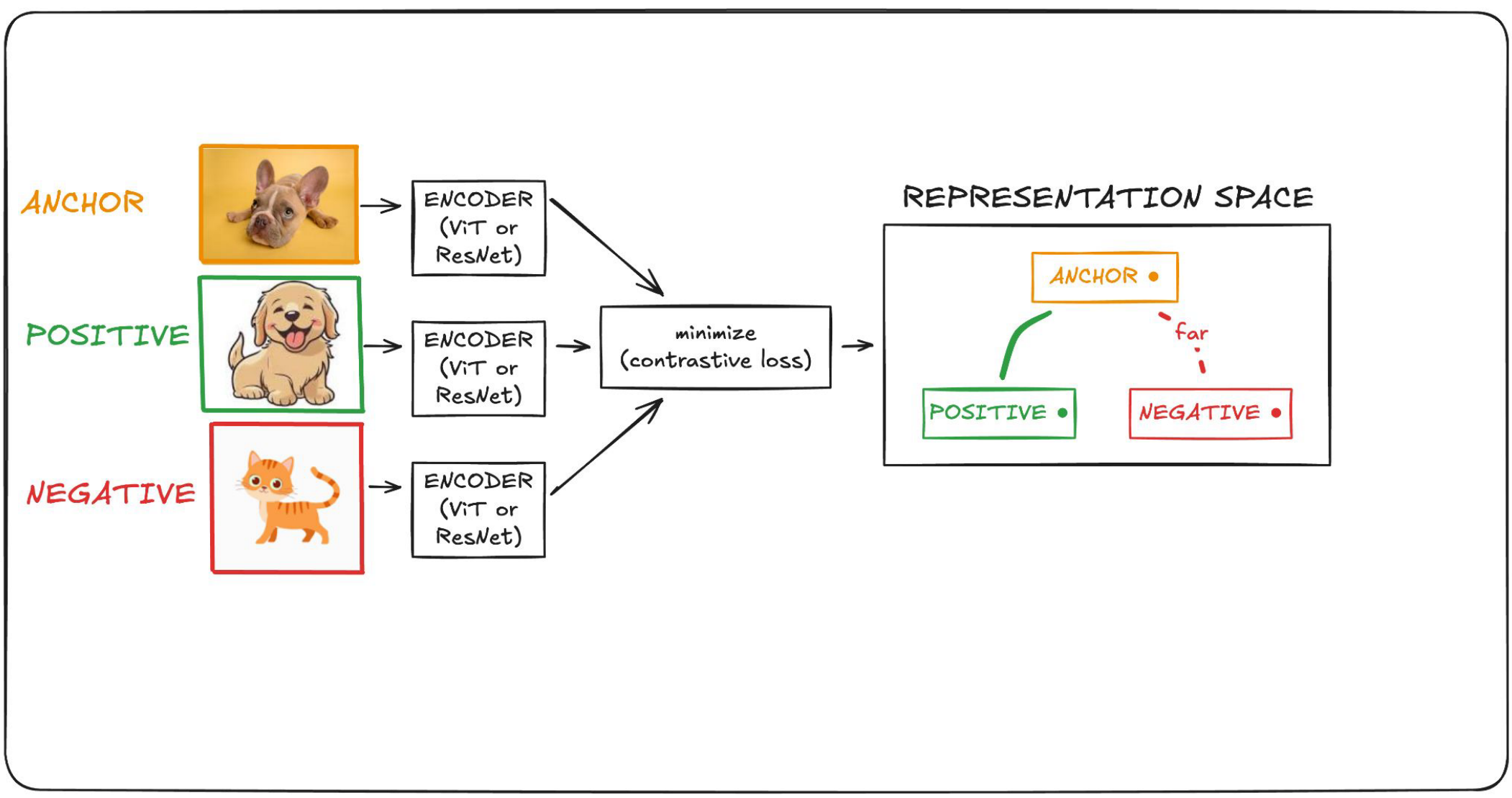
Text Encoder
Transformer
(BERT)

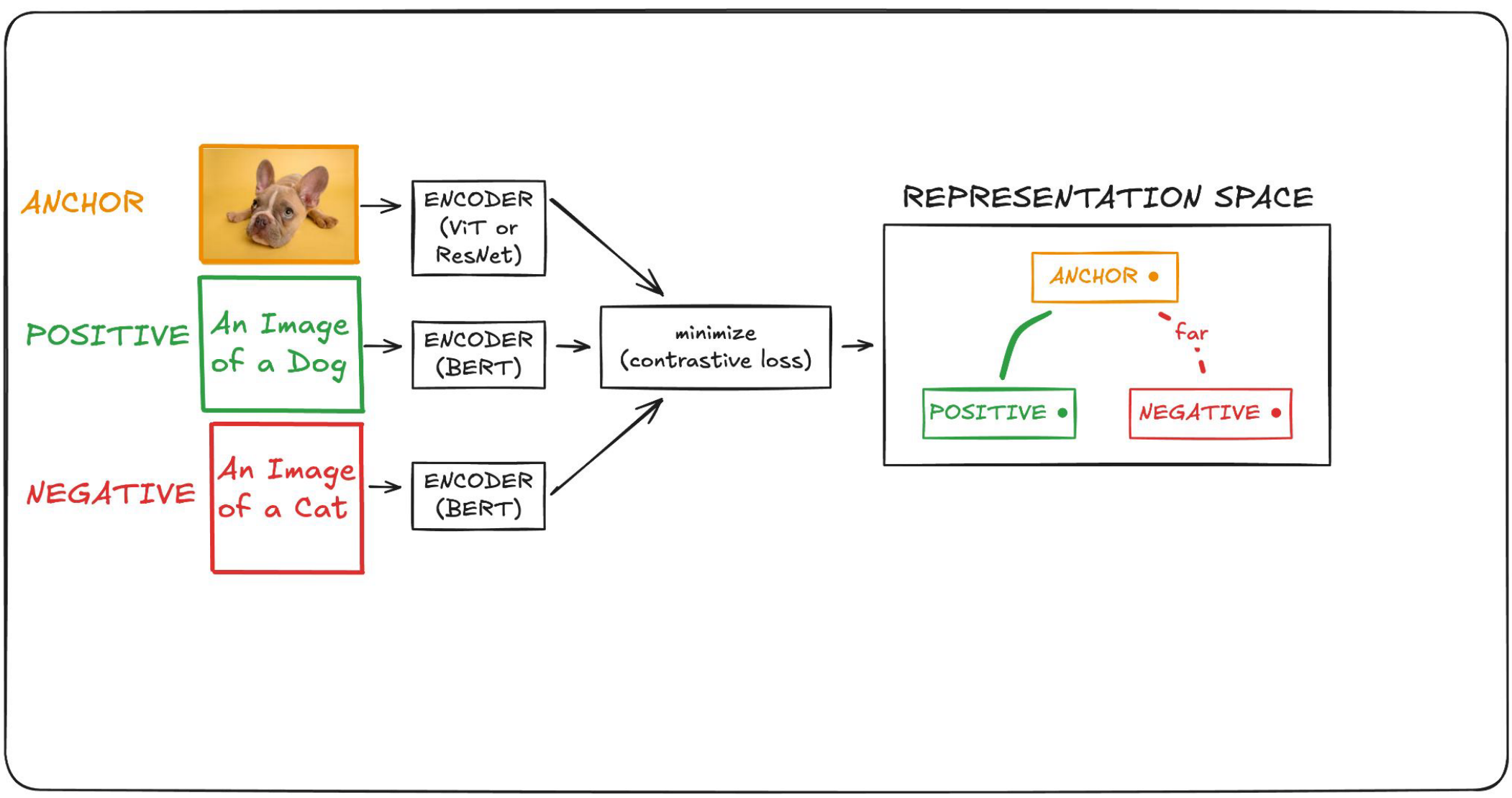
Joint Embedding Space
512-D vectors

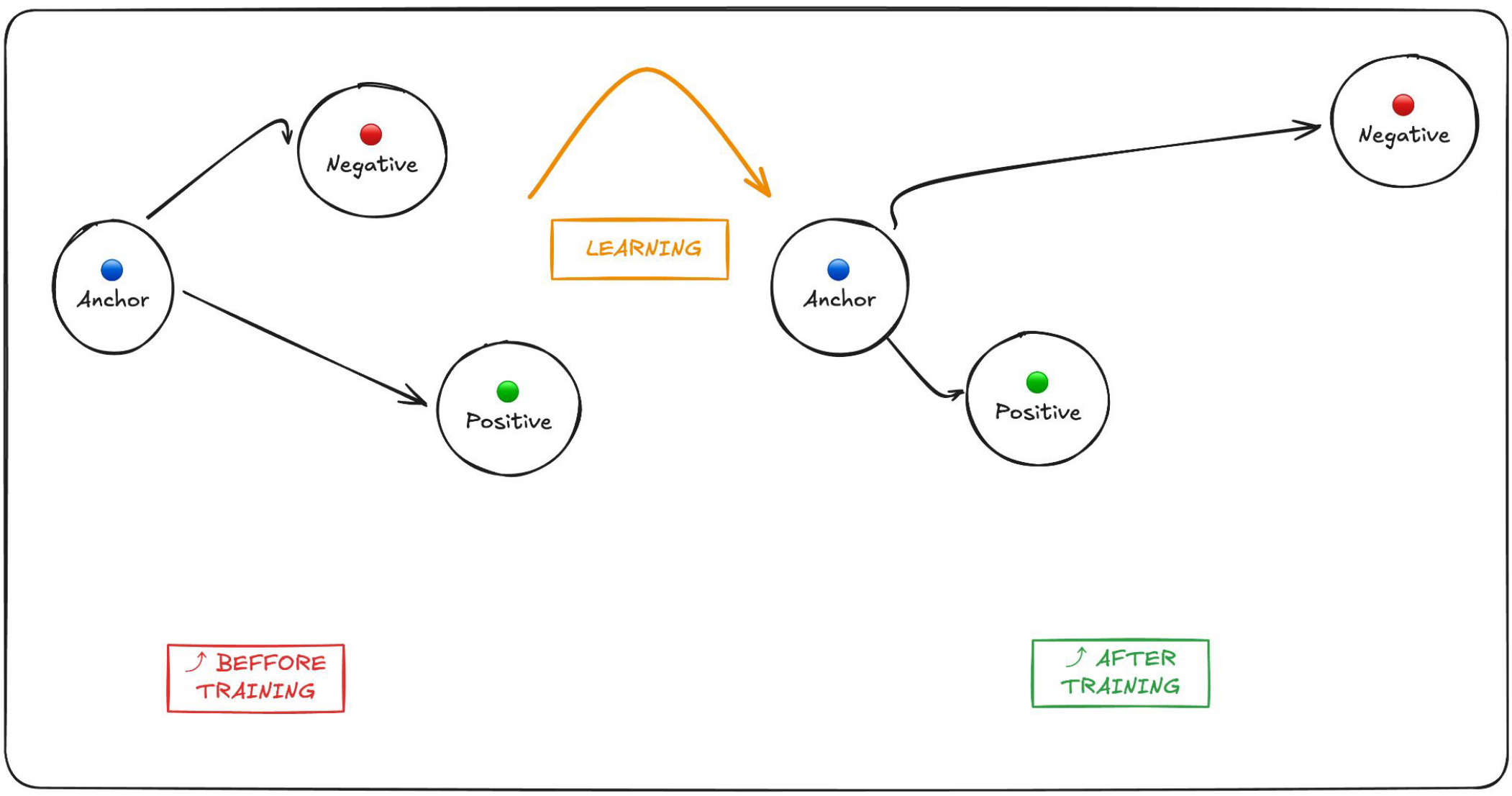
Match & Compare

# Approach


(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
$T_1$ $T_2$ $T_3$ ... $T_N$
Image Encoder
$I_1$ $I_2$ $I_3$ $I_N$
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
$I_1$
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of a dog.


ANCHOR
POSITIVE
NEGATIVE
ENCODER (ViT or ResNet)
ENCODER (ViT or ResNet)
ENCODER (ViT or ResNet)
minimize (contrastive loss)
REPRESENTATION SPACE
ANCHOR
far
POSITIVE
NEGATIVE


ANCHOR
ENCODER (ViT or ResNet)
POSITIVE
An Image of a Dog
ENCODER (BERT)
minimize (contrastive loss)
NEGATIVE
An Image of a Cat
ENCODER (BERT)
REPRESENTATION SPACE
ANCHOR
far
POSITIVE
NEGATIVE


Negative
Anchor
Positive
LEARNING
Anchor
Negative
Positive
↗ BEFFORE TRAINING
↗ AFTER TRAINING