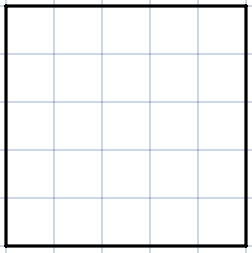
CLIP:



model → My cute cat
↳ A good representor of cat
↳ Can be fine-tuned

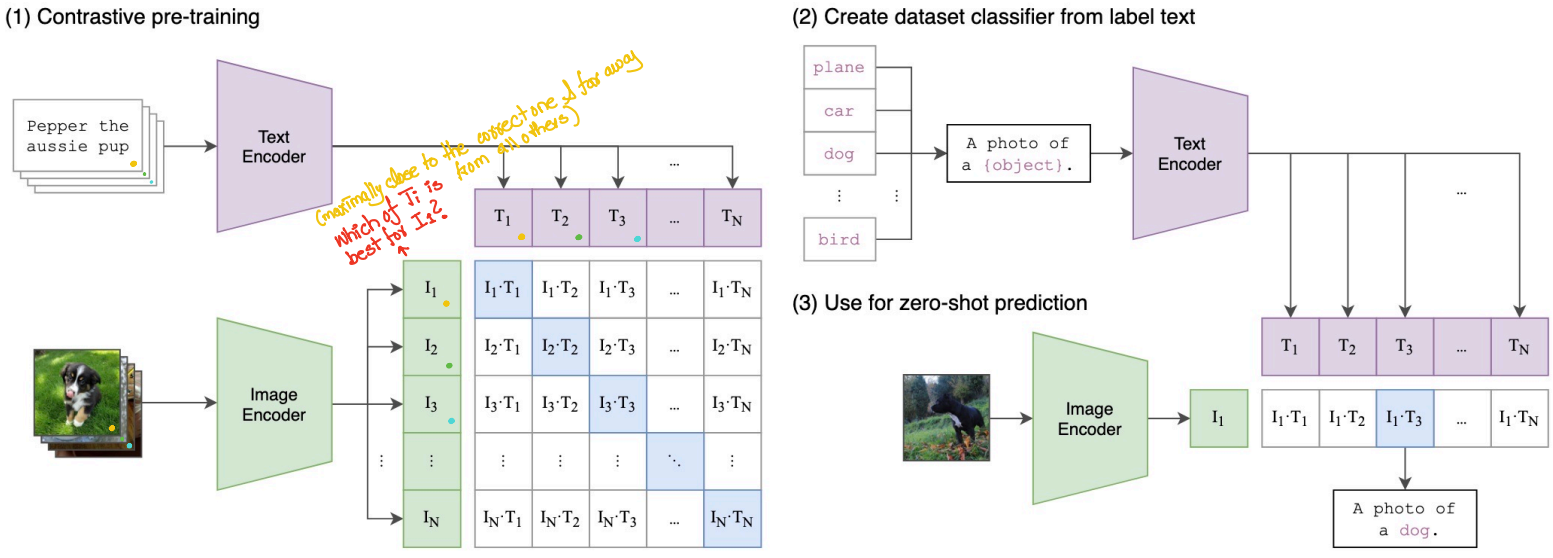


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

The Training Objective:

- Maximize similarity for correct pairs (diagonal elements) - these should be "maximally close"
- Minimize similarity for incorrect pairs (off-diagonal elements) - these should be "far away"

Why It's Called "Contrastive":

- It contrasts positive examples (what goes together) against negative examples (what doesn't go together)
- The model learns by comparing: "this image matches this text" vs "this image doesn't match these other texts"

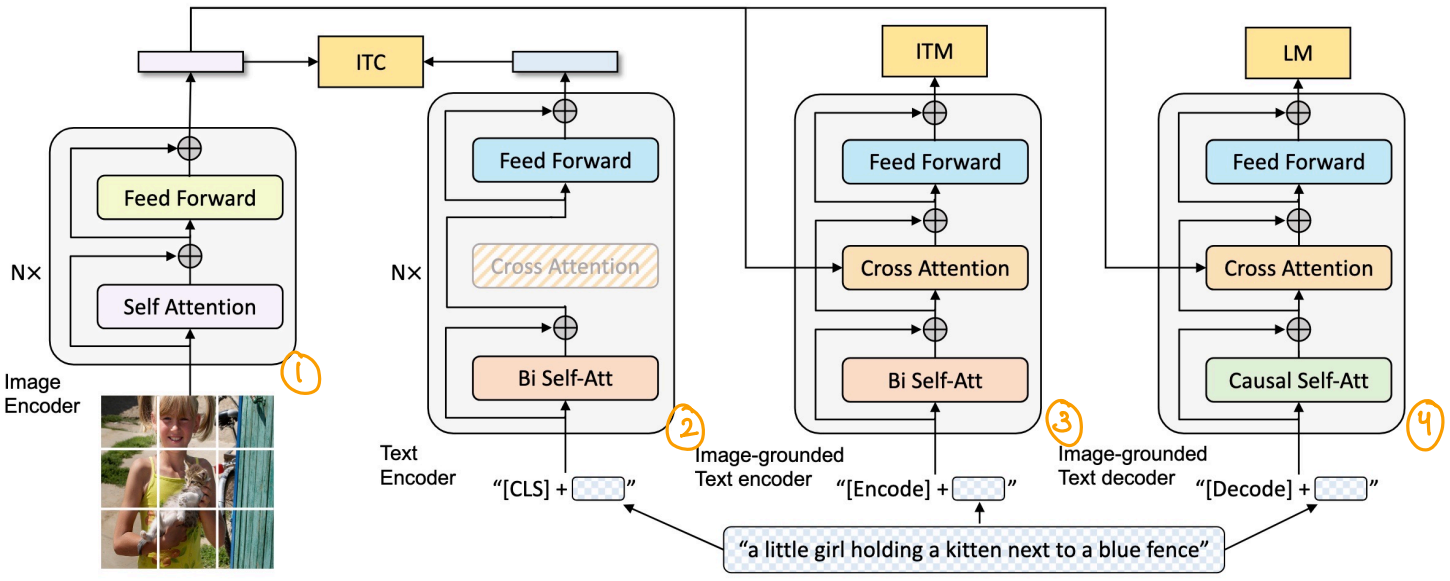
Example:

- If you have an image of a cat and descriptions like "a cat sitting", "a dog running", "a car driving":
  - The model learns that the cat image is most similar to "a cat sitting" (positive)
  - And least similar to "a dog running" or "a car driving" (negatives)

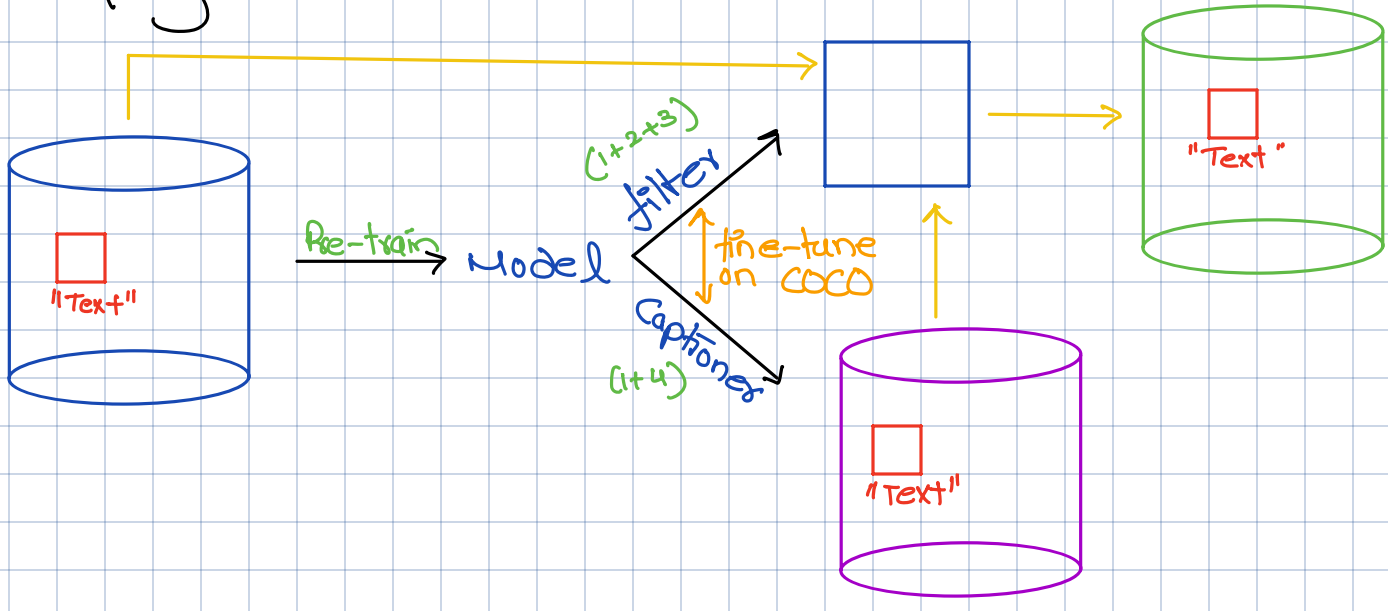
- This approach is powerful because it learns meaningful representations by understanding relationships and differences between data points, rather than just trying to predict a single correct answer.

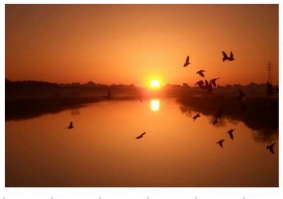
BLIP

Model:



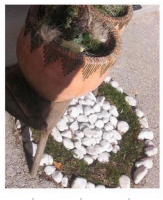
Boot-straping:





T_w: "from bridge near my house"

T_s: "a flock of birds flying over a lake at sunset"



T_w: "in front of a house door in Reichenfels, Austria"

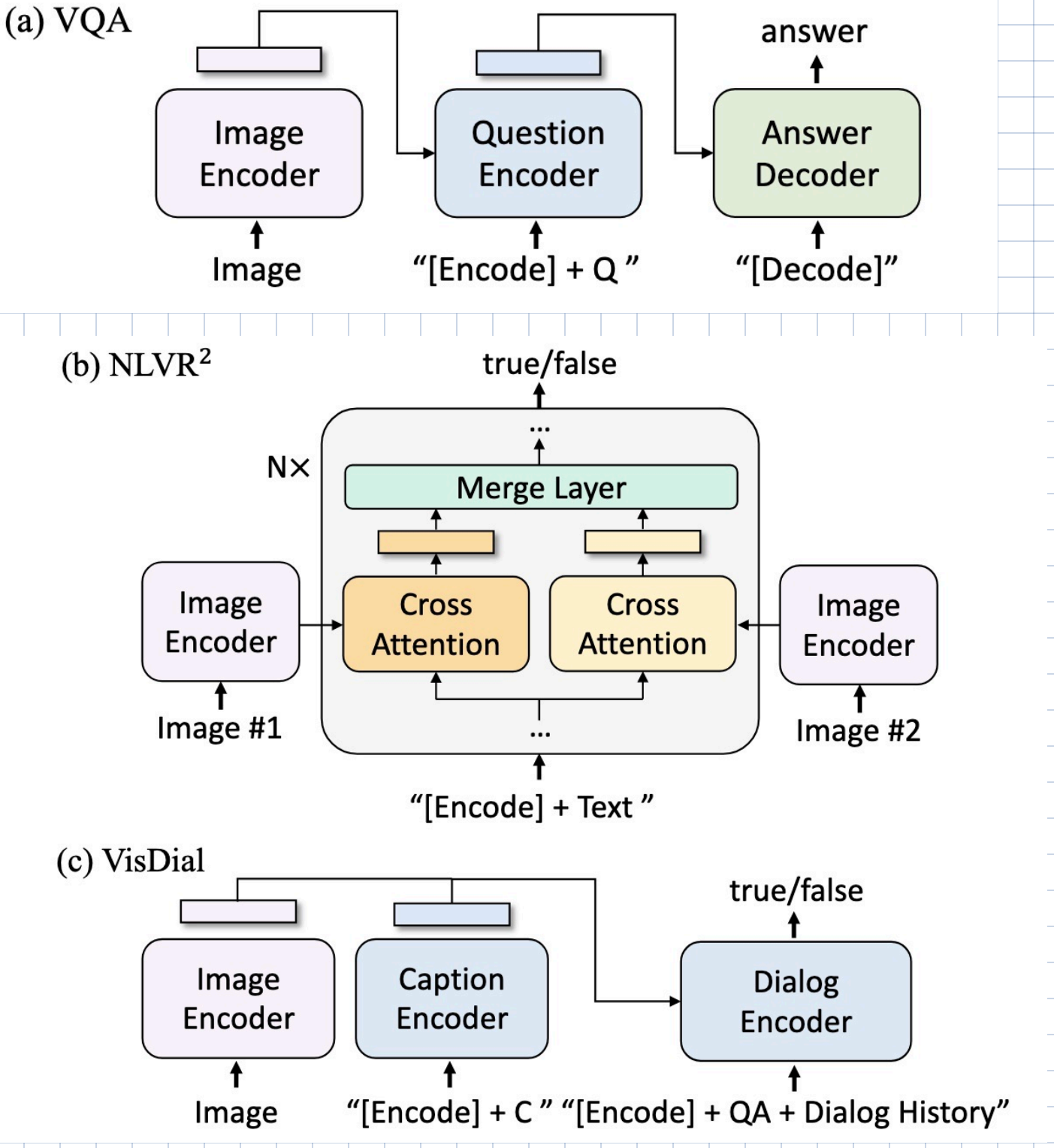
T_s: "a potted plant sitting on top of a pile of rocks"

T_w: "the current castle was built in 1180, replacing a 9th century wooden castle"

T_s: "a large building with a lot of windows on it"

Take components & Join to perform some task

Visual QnA:


(a) VQA
answer
Image Encoder
Question Encoder
Answer Decoder
Image
"[Encode] + Q "
"[Decode]"
(b) NLVR²
true/false
...
N×
Merge Layer
Image Encoder
Cross Attention
Cross Attention
Image Encoder
Image #1
...
Image #2
"[Encode] + Text "
(c) VisDial
true/false
Image Encoder
Caption Encoder
Dialog Encoder
Image
"[Encode] + C "
"[Encode] + QA + Dialog History"

flamingo:

Vision Encoder:

- Trained like CLIP
- Norm free ResNet as Vision encoder
- Bert as Text en-coder (discarded after training)

•Pre-training:

Trained on a combination of two internal (image, text) datasets:

ALIGN (1.8 billion) - noisy | **LTIP (312 million) - cleaner, longer descriptions**

The manner of combination is important for performance

(Ablation study) small NFNet-F0 with BERT-mini for different regimes:

Dataset	Combination strategy	ImageNet accuracy top-1	COCO					
			image-to-text			text-to-image		
			R@1	R@5	R@10	R@1	R@5	R@10
LTIP	None	40.8	38.6	66.4	76.4	31.1	57.4	68.4
ALIGN	None	35.2	32.2	58.9	70.6	23.7	47.7	59.4
LTIP + ALIGN	Accumulation	**45.6**	**42.3**	**68.3**	**78.4**	**31.5**	**58.3**	**69.0**
LTIP + ALIGN	Data merged	38.6	36.9	65.8	76.5	15.2	40.8	55.7
LTIP + ALIGN	Round-robin	41.2	40.1	66.7	77.6	29.2	55.1	66.6

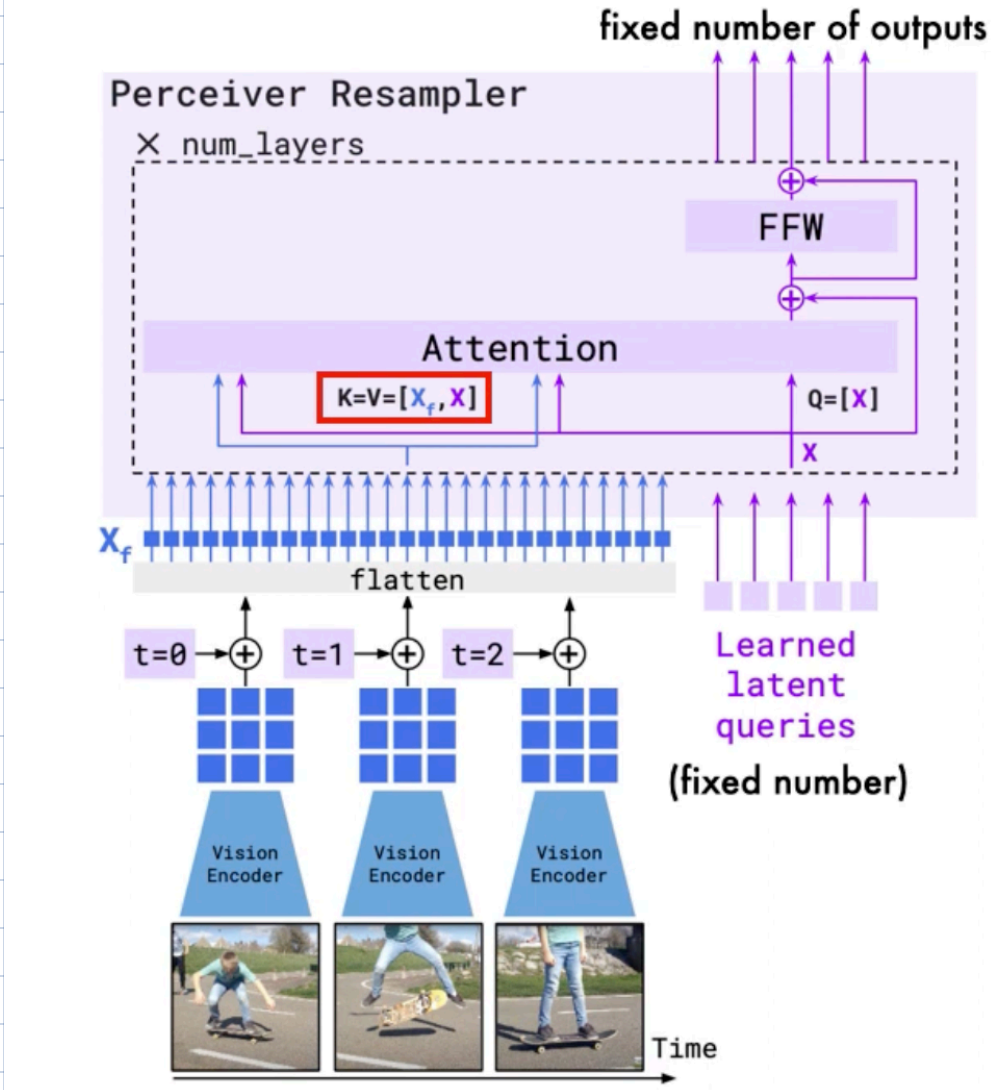
Accumulation: compute gradient on batch from each dataset, combine via weighted sum

Data merged: merge examples from each dataset into each batch

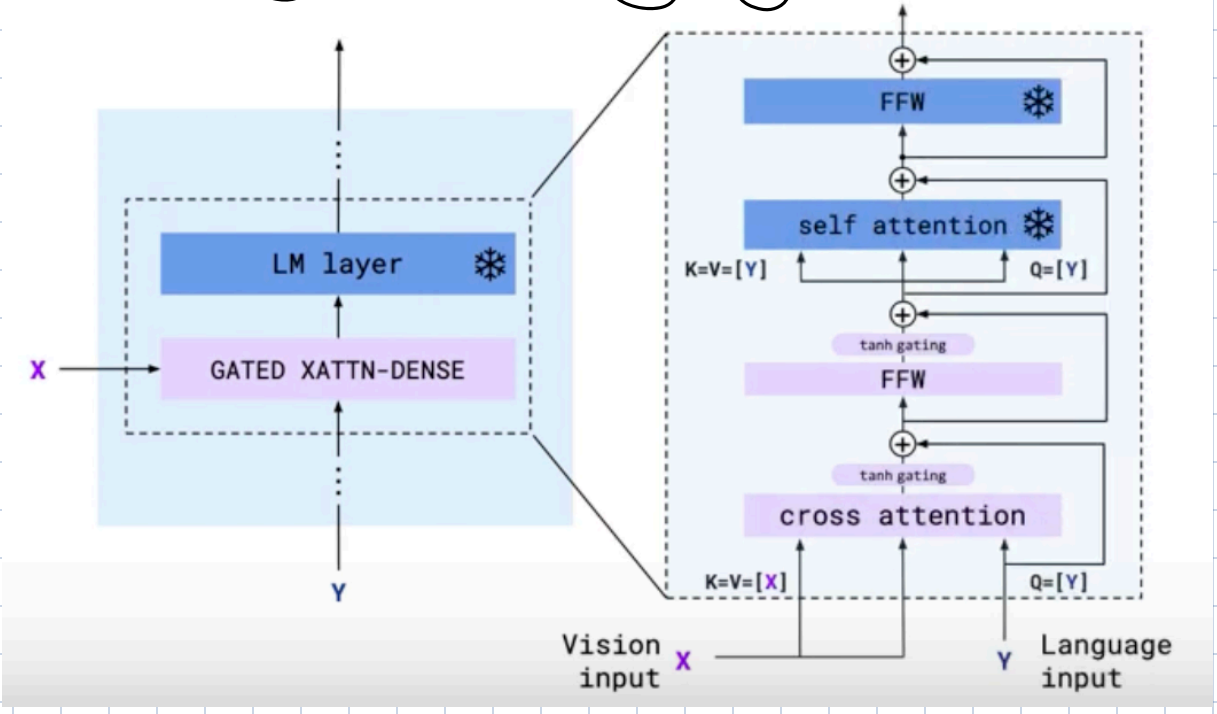
Round-robin: alternate batches from each dataset, update parameters each batch

Perceiver Re-sampler:

- Outputs fixed number of visual tokens (64)
- Temporal (time) encodings are added (Spatial grid position encoding not used as they did not help).



Conditioning the Language model



The Problem:
You have a language model that's really good at text. You want to add vision (images) to it, but you don't want to break what it already learned.

The Solution - Gating:
Think of it like a volume knob between the vision part and the text part of the model.

Why Tanh?
Remove vision signal
- Tanh gives you numbers between -1 and +1 → Add vision signal
- At the start of training, you set it to output close to 0 → Ignore it
- When the output is 0, it's like the volume knob is turned all the way down
- The vision part is essentially "muted" - so the model acts exactly like the original text-only model

→ Each gate has α that is set to 0 but is learnable

How it Preserves Behavior:
- **Day 1:** Vision volume = 0, model acts like original
- During training: Volume slowly increases as the model learns when vision is helpful
- The model gradually learns to "unmute" vision only when it needs it

It's like adding a new instrument to a band but starting with its volume at zero, then slowly turning it up as the musicians learn to play together.

Attention Masking:

Visual Index is just a numbering system for the images/videos in your sequence.

Example:

Input Sequence:

> "I saw a beautiful [IMAGE1] yesterday. The sunset [IMAGE2] was amazing. Later I watched [VIDEO1] about nature."

Visual Index Assignment:

- IMAGE1 gets visual index = 1
- IMAGE2 gets visual index = 2
- VIDEO1 gets visual index = 3

So we have:

- x_1 = IMAGE1 (first visual input)
- x_2 = IMAGE2 (second visual input)

- x_3 = VIDEO1 (third visual input)

Now the Function Φ in Action:

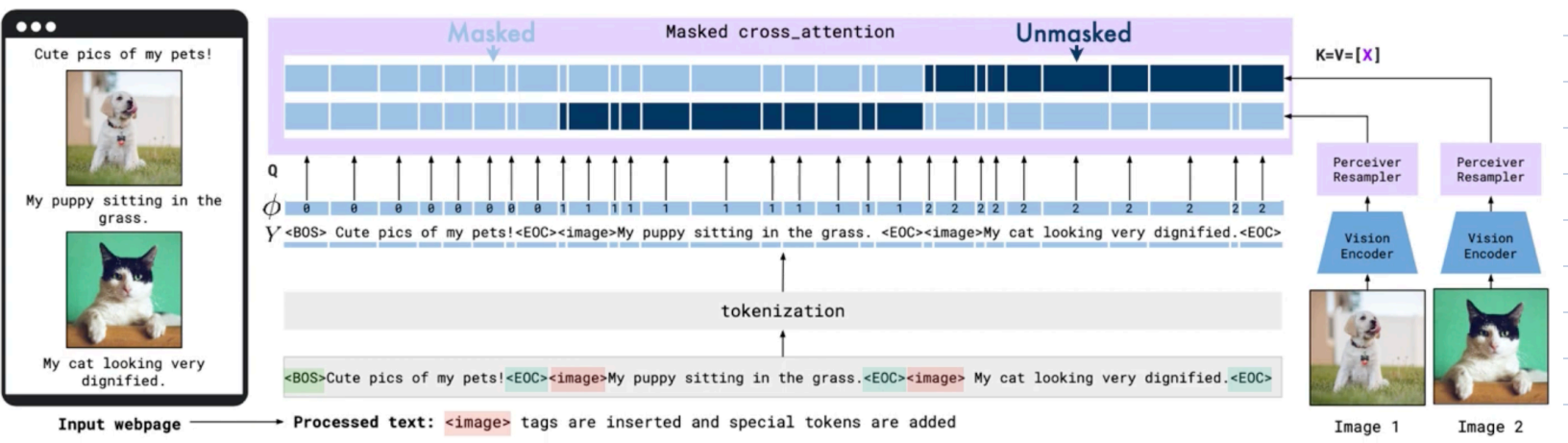
Text Position: 1 2 3 4 5 6 7 8 9 10 11 12
Tokens: "I saw a beautiful [IMG1] yesterday. The sunset [IMG2] was amazing..."
Φ(position): 0 0 0 0 1 1 1 1 2 2 2

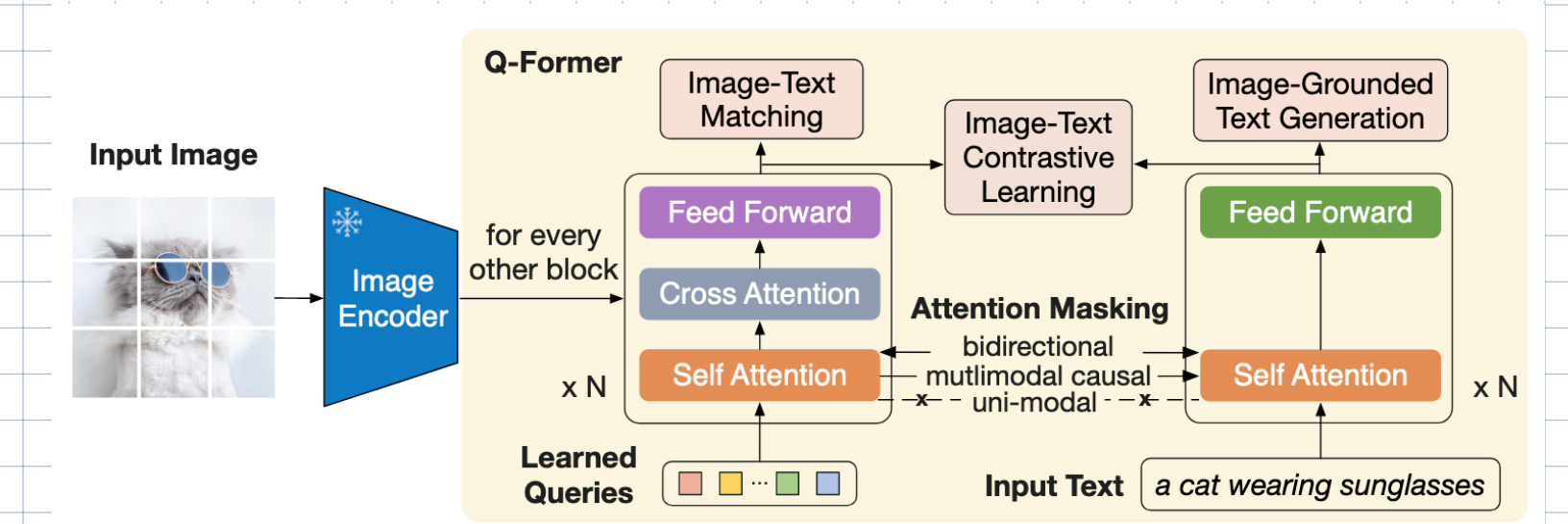
What Φ tells us:

- Position 1-4: Φ = 0 (no images seen yet)
- Position 5-8: Φ = 1 (can use x_1, the first image)
- Position 9-12: Φ = 2 (can use x_2, the second image)

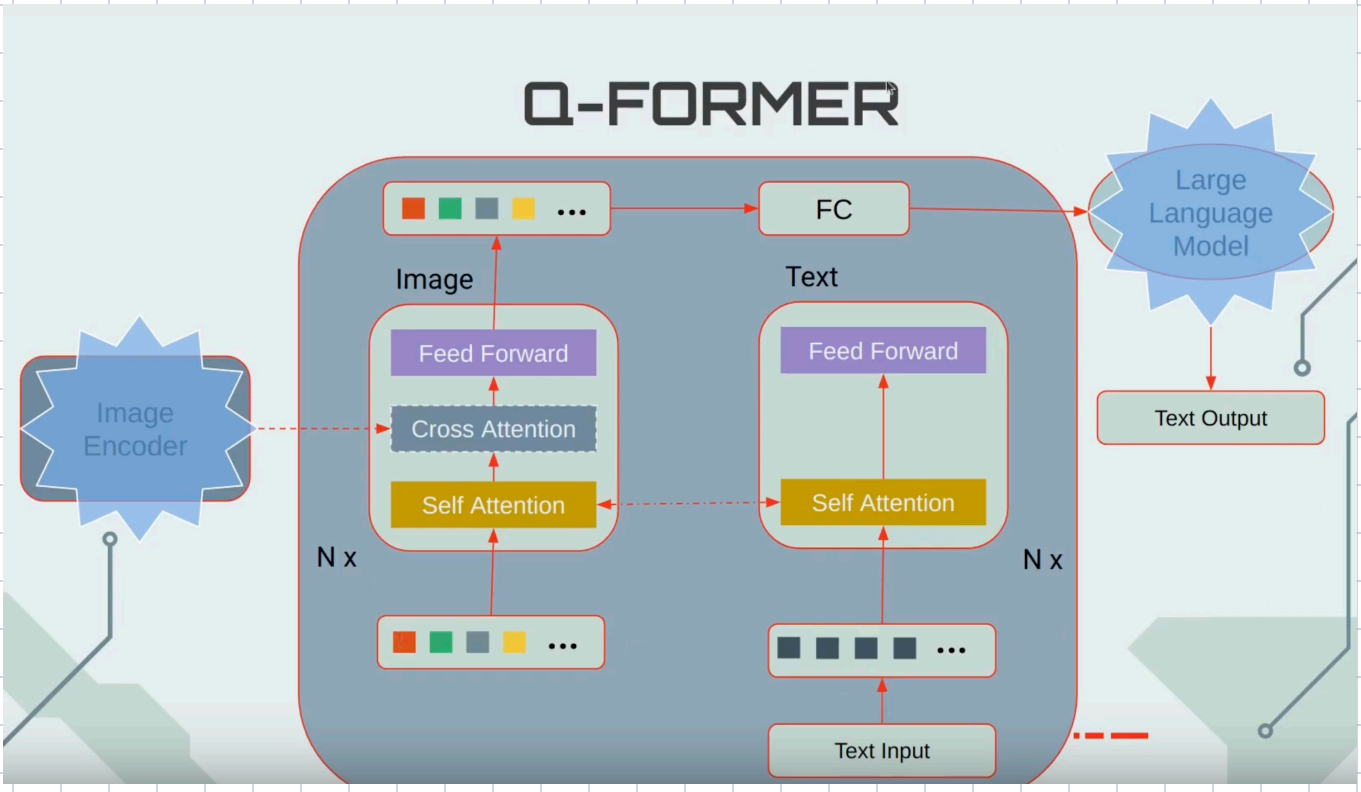
The Rule: When predicting token at position 10 ("was"), the model can use:

- Previous text: "I saw a beautiful [IMG1] yesterday. The sunset [IMG2]"
- Visual inputs: x_2 (because Φ(10) = 2)
- **NOTE:** Model has access to immediately preceding visual token only
- But NOT x_3 (the video) because it hasn't appeared yet

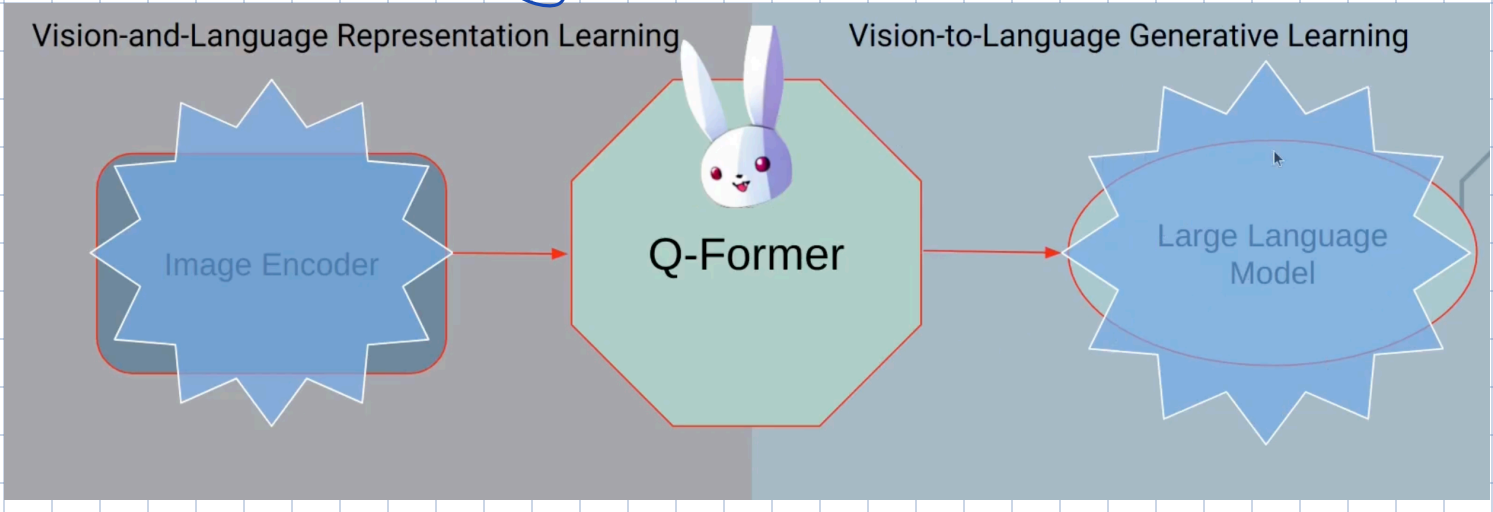




BLIP-2:

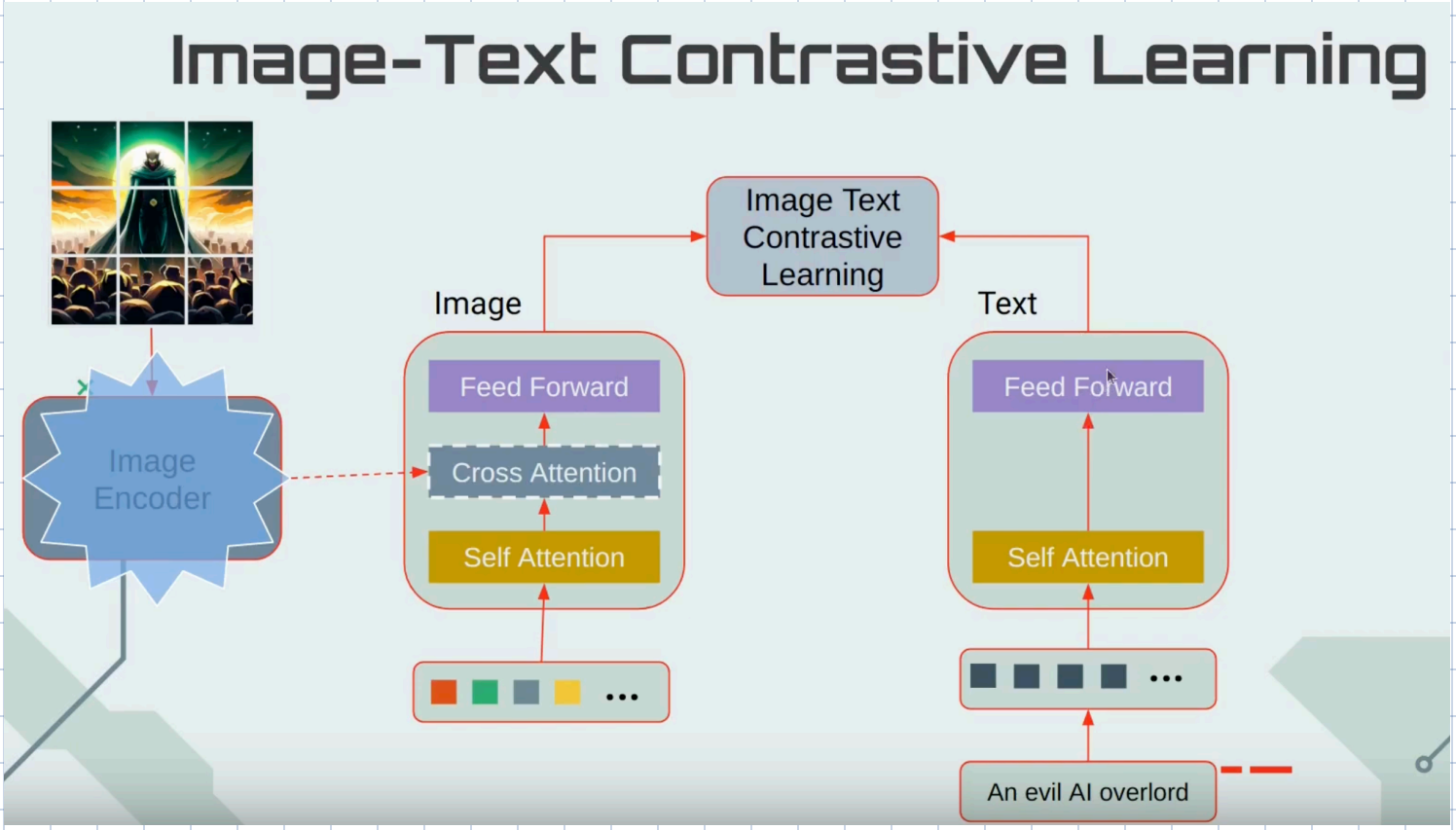


- It has two-stage training:

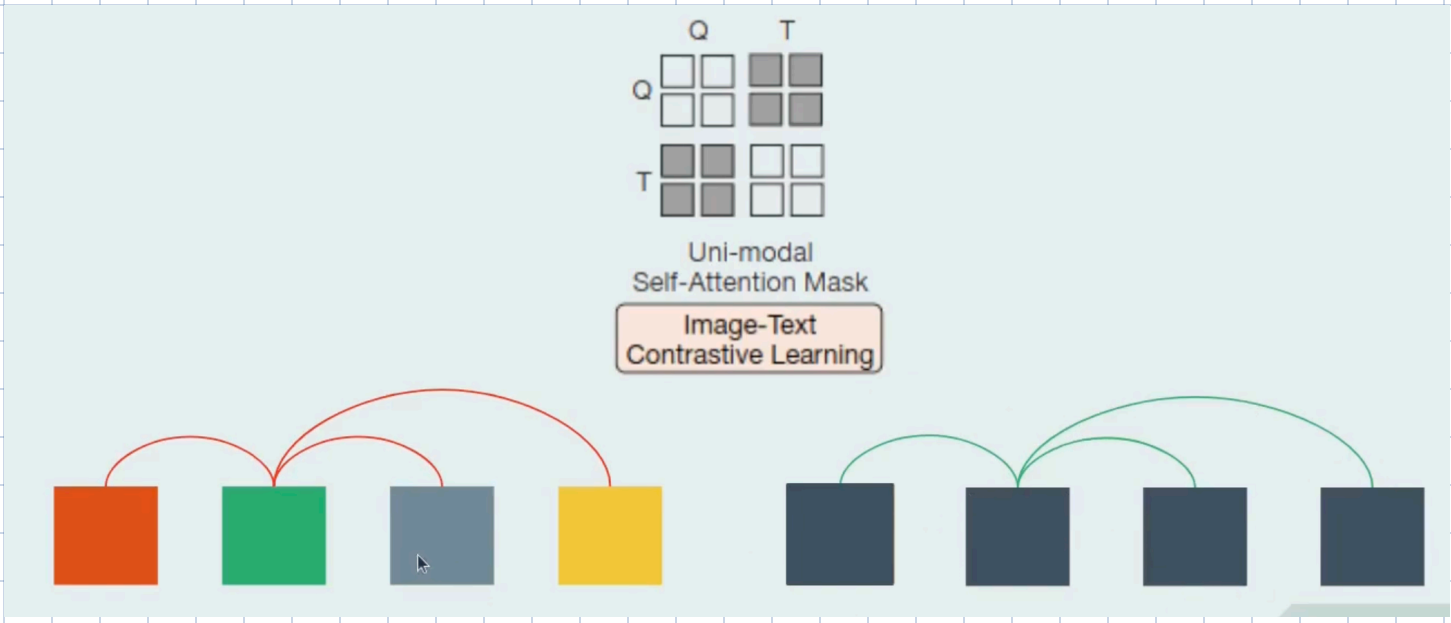


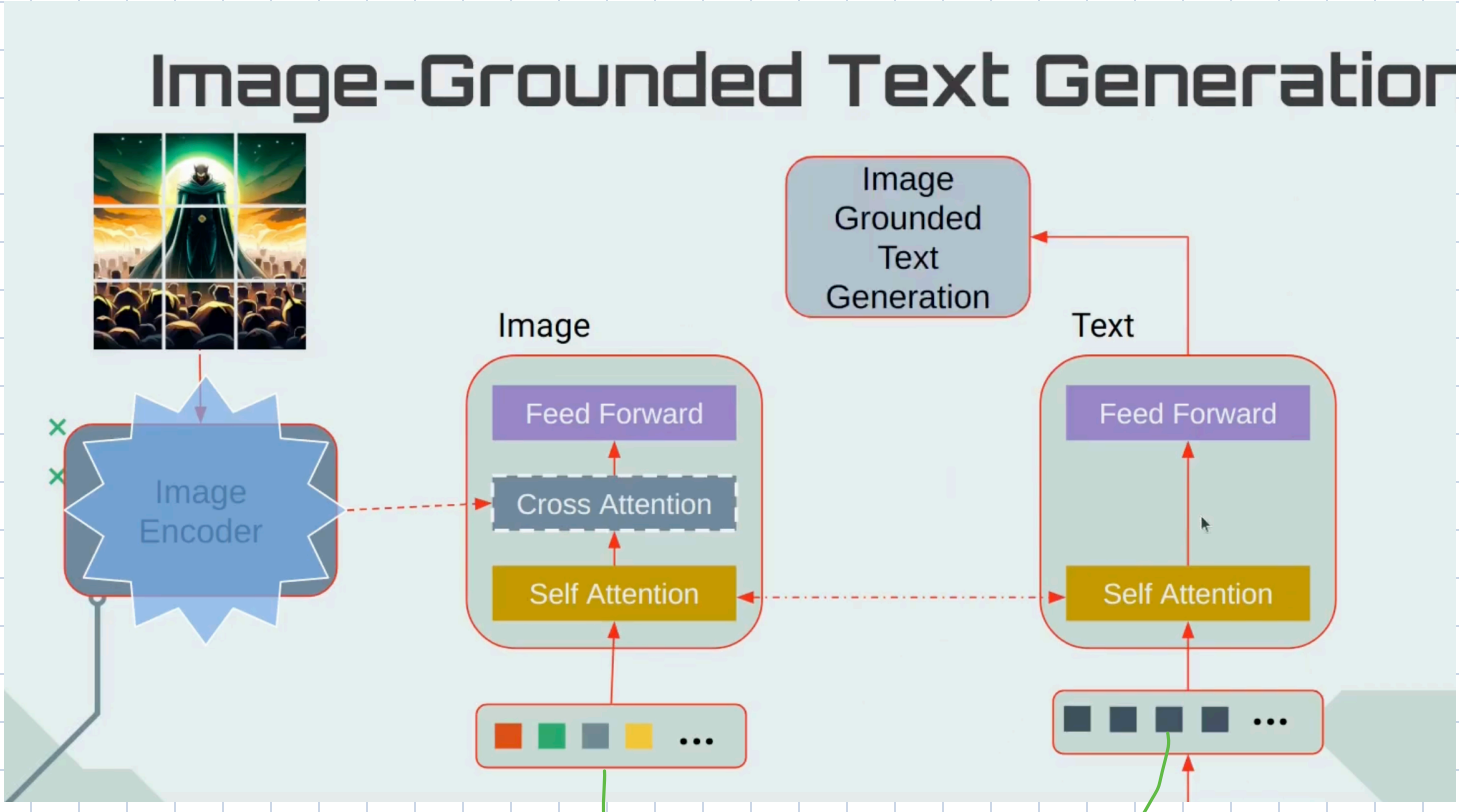
Vision and Language Representation learning

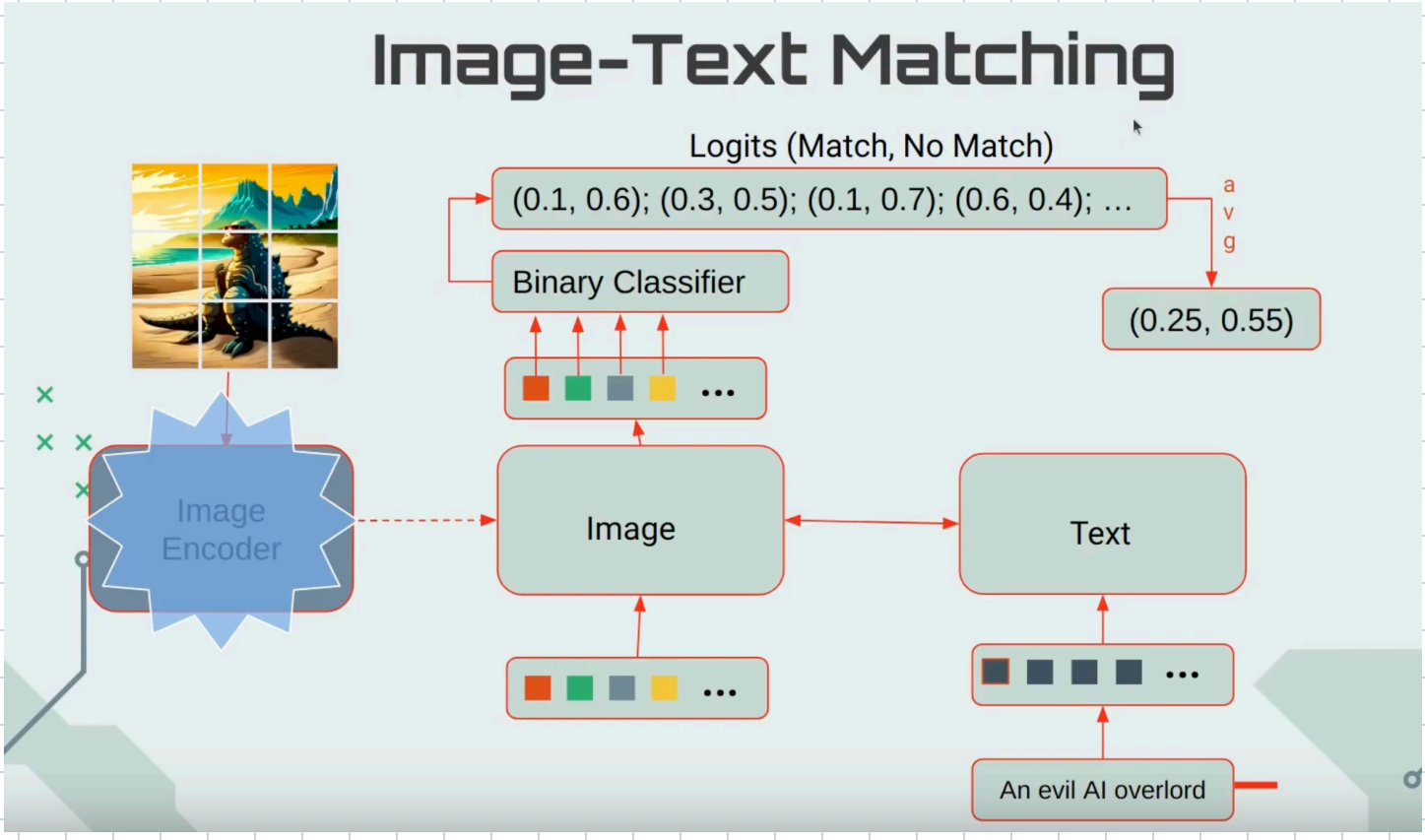
1. Image-text contrastive learning
2. Image Grounded Text Generation
3. Image text matching



This is what CLIP does







uses normal self-attn

Vision-to-Learning Generative Learning: