Flamingo: a Visual Language Model for Few-Shot Learning

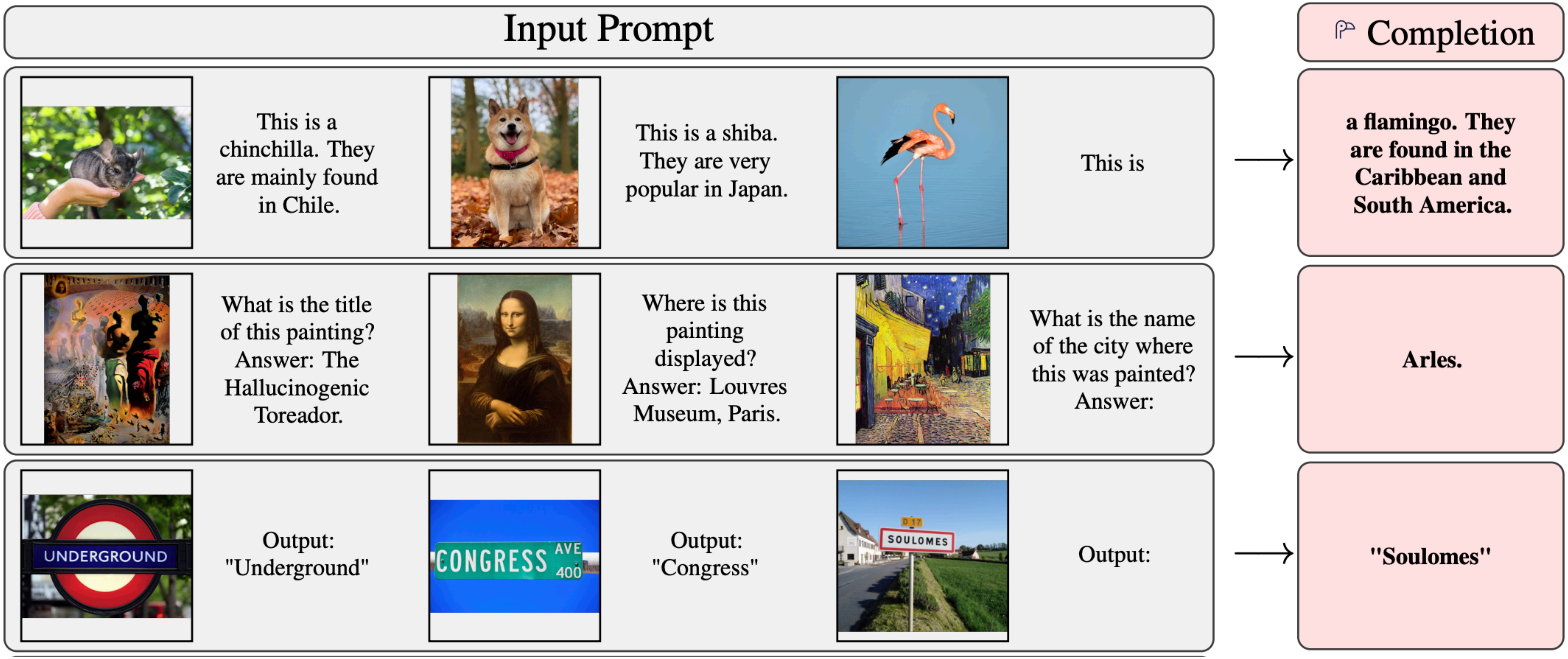
CSE 587 - Presentation

Pouria Mahdavinia, March 27

Task

- Few shot in-context learning for Vision-Language tasks
  - Classification
  - Captioning
  - Visual question answering
  - Visual dialogue

Task



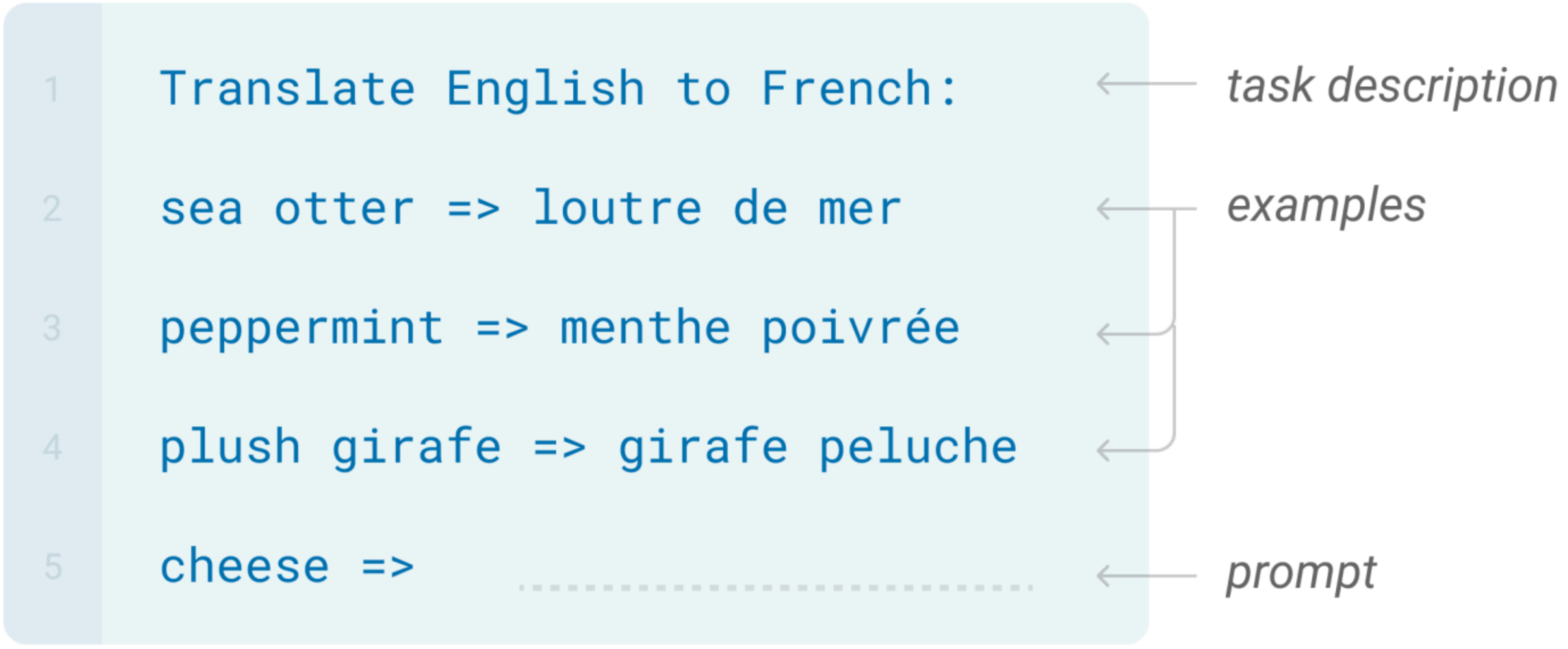
Input Prompt
Completion
This is a chinchilla. They are mainly found in Chile.
This is a shiba. They are very popular in Japan.
This is
a flamingo. They are found in the Caribbean and South America.
What is the title of this painting? Answer: The Hallucinogenic Toreador.
Where is this painting displayed? Answer: Louvres Museum, Paris.
What is the name of the city where this was painted? Answer:
Arles.
UNDERGROUND
Output: "Underground"
CONGRESS AVE 400
Output: "Congress"
D 17
SOULOMES
Output:
"Soulomes"

Motivation

GPT3

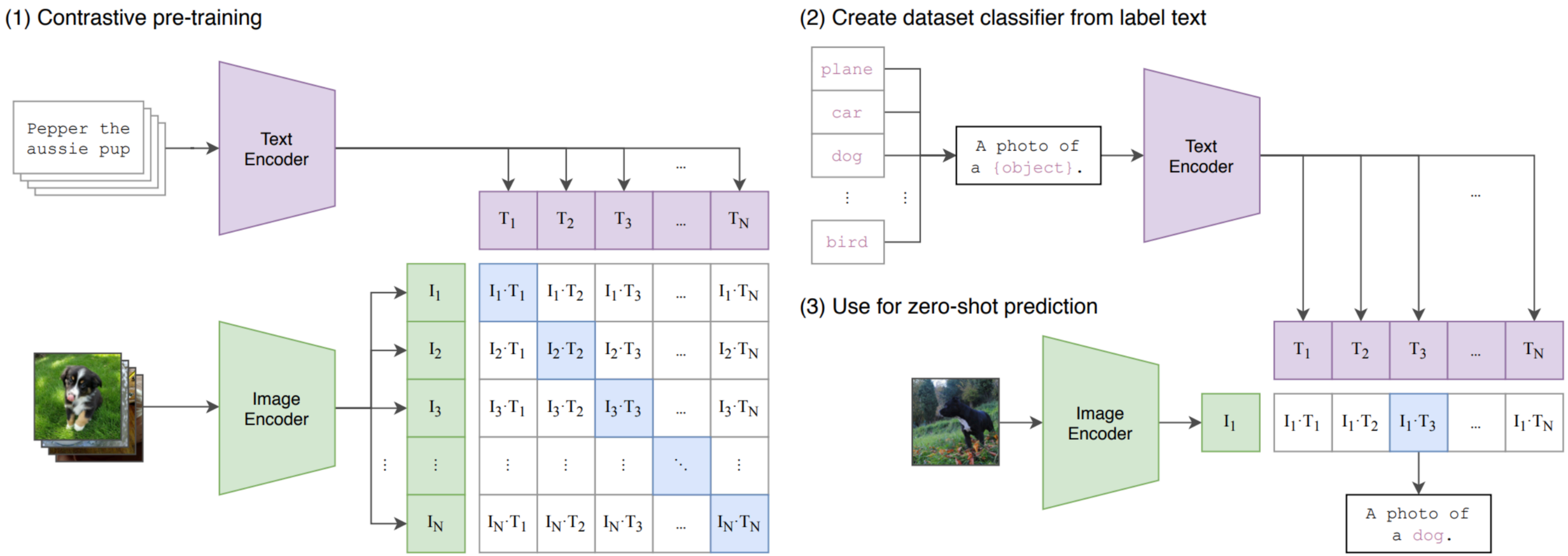
- Large-scale generative LMs -> Good few shot learners

- Only work with text data



Motivation

CLIP



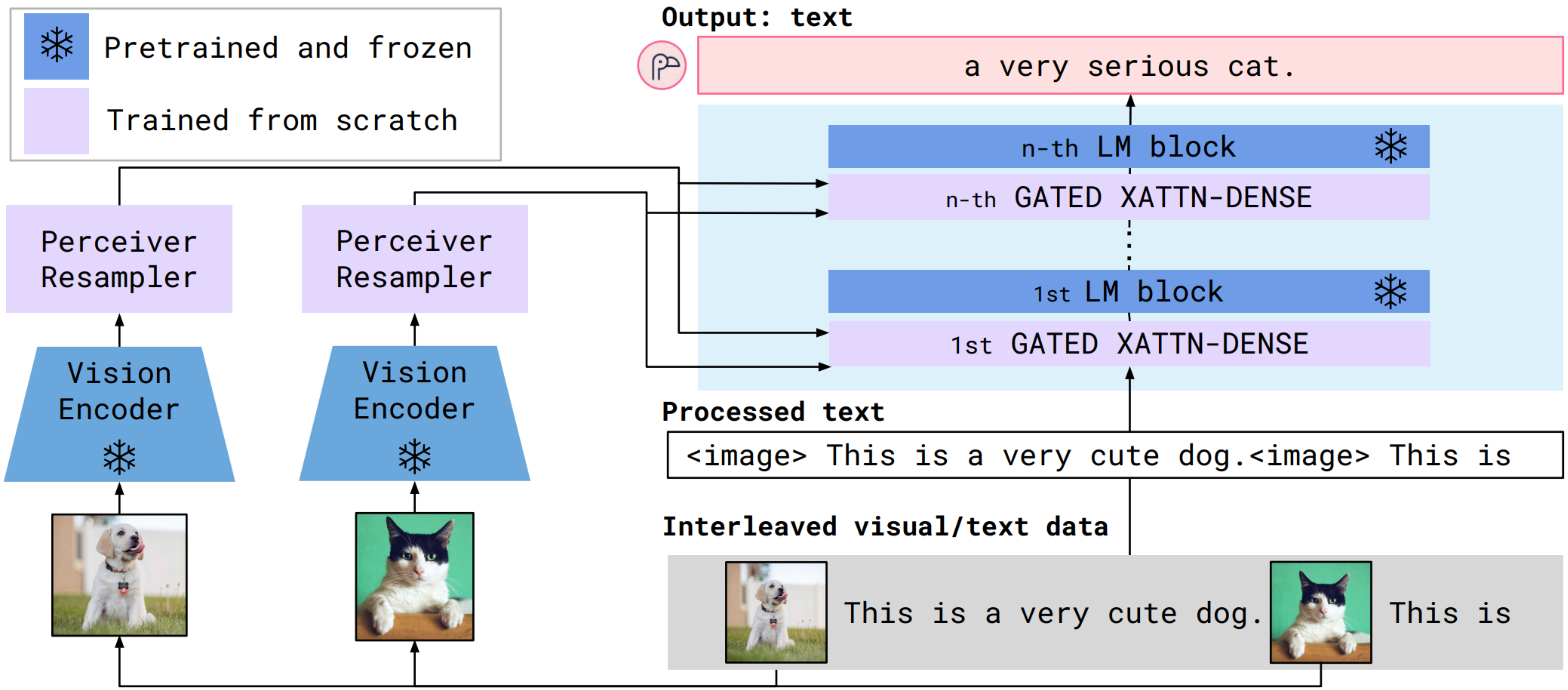
Can not generate text -> Not good for open-ended tasks (captioning or VQA)

Challenges of Visual LMs

1. Combining pre-trained large-scale LMs (trained only on text), with Vision encoders
    - High computation cost of training from scratch
    - Fusing the visual feature to embedded text
2. Accommodating both Image/Video input with arbitrary length in a computationally efficient manner
3. Need for huge dataset
    - The size of image-text pair datasets like CLIP and ALIGN might not be enough for good few-shot learning performance

Approach

Flamingo architecture overview



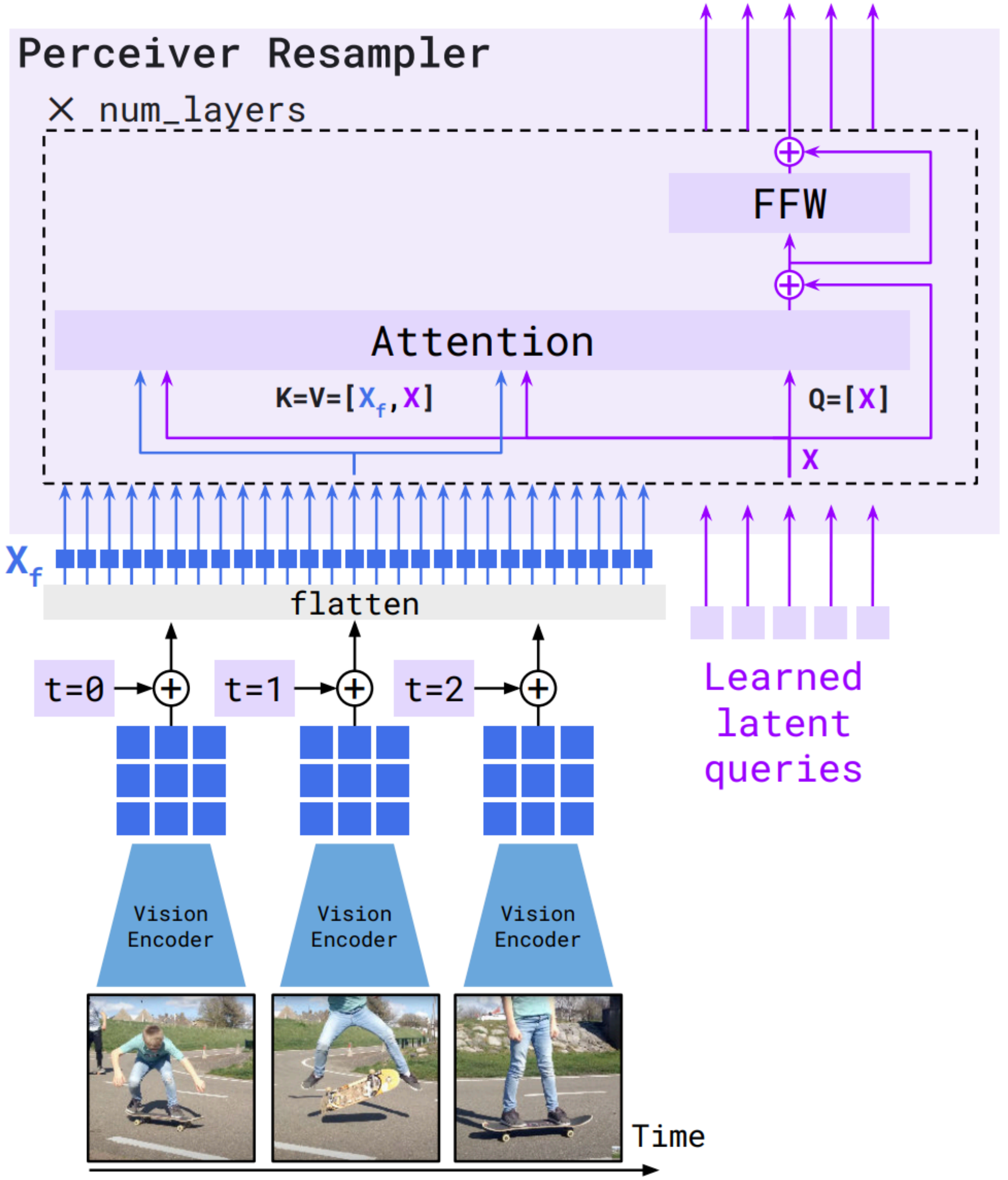
Approach

Vision encoder

- F6 Normalizer-Free ResNet (NFNet) -> Computation efficiency
- Contrastive pre-training similar to CLIP
    - Deployed BERT as text-encoder, and NFNet for vision encoder
- Simplify the CLIP, by using global average pooling instead of global attention pooling
- Trained on ALIGN (1.8 billion image-text pair), and LTIP (312 million image-text pair) using accumulation combination strategy

Approach

Perceiver Resampler: from varying-size large feature maps to few visual tokens



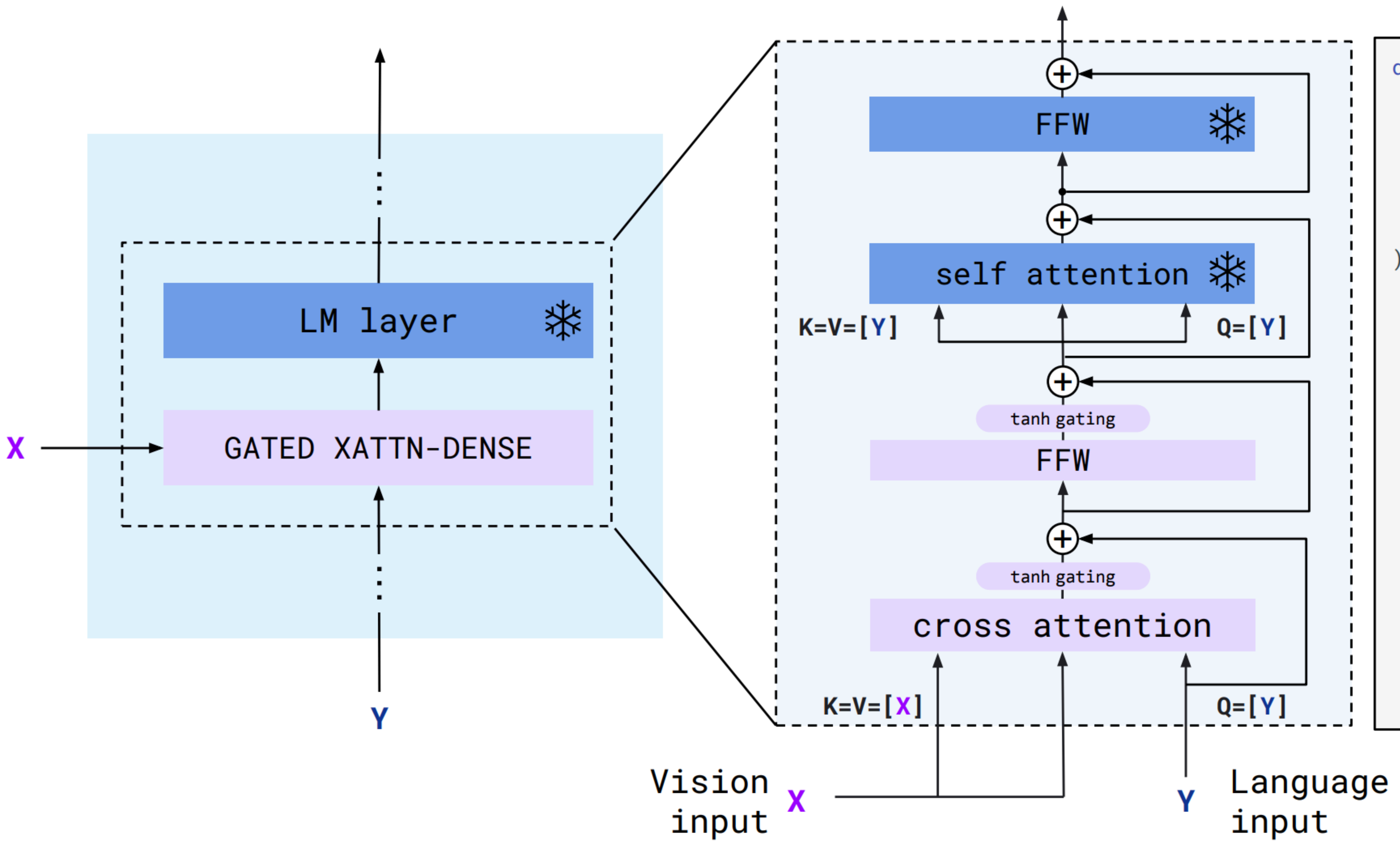

```python
def perceiver_resampler(
    x_f,  # The [T, S, d] visual features (T=time, S=space)
    time_embeddings,  # The [T, 1, d] time pos embeddings.
    x,  # R learned latents of shape [R, d]
    num_layers,  # Number of layers
):
  """The Perceiver Resampler model."""

  # Add the time position embeddings and flatten.
  x_f = x_f + time_embeddings
  x_f = flatten(x_f)  # [T, S, d] -> [T * S, d]
  # Apply the Perceiver Resampler layers.
  for i in range(num_layers):
    # Attention.
    x = x + attention_i(q=x, kv=concat([x_f, x]))
    # Feed forward.
    x = x + ffw_i(x)
  return x
```

Approach

Gated XATTN-Dense layers

Why Gated? Keeping LM intact at initialization -> Improve stability and performance




```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter – init at 0.
    alpha_dense, # ffw gating parameter – init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 1. Gated Cross Attention
  y = y + tanh(alpha_xattn) * attention(q=y, kv=x)
  # 2. Gated Feed Forward (dense) Layer
  y = y + tanh(alpha_dense) * ffw(y)

  # Regular self-attention + FFW on language
  y = y + frozen_attention(q=y, kv=y)
  y = y + frozen_ffw(y)
  return y  # output visually informed language features
```

Approach

Interleaved visual data and text support

Datasets

Image-Text Pairs dataset [N=1, T=1, H, W, C]

Video-Text Pairs dataset [N=1, T>1, H, W, C]

Multi-Modal Massive Web (M3W) dataset [N>1, T=1, H, W, C]

1. MultiModel Massive Web (M3W)
    - Collected from 43 million webpages
    - Extract first five images, and randomly sample 256 Token subsequence
    - 185M images, and 182 GB of text
2. Image/Video-Text pairs data
    - ALIGN (1.8B image-text pairs) + LTIP (312M image-text pairs, better quality) + VTP (27M short videos, around 22 seconds each)

Training strategies

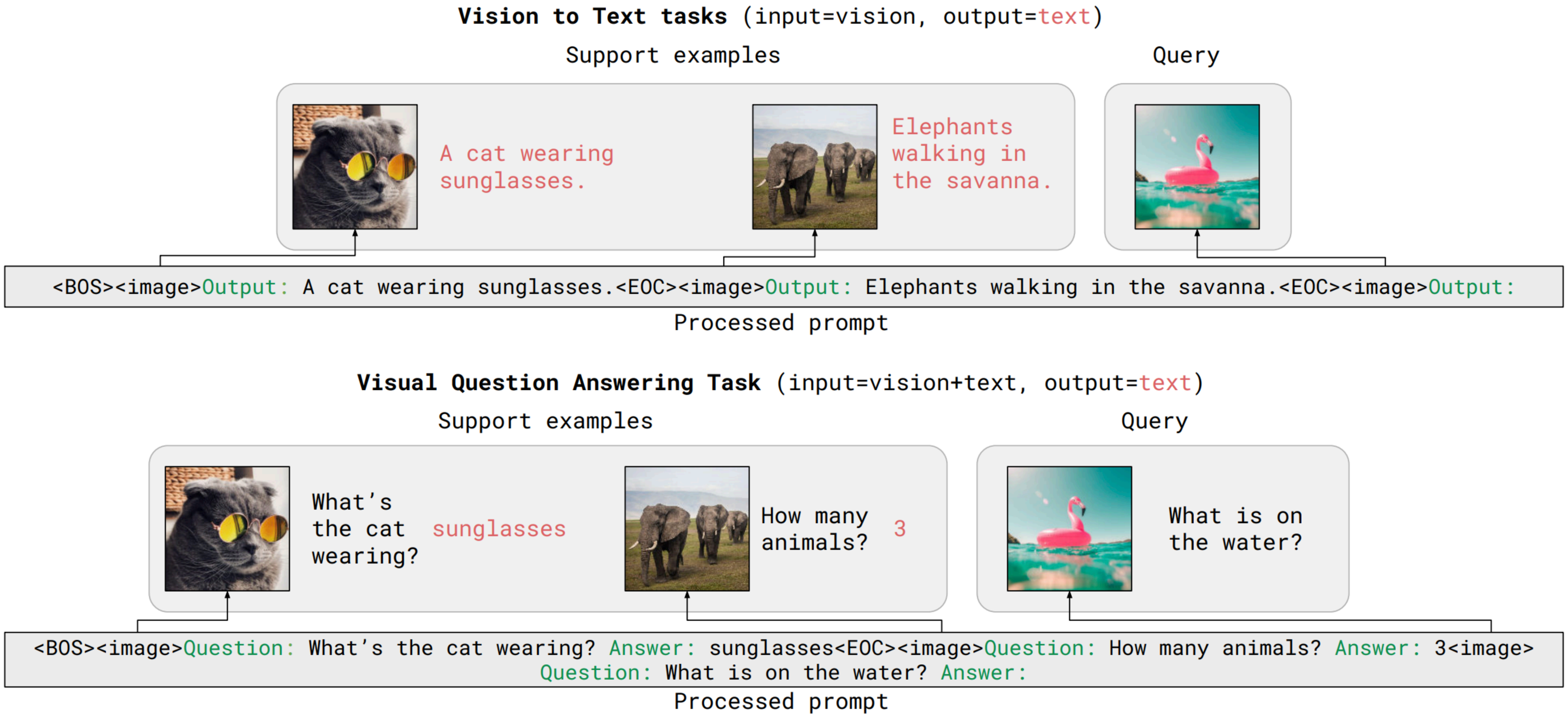
- Training objective: Weighted sum on different datasets minimizing the empirical negative log likelihood

$$\sum_{m=1}^{M} \lambda_m \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}_m} \left[-\sum_{\ell=1}^{L} \log p(y_\ell | y_{<\ell}, x_{\leq\ell}) \right]$$

- Optimizer -> AdamW
- Learning rate schedule: Linear warmup, and then flat LR
- Mixing weights: M3W -> 1 , LTIP -> 0.2 , ALIGN -> 0.2 , VTP -> 0.03

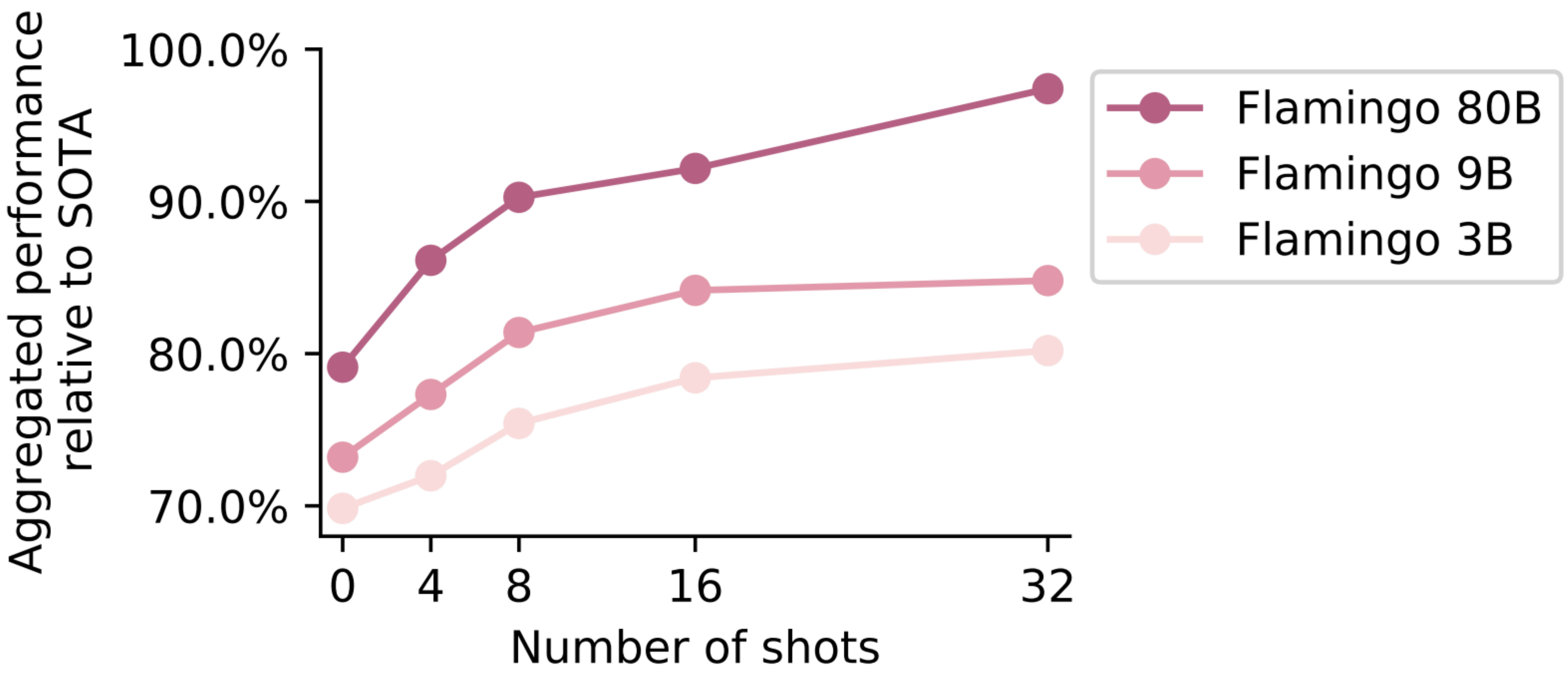
Task adaptation with few-shot in-context learning

Multimodal prompt



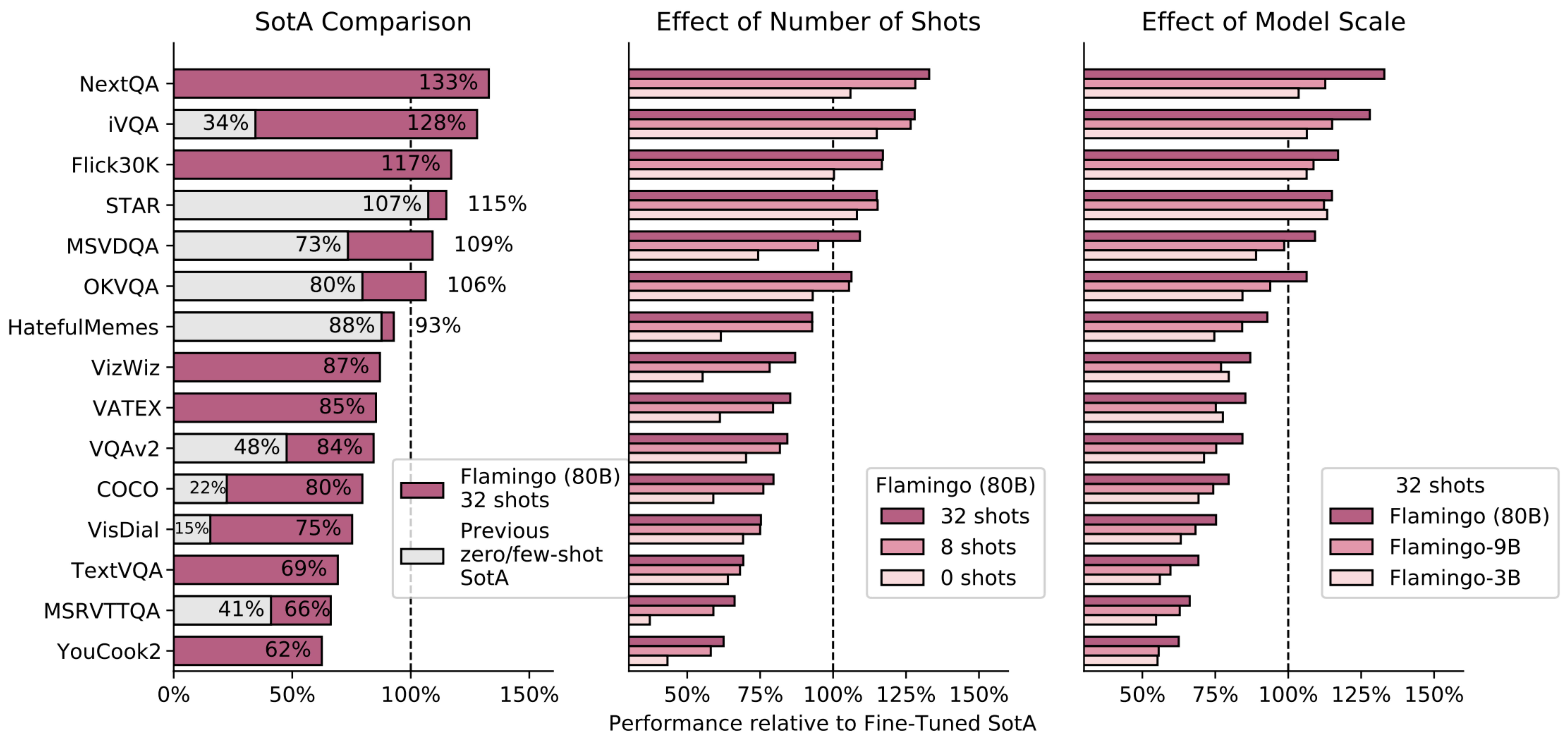
Flamingo models

	Requires model sharding	Frozen Language	Frozen Vision	Trainable GATED XATTN-DENSE	Trainable Resampler	Total count
Flamingo-3B	✗	1.4B	435M	1.2B (every)	194M	**3.2B**
Flamingo-9B	✗	7.1B	435M	1.6B (every 4th)	194M	**9.3B**
Flamingo	✓	70B	435M	10B (every 7th)	194M	**80B**



Evaluation

Overview of Flamingo performance



Ablation studies

	Ablated setting	Flamingo 3B value	Changed value	Param. count ↓	Step time ↓	COCO CIDEr↑	OKVQA top1↑	VQAv2 top1↑	ImageNet top1↑	MSVDQA top1↑	VATEX CIDEr↑	Kinetics top1-top5↑	Overall score↑
	Flamingo 3B model (short training)			3.2B	1.74s	86.5	42.1	55.8	59.9	36.3	53.4	49.4	**68.4**
(i)	Training data	All data	M3W	3.2B	0.68s	58.0	37.2	48.6	35.7	29.5	33.6	34.0	50.7
			w/o VTP	3.2B	1.42s	84.2	43.0	53.9	59.6	34.5	46.0	45.8	65.4
			w/o LTIP/ALIGN	3.2B	0.95s	66.3	39.2	51.6	41.4	32.0	41.6	38.2	56.5
			w/o M3W	3.2B	1.02s	54.1	36.5	52.7	24.9	31.4	23.5	28.3	46.9
(ii)	Optimisation	Grad. accumulation	Round Robin	3.2B	1.68s	76.1	39.8	52.1	50.7	33.2	40.8	39.7	59.7
(iii)	Tanh gating	✓	✗	3.2B	1.74s	78.4	40.5	52.9	54.0	35.9	47.5	46.4	64.0
(iv)	Cross-attention architecture	GATED XATTN-DENSE	VANILLA XATTN	2.4B	1.16s	80.6	41.5	53.4	59.0	32.9	50.7	46.8	65.2
			GRAFTING	3.3B	1.74s	79.2	36.1	50.8	47.5	32.2	47.8	27.9	57.4
(v)	Cross-attention frequency	Every	Single in middle	2.0B	0.87s	71.5	38.1	50.2	44.0	29.1	42.3	28.3	54.6
			Every 4th	2.3B	1.02s	82.3	42.7	55.1	57.1	34.6	50.8	45.5	65.9
			Every 2nd	2.6B	1.24s	83.7	41.0	55.8	59.6	34.5	49.7	47.4	66.2
(vi)	Resampler	Perceiver	MLP	3.2B	1.85s	78.6	42.2	54.7	53.6	35.2	44.7	42.1	63.3
			Transformer	3.2B	1.81s	83.2	41.7	55.6	59.0	31.5	48.3	47.4	65.1
(vii)	Resampler size	Medium	Small	3.1B	1.58s	81.1	40.4	54.1	60.2	36.0	50.2	48.9	66.4
			Large	3.4B	1.87s	84.4	42.2	54.4	60.4	35.1	51.4	49.4	67.3
(viii)	Multi-Img att.	Only last	All previous	3.2B	1.74s	70.0	40.9	52.0	52.3	32.1	46.8	42.0	60.8
(ix)	p_{next}	0.5	0.0	3.2B	1.74s	85.0	41.6	55.2	60.3	36.7	50.6	49.9	67.8
			1.0	3.2B	1.74s	81.3	43.3	55.6	57.8	36.8	52.7	47.8	67.6
(x)	Vision encoder	NFNet-F6	CLIP ViT-L/14	3.1B	1.58s	76.5	41.6	53.4	49.5	33.2	44.5	42.3	61.4
			NFNet-F0	2.9B	1.45s	73.8	40.5	52.8	49.8	31.1	42.9	36.6	58.9
(xi)	LM pretraining	MassiveText	C4	3.2B	1.74s	81.3	34.4	47.1	60.6	30.9	53.9	46.9	62.5
(xii)	Freezing Vision	✓	✗ (random init)	3.2B	4.70s*	74.5	41.6	52.7	45.2	31.4	35.8	32.6	56.6
			✗ (pretrained)	3.2B	4.70s*	83.5	40.6	55.1	55.6	34.6	50.7	41.2	64.5
(xiii)	Freezing LM	✓	✗ (random init)	3.2B	2.42s	74.8	31.5	45.6	59.5	26.9	50.1	43.4	58.2
			✗ (pretrained)	3.2B	2.42s	81.2	33.7	47.4	60.7	31.0	53.9	49.9	62.9
(xiv)	Co-train LM on MassiveText	✗	✓ (random init)	3.2B	5.34s*	69.3	29.9	46.1	59.9	28.1	45.5	46.9	57.4
			✓ (pretrained)	3.2B	5.34s*	83.0	42.5	53.3	60.9	35.1	51.1	50.1	67.2

Limitation and Future work

- Performance gap on classification task comparing to contrastive models such as CLIP, it would be nice future work to bridge this gap (I.e. Calibrate the prompt selection)
- Inheriting the weakness of casual (auto-regressive) pre-trained LM (Replacing it with more expressive bidirectional models)
- Hallucinations and ungrounded guesses in open-ended visual question answering
- Adding additional modalities such as audio for improving the performance