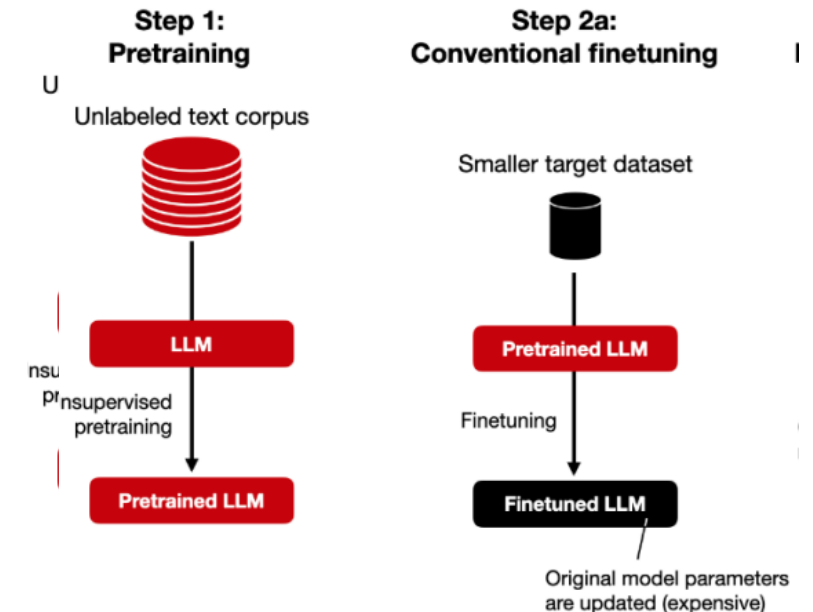


Flamingo: a Visual Language Model for Few-Shot Learning

Alayrac, J. B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., ... & Simonyan, K. (2022). Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35, 23716-23736.

Introduction










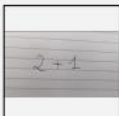
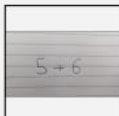
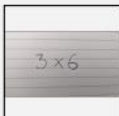
- LLM은 이미 많은 데이터로 학습되어 있기에 새로운 task에 맞춰 fine-tuning 할 때는 많은 데이터가 필요하지 않아도 높은 성능을 낼 수 있음
- 이와 같이 language 정보를 prompt에 담아 vision 분야에도 적은 이미지 instance로도 좋은 성능을 보이기 위한 것이 목적
 - LLM과 같이 대량의 데이터로 학습하여 어떤 prompt에도 높은 성능을 보일 수 있도록 generalized 필요(대량의 multimodal data를 통해 사전학습 진행)
 - Text 뿐 아니라 visual information을 인식할 LM 필요



Introduction

- VLLM(Vision Large Language Model)은 대규모 데이터셋으로 학습을 진행했기에, 원래 학습시킨 task가 아닌 다른 목적의 task로 fine-tuning 진행 시 다수의 annotated dataset(이미지와 이미지에 대한 설명이 쌍으로 있는 데이터셋)이 필요로 함(논문에 의하면 1000개 이상의 대규모 데이터셋)
- Task 별로 하이퍼 파라미터 튜닝 또한 섬세하게 진행해야할 필요가 존재
- Label instance들 간의 similarity score를 계산하는 방식으로 대부분 모델들이 작동하기에 한정된 task에서만 가능

➡ 본 논문은 다양한 task 별 소수의 prompt example로도 높은 few-shot learning 성능을 보이도록 하는 것을 목표로 함

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.
	Output: "Underground"		Output: "Congress"		Output: → "Soulomes"
	2+1=3		5+6=11		→ 3x6=18

Approach

- 이미지들에 대해서 핵심 feature를 추출한 뒤(왼쪽), 이미지 정보와 텍스트 정보를 합쳐(오른쪽) downstream task를 처리하는 구조
- 각각의 unimodal 모델을 freeze하고 Perceiver^[1]와 Cross Attention 를 통해 두 모달리티를 잇는 다리 역할을 하는 레이어만 학습

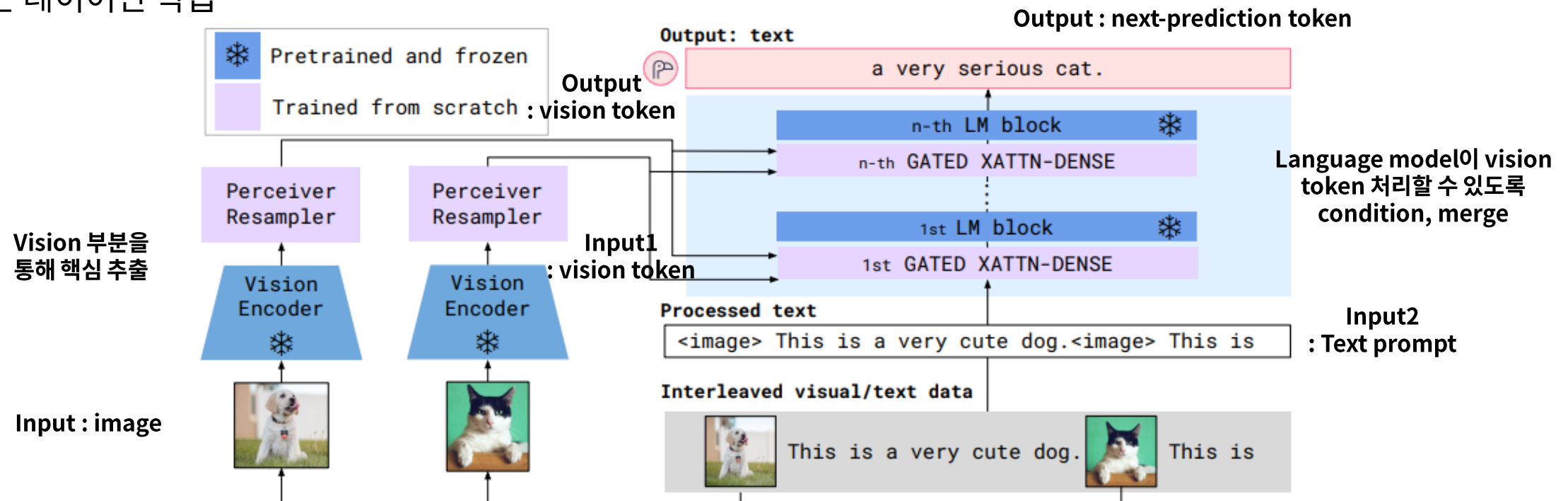
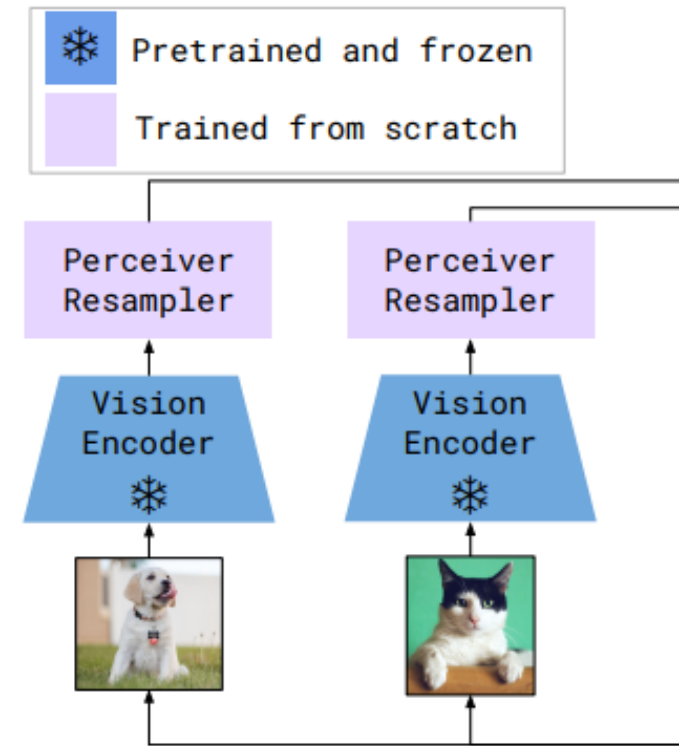


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Approach

Visual processing and the Perceiver Resampler

- Vision Encoder
 - Pretrain NFNet^[2]을 freeze하여 사용
 - Video 혹은 image에 대해 feature vector를 추출
- Perceiver Resampler
 - Vision encoder와 frozen LM을 이어주는 부분(차원 통일)
 - Vision features -> 일정한 개수의 visual token(이미지의 특정 부분의 구조적이고 공간적인 정보) 들로 변환(computational complexity ↓)
 - Vision language resampler module : Transformer에 들어갈 latent input queries를 학습하고 visual features를 cross-attend



Approach

Visual processing and the Perceiver Resampler

- Perceiver Resampler(Transformer 구조)
 - X_f : Vision encoder에서 생성된 grid 형태의 spatio-temporal visual features flatten, key와 value로 사용
 - Learned latent queries: 정해진 개수의 latent input queries를 학습하여 transformer의 query로 사용
 - 원래 query와 똑같은 개수의 visual token을 output으로 리턴
 - 이러한 perceiver resampler의 성능은 plain transformer의 성능을 능가

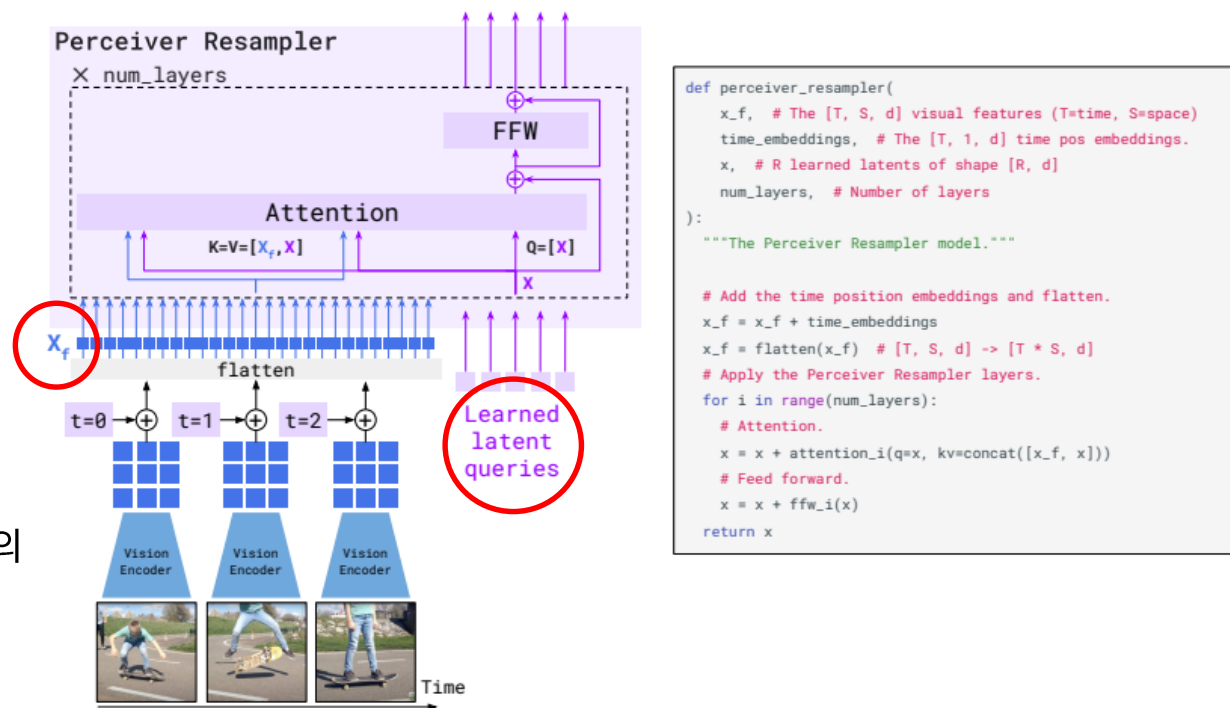
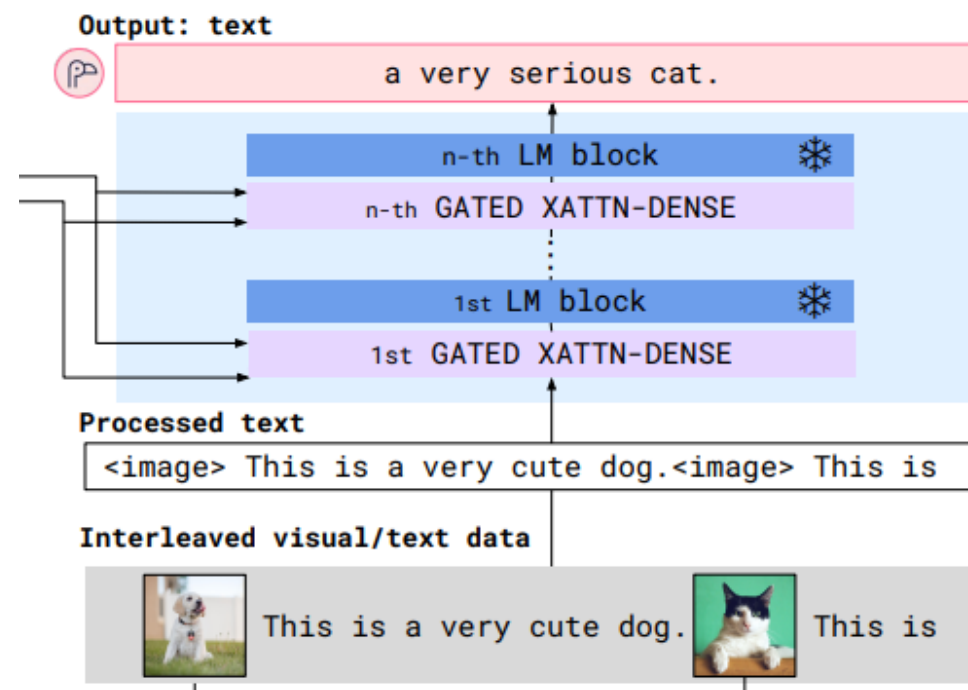


Figure 5: **The Perceiver Resampler** module maps a *variable* size grid of spatio-temporal visual features output by the Vision Encoder to a *fixed* number of output tokens (five in the figure), independently from the input image resolution or the number of input video frames. This transformer has a set of learned latent vectors as queries, and the keys and values are a concatenation of the spatio-temporal visual features with the learned latent vectors.

Approach

Conditioning frozen language models on visual representations

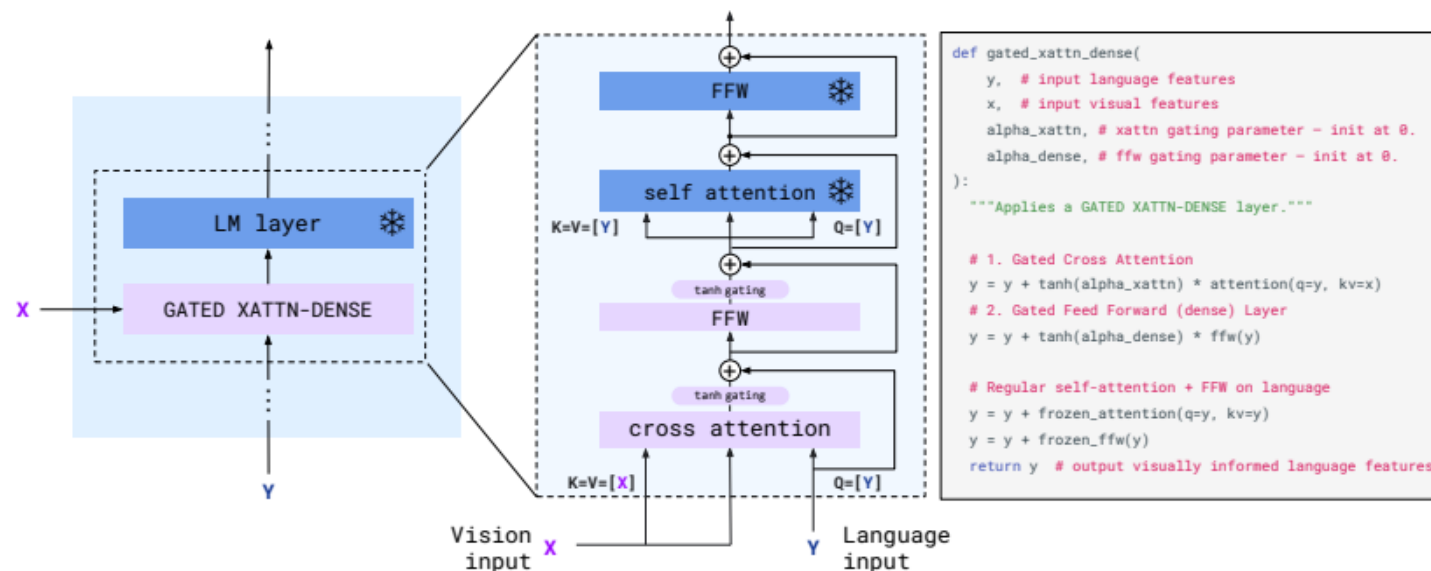
- Language Model
 - transformer decoder는 perceiver resampler로부터 visual representations이 포함된 token을 전달받음 -> text only LM에서 visual 정보도 처리 가능
 - Decoder를 통해 최종 output text를 리턴받음



Approach

Conditioning frozen language models on visual representations

- GATED XATTN-DENSE
 - Scratch부터 학습
 - Key, value : vision feature token / query : language input
 - Cross attention(서로 다른 2개의 sequence 처리) 수행
 - > language input에 대해서 어떤 vision token에 focus를 해야하는지 고르게 되는 역할
 - Tanh gating : conditioned model이 원래 pretrained LM과 호환되도록 하는 역할
 - > LM과 자연스럽게 연결



Approach

Multi-visual input support: per-image/video attention masking

- Transformer^[3] Decoder에서도 번역을 학습하는 과정에서 정답 부분(다음 부분)을 Masking을 적용한 Cross Attention을 수행하듯이, Full text to image cross attention matrix에서 각 Image와 Text는 본인의 매칭 쌍에 해당하는 visual token만 사용하고 나머지는 masking하여 모델링
- Text token이 주어졌을 때(conditioned), 해당 text가 나타나기 직전의 visual token만을 사용
- Image/video x 에 대해 conditioned된 text y 의 likelihood(k 번째 토큰할 때, 처음부터 $k-1$ 번째까지 사용한다)
-> Interleaved text and visual sequence를 처리할 수 있기에, 제안하는 모델은 incontext few-shot learning 가능

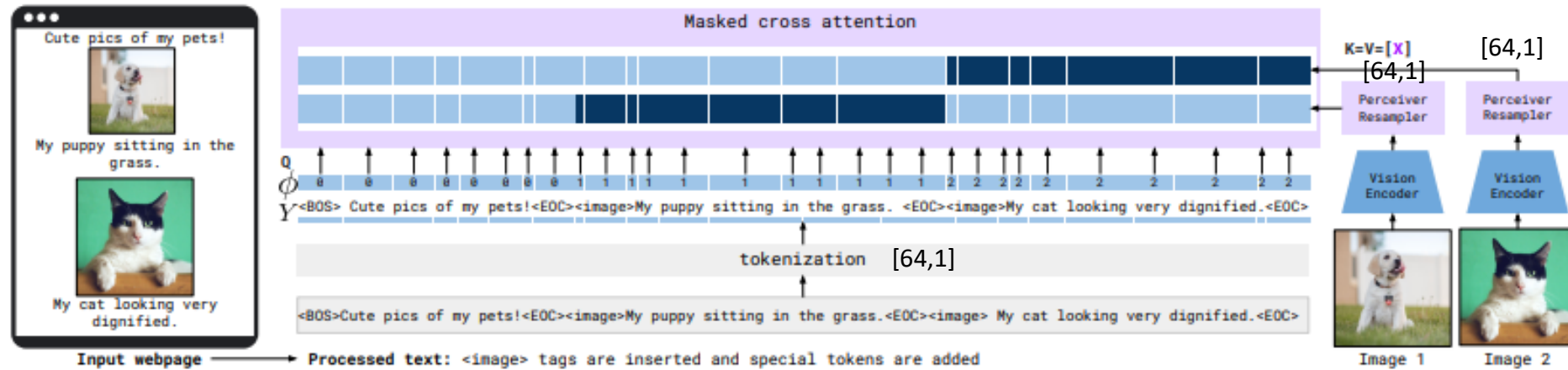
$$\begin{array}{c} \text{텍스트} \\ p(y|x) \\ \text{이미지/} \\ \text{비디오} \end{array} = \prod_{\ell=1}^L \begin{array}{c} \text{시퀀스 내 앞선 텍스트} \\ \text{토큰들의 집합} \\ p(y_{\ell}|y_{<\ell}, x_{\leq \ell}), \\ \text{시퀀스 내 앞선 이미지} \\ \text{토큰들의 집합} \end{array}$$

Approach

Multi-visual input support: per-image/video attention masking

- Cross Attention 수행 시, 이미지와 텍스트가 섞여있는 데이터(html 구조 같은 데이터)를 임베딩 진행
 - 임베딩을 진행하기 위해 문장이 끝나는 부분에는 <BOS>, <EOC> 라는 토큰을, 이미지가 들어가는 부분에는 <image> 토큰 삽입
- 데이터에 들어갈 이미지들을 encoder를 통해 visual token화

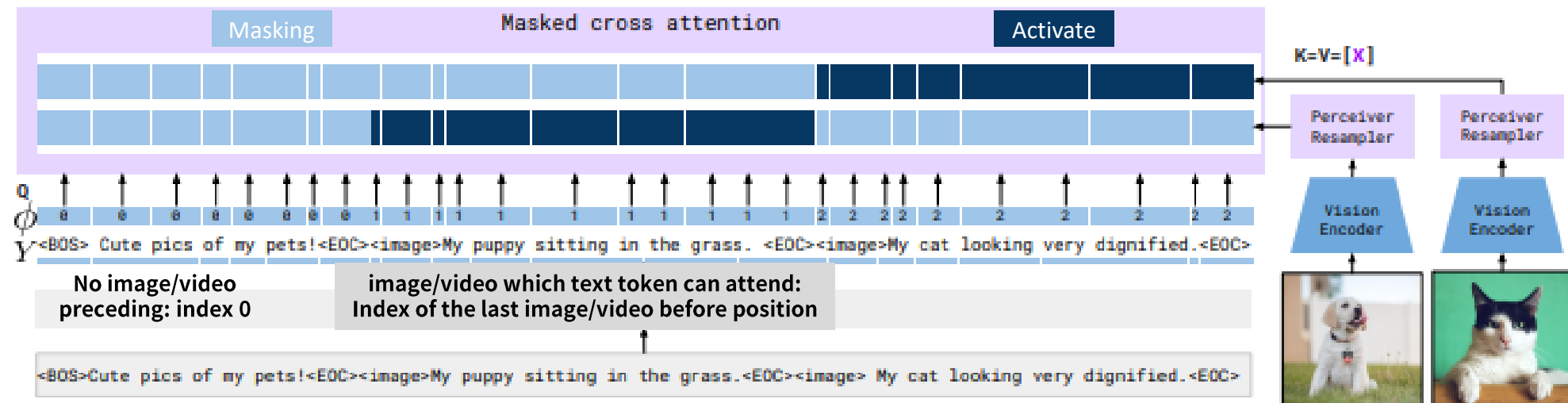
Text interleaved image/videos



Approach

Multi-visual input support: per-image/video attention masking

- Text token의 바로 이전 visual token만(Image와 Text는 본인의 매칭쌍에 해당하는 데이터) cross-attend 하기 위해 나머지 visual token masking
- Single image만 cross attention에 사용하기에 visual input을 쉽게 처리할 수 있음



Approach

Training on a mixture of vision and language datasets

- M3W
 - 43M개의 웹사이트로부터 수집한 interleaved image and text dataset
 - <image> tag와 <EOC> token을 사용하여 image와 text의 위치 표시
- Pairs of image/video and text
 - ALIGN dataset(1.8B image-alt text pairs)에 실험자들이 수집한 더 좋은 퀄리티의 LTIP dataset(312M image-text pairs) 추가
 - Video를 수집한 것은 VTP dataset(27M short videos-sentence description pairs)
 - M3W syntax와 align 하기 위해 각종 tag 추가

-> Flamingo는 거대한 데이터셋으로부터 학습해 기존 Vision Model들과는 차별화되는 성능을 보여줌

Approach

Training on a mixture of vision and language datasets

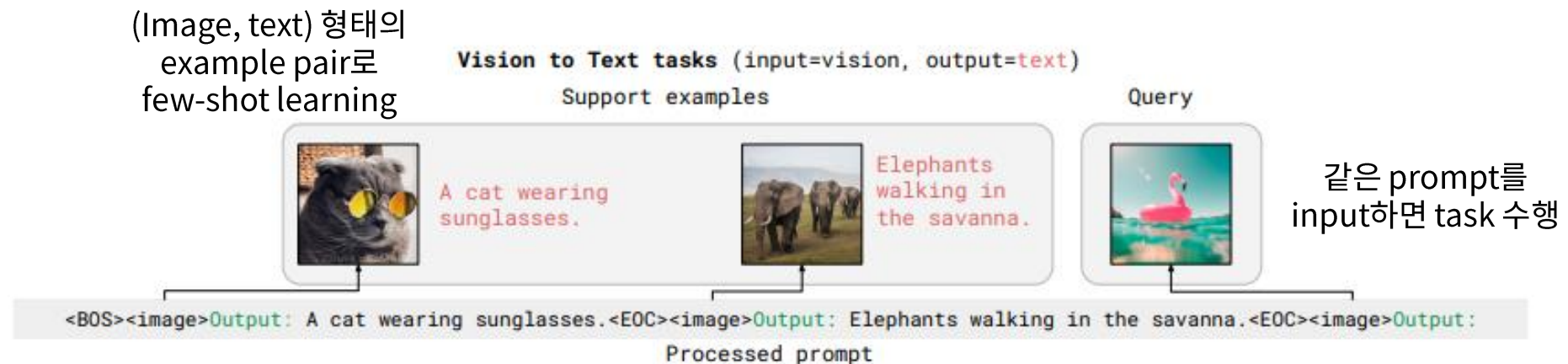
- Multi-objective training and optimization strategy
 - Visual input이 주어질 때, per-dataset expected negative log-likelihoods of text(주어진 텍스트 데이터에 대해 언어 모델이 얼마나 잘 예측하는지를 평균적으로 나타내는 지표)의 weight sum을 최소화
 - 각 dataset에 대한 weight인 λ_m 을 tuning 하는 것이 optimization의 핵심

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

Approach

Task adaptation with few-shot in-context learning

- 한번 학습된 Flamingo는 multimodal prompt에 condition 하기만 하면 새로운 visual task에 사용할 수 있음
- 새로운 task에 빠르게 adapt 하기 위해 in-context learning(주어진 문맥 안에서 다른 예시들을 참고하여 새로운 작업을 수행)을 사용



Experiments

Few Shot Learning

- 실험은 4-shot learning으로 진행
- 16개의 benchmark에서 제안하는 모델의 성능이 이전 zero-shot/few-shot methods를 능가
- 비교군 : multi input을 받을 수 있는 모델들을 few-shot fine-tuning 진행한 방법론들

Method	FT	Shot	OKVQA (I)	VQAav2 (I)	COCO (I)	MSVDQA (V)	VATEX (V)	VizWiz (I)	Flick30K (I)	MSRVTTQA (V)	iVQA (V)	YouCook2 (V)	STAR (V)	VisDial (I)	TextVQA (I)	NextQA (I)	HatefulMemes (I)	RareAct (V)
Zero/Few shot SOTA	X		[34] 43.3 (16)	[114] 38.2 (4)	[124] 32.2 (0)	[58] 35.2 (0)	-	-	-	[58] 19.2 (0)	[135] 12.2 (0)	-	[143] 39.4 (0)	[79] 11.6 (0)	-	-	[85] 66.1 (0)	[85] 40.7 (0)
Flamingo-3B	X	0	41.2	49.2	73.0	27.5	40.1	28.9	60.6	11.0	32.7	55.8	39.6	46.1	30.1	21.3	53.7	58.4
	X	4	43.3	53.2	85.0	33.0	50.0	34.0	72.0	14.9	35.7	64.6	41.3	47.3	32.7	22.4	53.6	-
	X	32	45.9	57.1	99.0	42.6	59.2	45.5	71.2	25.6	37.7	76.7	41.6	47.3	30.6	26.1	56.3	-
Flamingo-9B	X	0	44.7	51.8	79.4	30.2	39.5	28.8	61.5	13.7	35.2	55.0	41.8	48.0	31.8	23.0	57.0	57.9
	X	4	49.3	56.3	93.1	36.2	51.7	34.9	72.6	18.2	37.7	70.8	42.8	50.4	33.6	24.7	62.7	-
	X	32	51.0	60.4	106.3	47.2	57.4	44.0	72.8	29.4	40.7	77.3	41.2	50.4	32.6	28.4	63.5	-
Flamingo	X	0	50.6	56.3	84.3	35.6	46.7	31.6	67.2	17.4	40.7	60.1	39.7	52.0	35.0	26.7	46.4	60.8
	X	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	55.6	36.5	30.8	68.6	-
	X	32	57.8	67.6	113.8	52.3	65.1	49.8	75.4	31.0	45.3	86.8	42.2	55.6	37.9	33.5	70.0	-
Pretrained FT SOTA	✓		54.4 [34] (10K)	80.2 [140] (444K)	143.3 [124] (500K)	47.9 [28] (27K)	76.3 [153] (500K)	57.2 [65] (20K)	67.4 [150] (30K)	46.8 [51] (130K)	35.4 [135] (6K)	138.7 [132] (10K)	36.7 [128] (46K)	75.2 [79] (123K)	54.7 [137] (20K)	25.2 [129] (38K)	79.1 [62] (9K)	-

Table 1: **Comparison to the state of the art.** A *single* Flamingo model reaches the state of the art on a wide array of image (I) and video (V) understanding tasks with few-shot learning, significantly outperforming previous best zero- and few-shot methods with as few as four examples. More importantly, using only 32 examples and without adapting any model weights, Flamingo *outperforms* the current best methods – fine-tuned on thousands of annotated examples – on seven tasks. Best few-shot numbers are in **bold**, best numbers overall are underlined.

Experiments

Few Shot Learning

- 기존 SOTA 모델과 Flamingo 모델의 벤치마크별 Few Shot Learning 성능을 비교한 그래프
- 크기별 Flamingo 모델을 Shot 개수에 따라 성능을 비교한 그래프
 - Few Shot으로 입력 가능한 데이터 개수가 Flexible
 - 샷 개수가 늘어날 수록 성능이 좋아진다는 점
 - 큰 모델일수록 다양한 개수의 샷을 잘 활용한다는 점

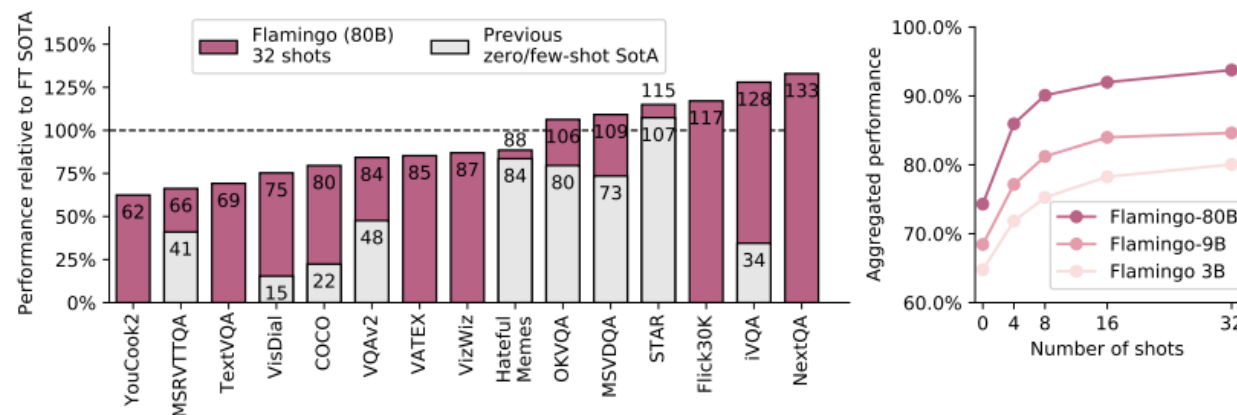


Figure 2: **Flamingo results overview.** *Left:* Our largest model, dubbed *Flamingo*, outperforms state-of-the-art fine-tuned models on 6 of the 16 tasks we consider with no fine-tuning. For the 9 tasks with published few-shot results, *Flamingo* sets the new few-shot state of the art. *Note:* We omit RareAct, our 16th benchmark, as it is a zero-shot benchmark with no available fine-tuned results to compare to. *Right:* Flamingo performance improves with model size and number of shots.

Experiments

Fine tuning

- Fine Tuning 을 진행한 Flamingo 모델과 각 벤치마크별 SOTA 모델들의 성능을 비교
- 이때,
 - Fine Tuning을 진행한 Flamingo 모델은 32 Few Shot 모델보다 거의 항상 성능이 더 좋음
 - 5개의 태스크에서 Flamingo Fine Tuning 모델이 SOTA 성능을 넘었음

Method	VQAV2		COCO	VATEX	VizWiz		MSRVTTQA	VisDial		YouCook2	TextVQA		HatefulMemes
	test-dev	test-std	test	test	test-dev	test-std	test	valid	test-std	valid	valid	test-std	test seen
32 shots	67.6	-	113.8	65.1	49.8	-	31.0	56.8	-	86.8	36.0	-	70.0
Fine-tuned	82.0	82.1	138.1	84.2	65.7	65.4	47.4	61.8	59.7	118.6	57.1	54.1	86.6
SotA	81.3 [†]	81.3 [†]	149.6[†]	81.4 [†]	57.2 [†]	60.6 [†]	46.8	75.2	75.4[†]	138.7	54.7	73.7	84.6 [†]
	[133]	[133]	[119]	[153]	[65]	[65]	[51]	[79]	[123]	[132]	[137]	[84]	[152]

Table 2: **Comparison to SotA when fine-tuning *Flamingo*.** We fine-tune *Flamingo* on all nine tasks where *Flamingo* does not achieve SotA with few-shot learning. *Flamingo* sets a new SotA on five of them, outperforming methods (marked with [†]) that use tricks such as model ensembling or domain-specific metric optimisation (e.g., CIDEr optimisation).

Experiments

Ablation studies

- 모든 모델의 요소가 주요하게 작용

Ablated setting	Flamingo-3B original value	Changed value	Param. count ↓	Step time ↓	COCO CIDEr↑	OKVQA top1↑	VQAv2 top1↑	MSVDQA top1↑	VATEX CIDEr↑	Overall score↑
Flamingo-3B model			3.2B	1.74s	86.5	42.1	55.8	36.3	53.4	70.7
(i)	Training data	All data	w/o Video-Text pairs	3.2B	1.42s	84.2	43.0	53.9	34.5	67.3
			w/o Image-Text pairs	3.2B	0.95s	66.3	39.2	51.6	32.0	60.9
			Image-Text pairs → LAION	3.2B	1.74s	79.5	41.4	53.5	33.9	66.4
			w/o M3W	3.2B	1.02s	54.1	36.5	52.7	31.4	53.4
(ii)	Optimisation	Accumulation	Round Robin	3.2B	1.68s	76.1	39.8	52.1	33.2	62.9
(iii)	Tanh gating	✓	✗	3.2B	1.74s	78.4	40.5	52.9	35.9	66.5
(iv)	Cross-attention architecture	GATED XATTN-DENSE	VANILLA XATTN	2.4B	1.16s	80.6	41.5	53.4	32.9	66.9
			GRAFTING	3.3B	1.74s	79.2	36.1	50.8	32.2	63.1
(v)	Cross-attention frequency	Every	Single in middle	2.0B	0.87s	71.5	38.1	50.2	29.1	59.8
			Every 4th	2.3B	1.02s	82.3	42.7	55.1	34.6	68.8
			Every 2nd	2.6B	1.24s	83.7	41.0	55.8	34.5	68.2
(vi)	Resampler	Perceiver	MLP	3.2B	1.85s	78.6	42.2	54.7	35.2	66.6
			Transformer	3.2B	1.81s	83.2	41.7	55.6	31.5	66.7
(vii)	Vision encoder	NFNet-F6	CLIP ViT-L/14	3.1B	1.58s	76.5	41.6	53.4	33.2	64.9
			NFNet-F0	2.9B	1.45s	73.8	40.5	52.8	31.1	62.7
(viii)	Freezing LM	✓	✗ (random init)	3.2B	2.42s	74.8	31.5	45.6	26.9	57.8
			✗ (pretrained)	3.2B	2.42s	81.2	33.7	47.4	31.0	62.7

Table 3: **Ablation studies.** Each row should be compared to the baseline Flamingo run (top row). Step time measures the time spent to perform gradient updates on all training datasets.

Discussion

Contribution

- 여러 멀티모달 task에서 좋은 성능을 보이는 VLM 모델 Flamingo 개발
- unimodal 모델을 freeze하고 두 모달리티를 잇는 다리 역할을 하는 레이어만 학습하여 계산 리소스를 절약& 각 unimodal 모델이 갖고 있는 능력들을 활용
- Flamingo model 이 few-shot learnin만으로 다양한 task에 보이는 성능을 quantitatively하게 평가
- Few-shot learning model 중 여러 task에 대해 SOTA 달성

Limitations

- 낮은 Classification 성능
- 학습한 데이터에 따라 성능이 많이 저하되는 편향성을 보임

Reference

1. Jaegle, A., Borgeaud, S., Alayrac, J. B., Doersch, C., Ionescu, C., Ding, D., ... & Carreira, J. (2021, October). Perceiver IO: A General Architecture for Structured Inputs & Outputs. In International Conference on Learning Representations.
2. Brock, A., De, S., Smith, S. L., & Simonyan, K. (2021, July). High-performance large-scale image recognition without normalization. In International Conference on Machine Learning (pp. 1059-1071). PMLR.
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.