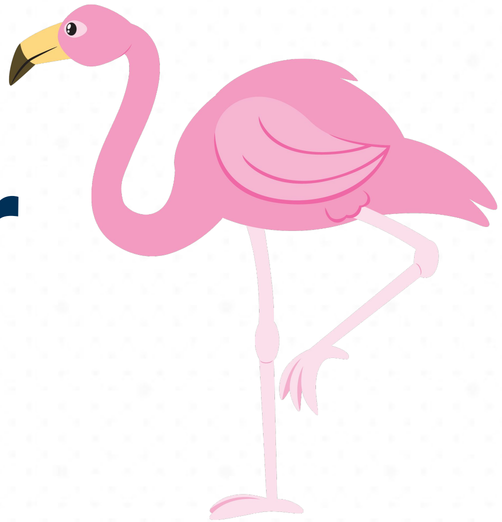
Flamingo: a Visual Language Model for Few-Shot Learning



Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, Karen Simonyan

NeurIPS 2022

Gong Zhang and Pranav Datta

Outline

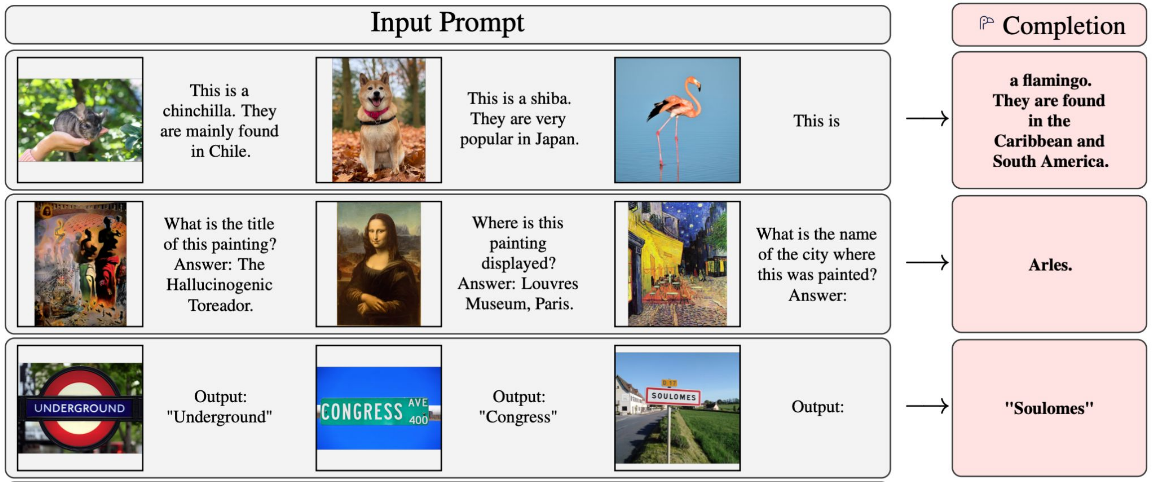
- Problem Statement
- Related Works
- Approach
- Experiments & Results
- Limitations
- Societal Implications
- Strengths
- Weaknesses
- Relationships to Other Papers

Problem Statement

Goal: Few-shot learning to perform novel multimodal tasks

Implications

- Key element of human intelligence
- Don't need to fine-tune models
  - Resource intensive
  - Task-specific annotated data



Contributions

- Flamingo: family of VLMs [1]
  - Connect frozen vision-only and language-only models
  - Interactive, generates open-ended text
- State-of-the-art learning on 16 tasks (Q)
  - Using just examples
  - VQA, captioning, visual dialogue, etc.

Q: Can it localize objects?

Georgia Tech

Related Works

Adapting models to novel tasks

Partial Fine-Tuning

- Adapter modules [2]
  - Few trainable parameters per task
  - Original network parameters stay fixed
- BitFit [3]
  - Only modifies bias term
  - Competitive performance to fine-tuned models

Prompt-Based Approach

- GPT-3 [4]
  - Show in-context examples within prompt
  - Scaled-up language model
- Prompt-Tuning [5] (Q)
  - Prompt optimization through gradient descent
  - Learn “soft prompts” to influence frozen LM to perform tasks

Q: Since prompt-tuning achieved better few-shot learning performance than GPT-3, could it also achieve better performance in multimodal space?

Georgia Tech

Related Works

Chinchilla: Base Language Model [6]

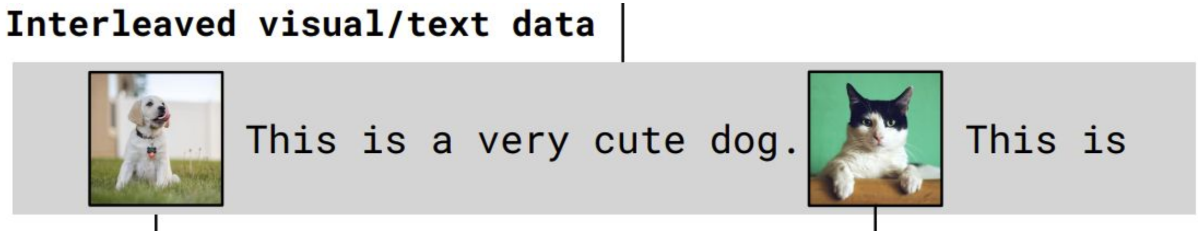
- SOTA accuracy on MMLU
  - MMLU: Exam-like questions on academic subjects
- Scaled training tokens at same rate as model size
- Trained on *MassiveText* [7]

Random	25.0%
Average human rater	34.5%
GPT-3 5-shot	43.9%
Gopher 5-shot	60.0%
***Chinchilla* 5-shot**	**67.6%**
Average human expert performance	*89.8%*

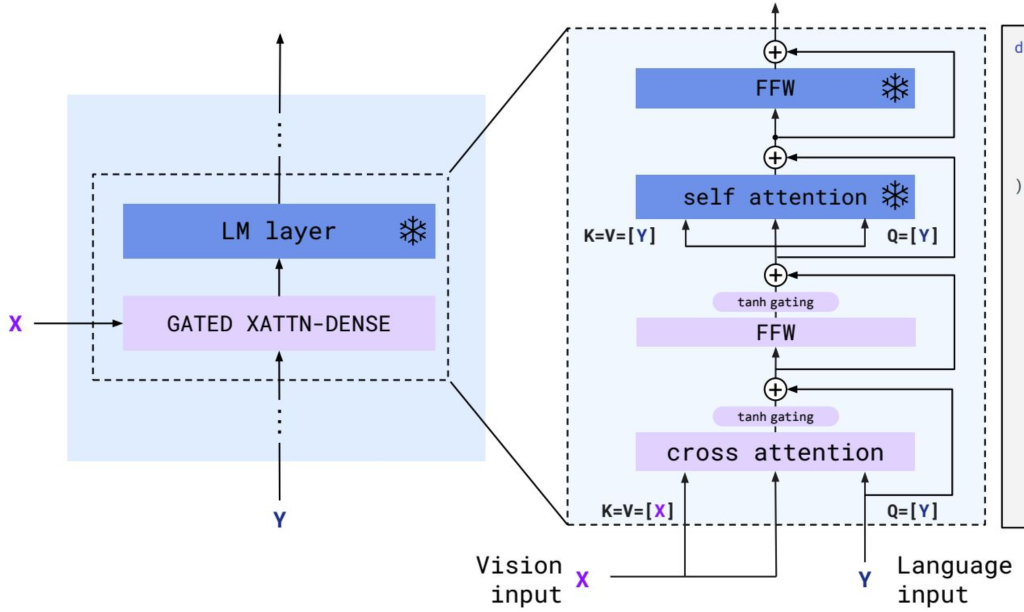
Georgia Tech

Approach

Text input interleaved with image



Visually-conditioned autoregressive text generation




```
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
```

Use of tanh and initialized to zero: to have no effect at training beginning

Georgia Tech

Approach

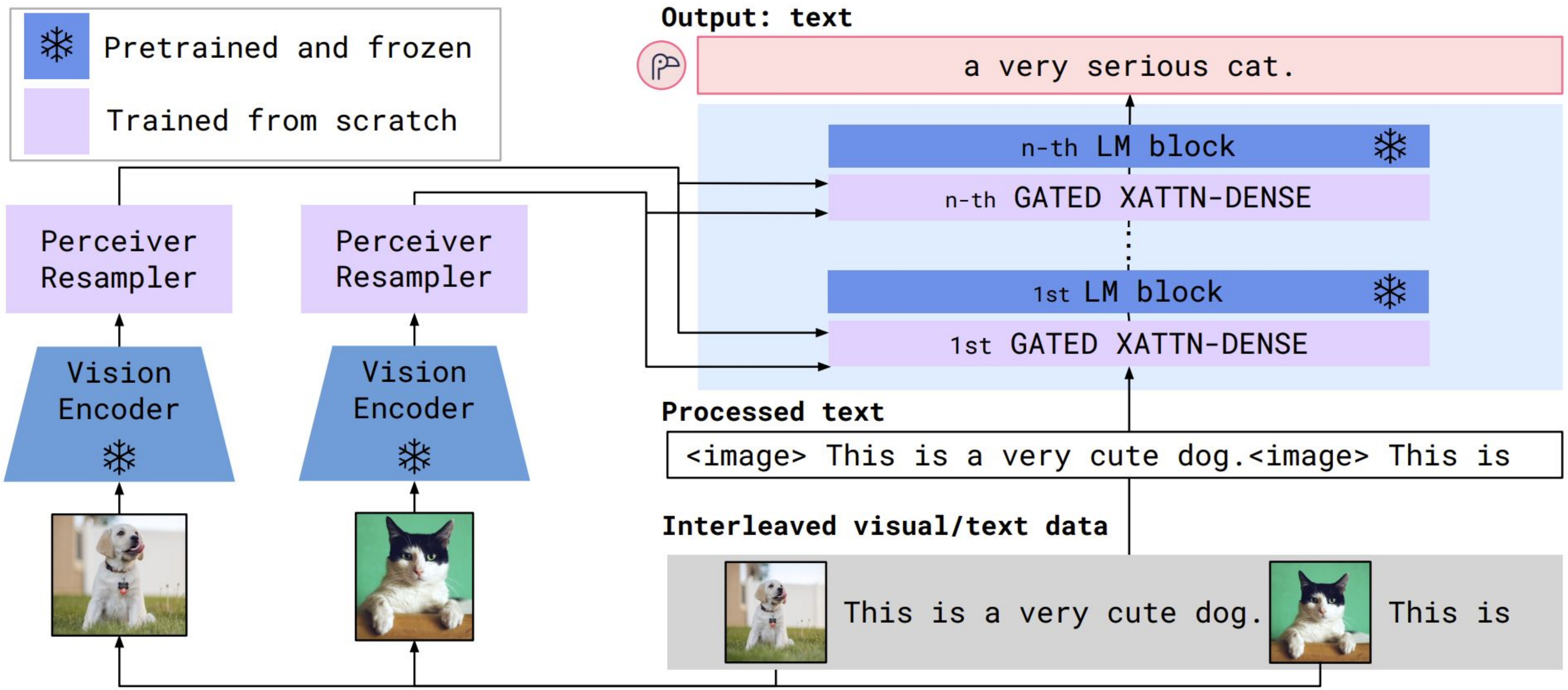


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Approach

Vision Encoder: From pixels to features

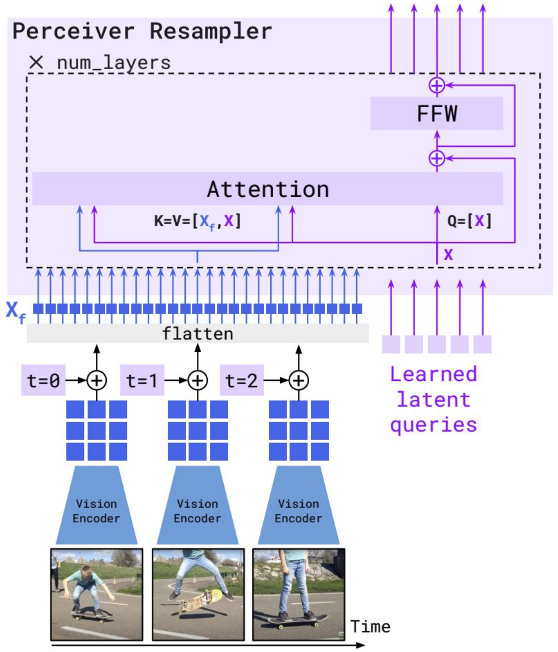
Architecture:

- Normalizer Free ResNet (NFNet)

Trained on:

- Datasets of image and text pairs, using the two-term contrastive loss from Radford et al.

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




```
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
```

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

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

At a given text token, the model attends to the visual tokens of the image that appeared just before it.

Approach

Training on a mixture of vision and language datasets

- Datasets
  - M3W:Interleaved image and text dataset.
  - ALIGN: 1.8B text-to-image
  - LTIP: 312M long-text and image
  - VTP: 27M short-video and text

Figure 9: **Training datasets.** Mixture of training datasets of different formats. N corresponds to the number of visual inputs for a single example. For paired image (or video) and text datasets, $N = 1$. T is the number of video frames ($T = 1$ for images). H, W, and C are height, width and color channels.

- Multi-objective training and optimisation strategy.
  - Tuning the per-dataset weights λm is key to performance.
  - Below weights were obtained empirically at a small model scale and kept fixed afterwards.

Dataset	M3W	ALIGN	LTIP	VTP
λm	1.0	0.2	0.2	0.03

Experiments and Results

Zero/Few-shot Performance

Method	FT	Shot	OKVQA (I)	VQAv2 (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	✗	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
	✗	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	-
	✗	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	✗	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
	✗	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	-
	✗	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	✗	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**
	✗	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	-
	✗	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	✔	(X)	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.

Georgia Tech

Experiments and Results

Fine-Tuning Performance

Method	VQAV2 test-dev	VQAV2 test-std	COCO test	VATEX test	VizWiz test-dev	VizWiz test-std	MSRVTTQA test	VisDial valid	VisDial test-std	YouCook2 valid	TextVQA valid	TextVQA test-std	HatefulMemes 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† [133]	81.3† [133]	**149.6**† [119]	81.4† [153]	57.2† [65]	60.6† [65]	46.8 [51]	**75.2** [79]	**75.4**† [123]	**138.7** [132]	54.7 [137]	**73.7** [84]	84.6† [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, outperfoming methods (marked with †) that use tricks such as model ensembling or domain-specific metric optimisation (e.g., CIDEr optimisation).

Experiments and Results

Ablation Study

	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	46.0	67.3
			w/o Image-Text pairs	3.2B	0.95s	66.3	39.2	51.6	32.0	41.6	60.9
			Image-Text pairs→ LAION	3.2B	1.74s	79.5	41.4	53.5	33.9	47.6	66.4
			w/o M3W	3.2B	1.02s	54.1	36.5	52.7	31.4	23.5	53.4
(ii)	Optimisation	Accumulation	Round Robin	3.2B	1.68s	76.1	39.8	52.1	33.2	40.8	62.9
(iii)	Tanh gating	✓	✗	3.2B	1.74s	78.4	40.5	52.9	35.9	47.5	66.5
(iv)	Cross-attention architecture	GATED XATTN-DENSE	VANILLA XATTN	2.4B	1.16s	80.6	41.5	53.4	32.9	50.7	66.9
			GRAFTING	3.3B	1.74s	79.2	36.1	50.8	32.2	47.8	63.1
(v)	Cross-attention frequency	Every	Single in middle	2.0B	0.87s	71.5	38.1	50.2	29.1	42.3	59.8
			Every 4th	2.3B	1.02s	82.3	42.7	55.1	34.6	50.8	68.8
			Every 2nd	2.6B	1.24s	83.7	41.0	55.8	34.5	49.7	68.2
(vi)	Resampler	Perceiver	MLP	3.2B	1.85s	78.6	42.2	54.7	35.2	44.7	66.6
			Transformer	3.2B	1.81s	83.2	41.7	55.6	31.5	48.3	66.7
(vii)	Vision encoder	NFNet-F6	CLIP ViT-L/14	3.1B	1.58s	76.5	41.6	53.4	33.2	44.5	64.9
			NFNet-F0	2.9B	1.45s	73.8	40.5	52.8	31.1	42.9	62.7
(viii)	Freezing LM	✓	✗ (random init)	3.2B	2.42s	74.8	31.5	45.6	26.9	50.1	57.8
			✗ (pretrained)	3.2B	2.42s	81.2	33.7	47.4	31.0	53.9	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.

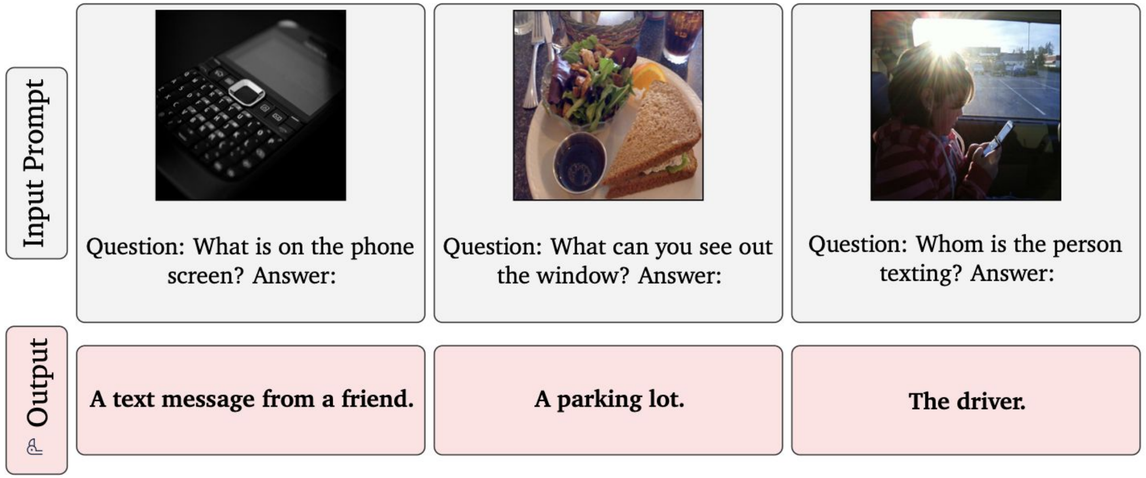
Limitations

Functional Limitations

- Hallucinations (Q)
- Poor generalization for long sequences
- Worse than contrastive models in classification
- Sensitivity to examples

Practical Limitations

- Text interface inconvenient for some tasks
- Expensive to train

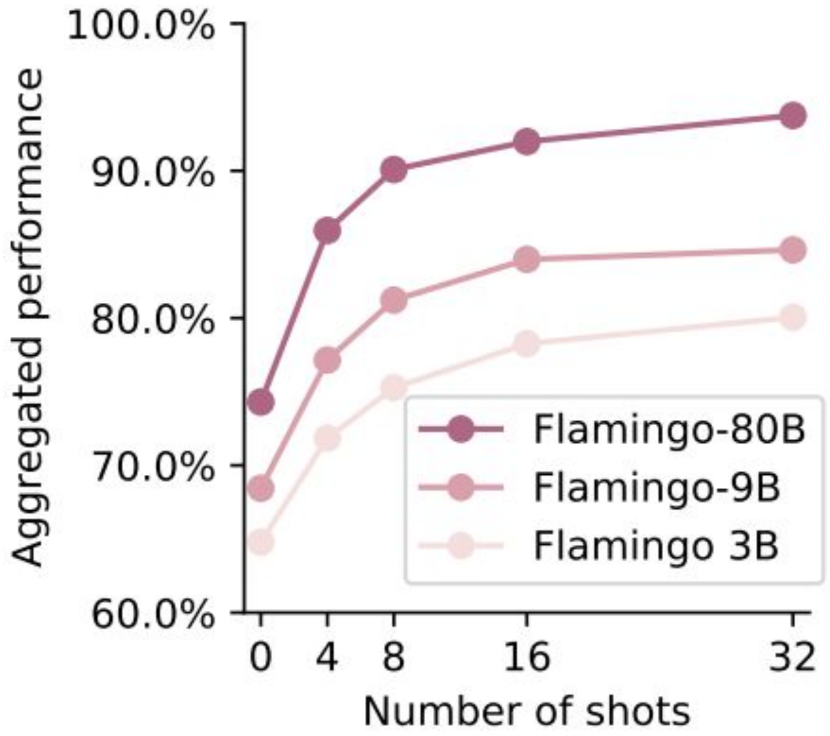


Q: Is the model simply inferring answers through the prompts without using images?

Limitations

Learning new task or identifying trained task?

- Performance plateaus as number of examples reach 32
- Non-trivial performance without images (Q)
- Examples may be locating task in memory (Q)
  - "Task Location" [8]



Q: Is the model learning a new task at inference or just identifying a task learned during training?

Q: Is it possible that the model's success is just due to the capabilities of the LM?

Societal Implications

Risks

- Good performance with less data
- Lower barrier for non-experts
- LLM risks
  - Offensive language
  - Propagating biases
  - Leaking private information

Benefits

- Good performance with less data
- Lower barrier for non-experts
- Identifying harmful behavior
  - Filtering toxic samples [9]
  - Probing another LM [10]

Strengths

Accessibility

- Few-shot task learning
- Chat interface
  - Non-expert use
  - Handles open-vocabulary prompts
  - Explainability and interpretability

Reusability

- Repurpose pretrained frozen models
  - Practical and environmental benefits
- New modalities can be introduced
- Only used 5 datasets for design decisions

Weaknesses

Performance Dependencies

- Weights of mixture dataset
- Large model size and large pretraining dataset size

Minor Issues

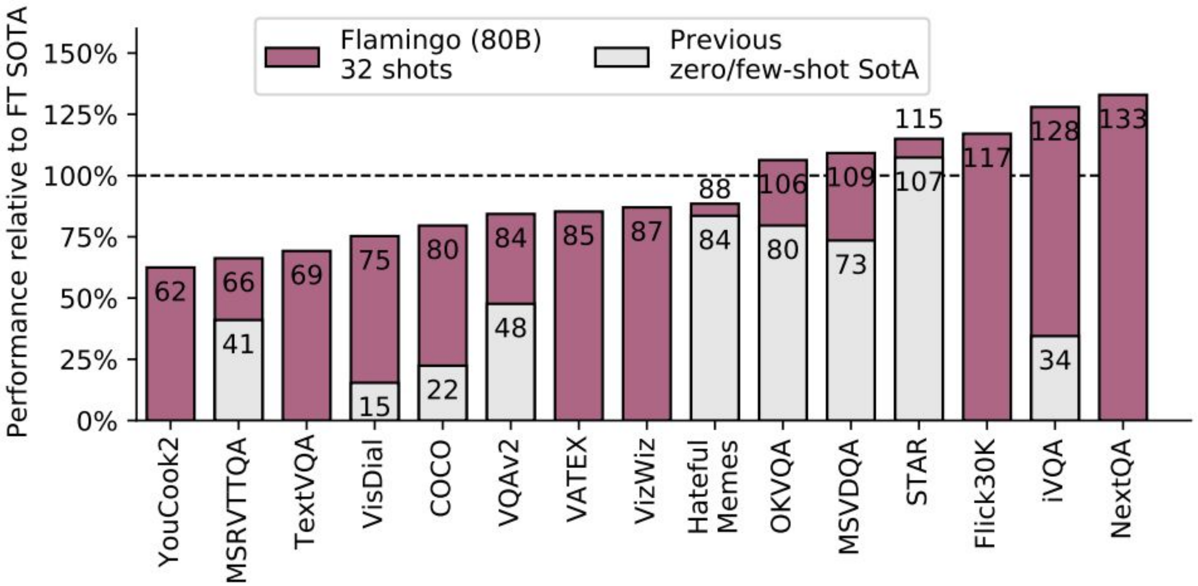
- Lack of detailed settings on downstream tasks, e.g. will <image> token also cross-attend to visual conditions?

Georgia Tech

Relationships to Other Papers

Frozen [11]

- Inspired Flamingo
- Could not achieve better performance than fine-tuned models
- Only handled images
- Only froze language model



References & Additional Resources

[1] Flamingo: a Visual Language Model for Few-Shot Learning, NeurIPS 2022
[2] Parameter-Efficient Transfer Learning for NLP, PMLR 2019
[3] BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models, ACL 2022
[4] Language Models are Few-Shot Learners, NeurIPS 2020
[5] The Power of Scale for Parameter-Efficient Prompt Tuning, ACL 2021
[6] Training Compute-Optimal Large Language Models, NeurIPS 2022
[7] Scaling Language Models: Methods, Analysis & Insights from Training Gopher, CoRR 2021
[8] Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm, CHI 2021
[9] LaMDA: Language Models for Dialog Applications
[10] Red Teaming Language Models with Language Models, ACL 2022
[11] Multimodal Few-Shot Learning with Frozen Language Models, NeurIPS 2021